

Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers

V200R009

CLI-based Configuration Guide - VPN Configuration

Issue 06

Date 2018-11-30

Copyright © Huawei Technologies Co., Ltd. 2018. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://e.huawei.com>

About This Document

Intended Audience

This document describes VPN features on the device and provides configuration procedures and configuration examples.

This document is intended for:

- Data configuration engineers
- Commissioning engineers
- Network monitoring engineers
- System maintenance engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.

Symbol	Description
 NOTE	<p>Calls attention to important information, best practices and tips.</p> <p>NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.</p>

Command Conventions

The command conventions that may be found in this document are defined as follows.

Convention	Description
Boldface	The keywords of a command line are in boldface .
<i>Italic</i>	Command arguments are in <i>italics</i> .
[]	Items (keywords or arguments) in brackets [] are optional.
{ x y ... }	Optional items are grouped in braces and separated by vertical bars. One item is selected.
[x y ...]	Optional items are grouped in brackets and separated by vertical bars. One item is selected or no item is selected.
{ x y ... }*	Optional items are grouped in braces and separated by vertical bars. A minimum of one item or a maximum of all items can be selected.
[x y ...]*	Optional items are grouped in brackets and separated by vertical bars. Several items or no item can be selected.
&<1-n>	The parameter before the & sign can be repeated 1 to n times.
#	A line starting with the # sign is comments.

Interface Numbering Conventions

Interface numbers used in this manual are examples. In device configuration, use the existing interface numbers on devices.

Security Conventions

- Password setting

- When configuring a password, the cipher text is recommended. To ensure device security, change the password periodically.
 - When you configure a password in plain text that starts and ends with %@%@, @%@%, %#%#, or %^%# (the password can be decrypted by the device), the password is displayed in the same manner as the configured one in the configuration file. Do not use this setting.
 - When you configure a password in cipher text, different features cannot use the same cipher-text password. For example, the cipher-text password set for the AAA feature cannot be used for other features.
- Encryption algorithm

Currently, the device uses the following encryption algorithms: 3DES, AES, RSA, SHA1, SHA2, and MD5. 3DES, RSA and AES are reversible, while SHA1, SHA2, and MD5 are irreversible. The encryption algorithms DES/3DES/RSA (RSA-1024 or lower)/MD5 (in digital signature scenarios and password encryption)/SHA1 (in digital signature scenarios) have a low security, which may bring security risks. If protocols allowed, using more secure encryption algorithms, such as AES/RSA (RSA-2048 or higher)/SHA2/HMAC-SHA2, is recommended. The encryption algorithm depends on actual networking. The irreversible encryption algorithm must be used for the administrator password, SHA2 is recommended.
 - Personal data

Some personal data may be obtained or used during operation or fault location of your purchased products, services, features, so you have an obligation to make privacy policies and take measures according to the applicable law of the country to protect personal data.
 - The terms mirrored port, port mirroring, traffic mirroring, and mirroring in this manual are mentioned only to describe the product's function of communication error or failure detection, and do not involve collection or processing of any personal information or communication data of users.

Mappings Between Product Software Versions and NMS Versions

The mappings between product software versions and NMS versions are as follows.

AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Product Software Version	eSight	iManager U2000
V200R009C00	V300R008C00	V200R017C60

Change History

Changes between document issues are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Changes in Issue 06 (2018-11-30)

This version has the following updates:

The following information is modified:

- [1.4 Licensing Requirements and Limitations for L2TP](#)
- [6.12.17 Example for Configuring L2TP Over IPSec to Implement Secure Communication Between the Headquarters and Branch](#)
- [6.5 Licensing Requirements and Limitations for IPSec](#)

Changes in Issue 05 (2018-07-06)

This version has the following updates:

The following information is modified:

- [7.4 Licensing Requirements and Limitations for A2A VPN](#)

Changes in Issue 04 (2018-03-23)

This version has the following updates:

The following information is added:

- [7.4 Licensing Requirements and Limitations for A2A VPN](#)
- [4.5 Licensing Requirements and Limitations for SVPN](#)

Changes in Issue 03 (2018-01-05)

This version has the following updates:

The following information is added:

- [3.6.8 \(Optional\) Configuring the DF Flag Bit for GRE Packets](#)

The following information is modified:

- [6.12.9 Example for Establishing an IPSec Tunnel Through NAT Traversal](#)
- [1.4 Licensing Requirements and Limitations for L2TP](#)
- [3.4 Licensing Requirements and Limitations for GRE](#)
- [6.5 Licensing Requirements and Limitations for IPSec](#)
- [5.4 Licensing Requirements and Limitations for DSVPN](#)

Changes in Issue 02 (2017-10-13)

This version has the following updates:

The following information is modified:

- VPN Access

- [5.8.7 Example for Configuring Shortcut Scenario of DSVPN \(BGP\)](#)
- [5.2.5.2 Dual Hubs in Load Balancing Mode](#)
- [5.8.6 Example for Configuring Shortcut Scenario of DSVPN \(OSPF\)](#)
- [5.8.5 Example for Configuring Shortcut Scenario of DSVPN \(RIP\)](#)

Changes in Issue 01 (2017-08-04)

Initial commercial release.

Contents

About This Document.....	ii
1 L2TP Configuration.....	1
1.1 Overview of L2TP.....	2
1.2 Understanding L2TP.....	3
1.2.1 Concepts.....	3
1.2.2 L2TP Implementation.....	6
1.2.3 Working Procedure.....	8
1.3 Application Scenarios for L2TP.....	11
1.3.1 Client-Initiated L2TP Connection.....	11
1.3.2 LAC-Initiated L2TP Connection upon Receiving a Call Connection Request.....	12
1.3.3 LAC-Initiated L2TP Connection upon Receiving a Call from a PPPoE User.....	12
1.3.4 L2TP Client-Initiated L2TP Connection.....	13
1.3.5 LAC-Initiated L2TP Connection When Users from Multiple Domains Are Connected.....	13
1.3.6 Authenticating VPDN Users Using the RADIUS Server.....	14
1.3.7 Allocating the Frame-IP and Frame-Route Attributes and the Specified Address Pool Name to L2TP Users by the RADIUS Server.....	15
1.3.8 Setting Up a Secure Tunnel Connection Using L2TP over IPSec Encapsulation.....	15
1.3.9 Setting Up a Secure Tunnel Connection Using IPSec over L2TP Encapsulation.....	16
1.4 Licensing Requirements and Limitations for L2TP.....	17
1.5 Default Settings for L2TP.....	17
1.6 Configuring L2TP.....	17
1.6.1 Configuring the LAC to Initiate Call-Triggered L2TP Connections.....	17
1.6.1.1 Configuring AAA Authentication and Accounting.....	20
1.6.1.2 Configuring the LAC to Accept Dial-Up Calls and Initiate L2TP Connections.....	23
1.6.1.3 Configuring the LNS to Respond to the L2TP Connection Request.....	25
1.6.2 Configuring L2TP Client-Initiated L2TP Connections.....	29
1.6.2.1 Configuring AAA Authentication and Accounting.....	31
1.6.2.2 Configuring the L2TP Client to Dial Up and Initiate L2TP Connections.....	34
1.6.2.3 Configuring the LNS to Respond to the L2TP Connection Request.....	36
1.6.3 Configuring Other L2TP Functions.....	39
1.6.3.1 Configuring LCP Renegotiation.....	40
1.6.3.2 Configuring CHAP Mandatory Authentication.....	40
1.6.3.3 Configuring Primary and Secondary LNSs.....	41

1.6.3.4 Configuring AVP Parameter Encryption.....	42
1.6.3.5 Configuring L2TP Tunnel Authentication.....	42
1.6.3.6 Configuring L2TP Tunnel Connectivity.....	43
1.6.4 Verifying the L2TP Configuration.....	44
1.7 Maintaining L2TP.....	44
1.7.1 Disconnecting an L2TP Tunnel Manually.....	44
1.7.2 Monitoring the Running Status of L2TP.....	45
1.7.3 Collecting L2TP Packet Statistics.....	45
1.8 Configuration Examples for L2TP.....	46
1.8.1 Example for Configuring Client-Initiated L2TP Connections.....	46
1.8.2 Example for Configuring the LAC to Initiate Call-Triggered L2TP Connections (Dial-Up Users).....	56
1.8.3 Example for Configuring the LAC to Initiate Call-Triggered L2TP Connections (PPPoE Users).....	58
1.8.4 Example for Configuring an L2TP Client-Initiated L2TP Connection.....	62
1.8.5 Example for Configuring L2TP Client-Initiated L2TP Connections.....	66
1.8.6 Example for Configuring L2TP Client-Initiated L2TP Connections Using the 3G Interface.....	72
1.9 Troubleshooting L2TP.....	77
1.9.1 User Failed to Dial Up to the LNS.....	78
1.9.2 Data Transmission Fails After L2TP Connections Are Established.....	80
1.10 FAQ About L2TP.....	81
1.10.1 Starting from Which Version Does the device Support NAT Traversal in L2TP?.....	81
1.10.2 L2TP Dialup Is Successful After Dozens of Attempts and Error 691 Is Displayed. Why?.....	81
1.10.3 How Can I Quickly Locate Why the LAC Cannot Set Up an L2TP Tunnel with the LNS?.....	81
1.10.4 How Do I Configure the LNS That Trusts the LAC Not to Perform Second Authentication on Remote Users?.....	82
1.10.5 What Can I Do If a PC Running the Windows 7 or XP Operating System Fails to Establish an L2TP over IPSec Tunnel with the Device?.....	82
1.11 References for L2TP.....	83
2 L2TPv3 Configuration.....	84
2.1 Overview of L2TPv3.....	84
2.2 Understanding L2TPv3.....	85
2.3 Application Scenarios for L2TPv3.....	87
2.4 Licensing Requirements and Limitations for L2TPv3.....	88
2.5 Configuring L2TPv3.....	88
2.5.1 Configuring a Static L2TPv3 Tunnel.....	89
2.5.2 Verifying the L2TPv3 Configuration.....	90
2.6 Monitoring the L2TPv3 Tunnel Running Status.....	91
2.7 Configuration Examples for L2TPv3.....	91
2.7.1 Example for Establishing a Static L2TPv3 Tunnel.....	91
2.7.2 Example for Configuring L2TPv3 over IPSec to Implement Secure Communication Between Branches.....	96
2.8 References for L2TPv3.....	103
3 GRE Configuration.....	105
3.1 Overview of GRE.....	106

3.2 Understanding GRE.....	106
3.2.1 Basic Concepts.....	106
3.2.2 GRE Security Mechanisms.....	109
3.2.3 Keepalive Detection.....	110
3.2.4 Ethernet over GRE.....	111
3.2.5 Ethernet over mGRE.....	112
3.3 Application Scenarios for GRE.....	114
3.3.1 Transmitting Data of Multi-Protocol Local Networks Through a GRE Tunnel.....	114
3.3.2 Enlarging the Operation Scope of a Network with a Hop Limit.....	114
3.3.3 Combining GRE with IPSec to Protect Multicast Data.....	115
3.3.4 Setting Up an L2VPN and an L3VPN Using a GRE Tunnel.....	115
3.3.5 Connecting CE Devices to an MPLS VPN Network.....	117
3.3.6 Ethernet over GRE Application.....	120
3.3.7 Ethernet over mGRE Application.....	120
3.4 Licensing Requirements and Limitations for GRE.....	121
3.5 Default Settings for GRE.....	121
3.6 Configuring a GRE Tunnel.....	121
3.6.1 Configuring a Tunnel Interface.....	122
3.6.2 Configuring a Route on a Tunnel Interface.....	125
3.6.3 (Optional) Configuring the Link Bridge Function.....	126
3.6.4 (Optional) Configuring a Security Mechanism for GRE.....	130
3.6.5 (Optional) Enabling the Keepalive Detection Function for GRE.....	130
3.6.6 (Optional) Configuring Ethernet over GRE.....	131
3.6.7 (Optional) Configuring Ethernet over mGRE.....	133
3.6.8 (Optional) Configuring the DF Flag Bit for GRE Packets.....	134
3.6.9 Verifying the GRE Tunnel Configuration.....	135
3.7 Maintaining the GRE Tunnel.....	135
3.7.1 Collecting and Viewing Statistics on Tunnel Interfaces.....	135
3.7.2 Monitoring the GRE Running Status.....	136
3.7.3 Resetting the Keepalive Packet Statistics on a Tunnel Interface.....	137
3.8 Configuration Examples for GRE.....	137
3.8.1 Example for Configuring a Static Route for GRE to Implement Interworking Between IPv4 Networks.....	138
3.8.2 Example for Configuring OSPF for GRE to Implement Interworking Between IPv4 Networks.....	142
3.8.3 Example for Configuring a GRE Tunnel to Implement Interworking Between IPv6 Networks.....	147
3.8.4 Example for Enlarging the Operation Scope of a Network with a Hop Limit.....	151
3.8.5 Example for Configuring BGP/MPLS IP VPN to Use a GRE Tunnel.....	156
3.8.6 Example for Configuring VLL to Use a GRE Tunnel.....	164
3.8.7 Example for Connecting a CE to a VPN Through a GRE Tunnel over a Public Network.....	170
3.8.8 Example for Connecting a CE to a VPN Through a GRE Tunnel over a VPN.....	178
3.8.9 Example for Configuring GRE to Implement Communication Between FR Networks.....	186
3.8.10 Example for Configuring an Ethernet over GRE Tunnel.....	189
3.8.11 Example for Configuring an Ethernet over mGRE Tunnel.....	194

3.9 Troubleshooting GRE.....	200
3.9.1 Failed to Ping the IP Address of the Remote Tunnel Interface.....	200
3.9.2 Tunnel Interface Alternates Between Up and Down States.....	202
3.10 FAQ About GRE.....	202
3.10.1 Can the MTU of the GRE Tunnel Interface Take Effect?.....	202
3.11 References for GRE.....	202
4 SVPN Configuration.....	204
4.1 Overview of SVPN.....	204
4.2 Understanding SVPN.....	206
4.3 Application Scenarios for SVPN.....	209
4.3.1 Fully Using WAN Links for Internet Access Through Lone Ranger SVPN.....	209
4.3.2 Fully Using WAN Links for Implementing Interworking Between the Enterprise Branch and Headquarters Through Hub-Spoke SVPN.....	210
4.4 Summary of SVPN Configuration Tasks.....	211
4.5 Licensing Requirements and Limitations for SVPN.....	212
4.6 Default Settings for SVPN.....	212
4.7 Configuring SVPN.....	213
4.7.1 Configuring Lone Ranger SVPN.....	213
4.7.1.1 Configuring a Tunnel Interface.....	213
4.7.1.2 Configuring an SVPN Proposal.....	214
4.7.1.3 Binding an SVPN Proposal to a Tunnel Interface.....	215
4.7.1.4 Importing Service Flows to an SVPN Tunnel.....	215
4.7.2 Configuring Hub-Spoke SVPN.....	216
4.7.2.1 Configuring a Tunnel Interface.....	216
4.7.2.2 Configuring an SVPN Proposal.....	218
4.7.2.3 Binding an SVPN Proposal to a Tunnel Interface.....	218
4.7.2.4 Importing Service Flows to an SVPN Tunnel.....	219
4.8 Configuration Examples for SVPN.....	221
4.8.1 Example for Configuring Lone Ranger SVPN of the Overflow Mode.....	221
4.8.2 Example for Configuring Hub-Spoke svpn of the Priority Mode.....	224
5 DSVPN Configuration.....	231
5.1 Overview of DSVPN.....	231
5.2 Understanding DSVPN.....	236
5.2.1 Basic Concepts.....	236
5.2.2 Implementation.....	239
5.2.3 DSVPN NAT Traversal.....	247
5.2.4 DSVPN Protected by IPSec.....	249
5.2.5 DSVPN Reliability.....	251
5.2.5.1 Dual Hubs in Active/Standby Mode.....	251
5.2.5.2 Dual Hubs in Load Balancing Mode.....	253
5.3 Application Scenarios for DSVPN.....	255
5.3.1 DSVPN Deployment on a Small- or Medium-sized Network.....	255

5.3.2 DSVPN Deployment on a Large-sized Network.....	257
5.3.3 Deploying DSVPN in Hierarchical Hub Networking.....	259
5.4 Licensing Requirements and Limitations for DSVPN.....	260
5.5 Default Settings for DSVPN.....	261
5.6 Configuring DSVPN.....	262
5.6.1 Configuring mGRE.....	262
5.6.2 Configuring Routes.....	263
5.6.3 Configuring NHRP.....	265
5.6.4 (Optional) Configuring an IPSec Profile.....	267
5.6.5 Verifying the DSVPN Configuration.....	268
5.7 Maintaining DSVPN.....	268
5.7.1 Clearing DSVPN Running Statistics.....	269
5.7.2 Monitoring DSVPN Running Statistics.....	269
5.8 Configuration Examples for DSVPN.....	269
5.8.1 Example for Configuring Non-Shortcut Scenario of DSVPN (Static Route).....	269
5.8.2 Example for Configuring Non-Shortcut Scenario of DSVPN (RIP).....	276
5.8.3 Example for Configuring Non-Shortcut Scenario of DSVPN (OSPF).....	282
5.8.4 Example for Configuring Non-Shortcut Scenario of DSVPN (BGP).....	288
5.8.5 Example for Configuring Shortcut Scenario of DSVPN (RIP).....	295
5.8.6 Example for Configuring Shortcut Scenario of DSVPN (OSPF).....	302
5.8.7 Example for Configuring Shortcut Scenario of DSVPN (BGP).....	308
5.8.8 Example for Configuring DSVPN NAT traversal.....	316
5.8.9 Example for Configuring Dual Hubs in Active/Standby Mode.....	323
5.8.10 Example for Configuring DSVPN Protected by IPSec.....	331
5.8.11 Example for Configuring a Dual-Hub DSVPN Protected by IPSec.....	342
5.8.12 Example for Configuring a DSVPN Based on the LTE Dialup Status.....	354
5.9 Troubleshooting DSVPN.....	369
5.9.1 Spoke Fails to Register with a Hub.....	369
5.9.2 Subnets Between Spokes Cannot Communicate Directly in Non-Shortcut Mode.....	370
5.9.3 Subnets Between Spokes Cannot Communicate Directly in Shortcut Mode.....	371
5.9.4 Backup Hub Only Forwards Data After the Master Hub Fails.....	371
5.10 References for DSVPN.....	372
6 IPSec Configuration.....	373
6.1 Overview of IPSec.....	374
6.2 Understanding IPSec.....	375
6.2.1 Basic Concepts of IPSec.....	375
6.2.1.1 Security Association.....	375
6.2.1.2 Security Protocol.....	376
6.2.1.3 Encapsulation Mode.....	380
6.2.1.4 Encryption.....	381
6.2.1.5 Authentication.....	382
6.2.1.6 Key Exchange.....	384

6.2.1.7 IKE.....	384
6.2.2 IPsec Fundamentals.....	387
6.2.2.1 Defining IPsec Protected Data Flows.....	387
6.2.2.2 Establishing an SA Through IKEv1 Negotiation.....	387
6.2.2.3 Establishing an SA Through IKEv2 Negotiation.....	390
6.2.3 IPsec Enhancements.....	392
6.2.3.1 L2TP over IPsec.....	392
6.2.3.2 GRE over IPsec.....	394
6.2.3.3 IPsec Multi-instance.....	395
6.2.3.4 Efficient VPN.....	396
6.2.4 IPsec Reliability.....	399
6.2.4.1 Link Redundancy.....	399
6.3 Application Scenarios for IPsec.....	401
6.3.1 Using IPsec VPN to Implement Secure Interconnection Between LANs.....	402
6.3.2 Using IPsec VPN to Provide Secure Remote Access for Mobile Users.....	404
6.3.3 Secure LAN Interconnection Through Efficient VPN.....	405
6.4 Summary of IPsec Configuration Tasks.....	406
6.5 Licensing Requirements and Limitations for IPsec.....	408
6.6 Default Settings for IPsec.....	410
6.7 Using an ACL to Establish an IPsec Tunnel.....	411
6.7.1 Defining Data Flows to Be Protected.....	413
6.7.2 Configuring an IPsec Proposal.....	417
6.7.3 Configuring an IPsec Policy.....	419
6.7.3.1 Configuring an IPsec Policy in Manual Mode.....	419
6.7.3.2 Configuring an IPsec Policy in ISAKMP Mode.....	422
6.7.3.3 Configuring an IPsec Policy Using an IPsec Policy Template.....	424
6.7.4 (Optional) Setting the IPsec SA Lifetime.....	427
6.7.5 (Optional) Enabling the Anti-replay Function.....	429
6.7.6 (Optional) Configuring IPsec Fragmentation Before Encryption.....	430
6.7.7 (Optional) Configuring Route Injection.....	431
6.7.8 (Optional) Configuring IPsec Check.....	433
6.7.9 (Optional) Enabling the QoS Function for IPsec Packets.....	433
6.7.10 (Optional) Configuring IPsec VPN Multi-instance.....	435
6.7.11 (Optional) Allowing New Users with the Same Traffic Rule as Original Branch Users to Access the Headquarters Network.....	436
6.7.12 (Optional) Configuring a Multi-link Shared IPsec Policy Group.....	436
6.7.13 (Optional) Configuring Redundancy Control of IPsec Tunnels.....	438
6.7.14 Applying an IPsec Policy Group to an Interface.....	439
6.7.15 Verifying the Configuration of IPsec Tunnel Establishment.....	441
6.8 Using a Virtual Tunnel Interface to Establish an IPsec Tunnel.....	441
6.8.1 Configuring an IPsec Proposal.....	442
6.8.2 Configuring an IPsec Profile.....	443
6.8.3 (Optional) Setting the SA Lifetime.....	445

6.8.4 (Optional) Enabling the Anti-replay Function.....	446
6.8.5 (Optional) Configuring IPSec Fragmentation Before Encryption.....	447
6.8.6 (Optional) Configuring IPSec Check.....	448
6.8.7 (Optional) Enabling the QoS Function for IPSec Packets.....	449
6.8.8 (Optional) Configuring Requesting, Sending or Accepting of Subnet Route Information.....	450
6.8.9 Configuring a Tunnel Interface or a Tunnel Template Interface.....	455
6.8.10 Verifying the Configuration of IPSec Tunnel Establishment Using a Virtual Tunnel Interface.....	458
6.9 Establishing an IPSec Tunnel Using an Efficient VPN Policy.....	459
6.9.1 Configuring the Remote Device.....	459
6.9.2 Configuring the Efficient VPN Server.....	466
6.9.3 Verifying the Efficient VPN Configuration.....	468
6.10 Configuring IKE.....	469
6.10.1 Configuring an IKE Proposal.....	469
6.10.2 Configuring an IKE Peer.....	471
6.10.3 (Optional) Setting the IKE SA Lifetime.....	478
6.10.4 (Optional) Configuring IKE Peer Status Detection.....	479
6.10.4.1 (Optional) Configuring Heartbeat Detection.....	480
6.10.4.2 (Optional) Configuring DPD.....	481
6.10.5 (Optional) Configuring an Identity Filter Set.....	482
6.10.6 (Optional) Configuring DSCP Priority for IKE Packets.....	485
6.10.7 (Optional) Configuring NAT Traversal.....	486
6.10.8 (Optional) Configuring IPSec VPN Multi-instance.....	487
6.10.9 (Optional) Configuring Network Resource Delivery.....	489
6.10.10 (Optional) Configuring ACL Delivery.....	489
6.10.11 (Optional) Enabling Dependency Between IPSec SA and IKE SA During IKEv1 Negotiation.....	490
6.10.12 (Optional) Configuring Rapid Switchover and Revertive Switching of an IKE Peer.....	491
6.10.13 Verifying the IKE Configuration.....	492
6.11 Maintaining IPSec.....	492
6.11.1 Monitoring the IPSec Running Status.....	493
6.11.2 Clearing IPSec Statistics.....	493
6.12 Configuration Examples for IPSec.....	494
6.12.1 Example for Manually Establishing an IPSec Tunnel.....	494
6.12.2 Example for Establishing an IPSec Tunnel in IKE Negotiation Mode Using Default Settings.....	498
6.12.3 Example for Establishing an IPSec Tunnel Between the Enterprise Headquarters and Branch Using an IPSec Policy Template.....	503
6.12.4 Example for Establishing Multiple IPSec Tunnels Between the Enterprise Headquarters and Branches Using IPSec Policy Groups.....	511
6.12.5 Example for Establishing IPSec Tunnels for Branch Access to the Headquarters Using Different Pre-shared Keys.....	518
6.12.6 Example for Establishing an IPSec Tunnel Between the Branch and Headquarters with a Redundant Gateway.....	525
6.12.7 Example for Establishing an IPSec Tunnel Between the Enterprise Headquarters and Branch Using a Multi-Link Shared IPSec Policy Group.....	532

6.12.8 Example for Establishing an IPsec Tunnel Between the Enterprise Headquarters and Branch Through PPPoE.....	537
6.12.9 Example for Establishing an IPsec Tunnel Through NAT Traversal.....	542
6.12.10 Example for Establishing an IPsec Tunnel in IKE Negotiation Mode by Specifying DNS.....	547
6.12.11 Example for Establishing an IPsec Tunnel Through Negotiation Initiated by the Branch User That Dynamically Obtains an IP Address.....	554
6.12.12 Example for Establishing an IPsec Tunnel Using a Tunnel Interface.....	559
6.12.13 Example for Establishing GRE over IPsec Using a Tunnel Interface.....	564
6.12.14 Example for Establishing IPsec over GRE Using a Tunnel Interface.....	568
6.12.15 Example for Establishing an IPsec over GRE Tunnel Between the Headquarters and Branch (Based on ACL).....	574
6.12.16 Example for Establishing IPsec over DSVPN Tunnels Between Hub and Spokes (Based on ACL).....	579
6.12.17 Example for Configuring L2TP Over IPsec to Implement Secure Communication Between the Headquarters and Branch.....	587
6.12.18 Example for Configuring a Tunnel Template Interface for IPsec Tunnel Setup.....	594
6.12.19 Example for Establishing an IPsec Tunnel Using an Efficient VPN Policy in Client Mode.....	599
6.12.20 Example for Configuring an IPsec Tunnel Using an Efficient VPN Policy in Network Mode.....	604
6.12.21 Example for Configuring an IPsec Tunnel Using an Efficient VPN Policy in Network-Plus Mode.....	608
6.12.22 Example for Configuring Efficient VPN in Network-auto-cfg Mode to Establish an IPsec Tunnel.....	612
6.12.23 Example for Configuring Automatic Upgrade of the Efficient VPN Remote Device.....	617
6.12.24 Example for Configuring Rapid Switchover and Revertive Switching.....	623
6.12.25 Example for Configuring Redundancy Control of IPsec Tunnels.....	631
6.13 Troubleshooting IPsec.....	643
6.13.1 IKE SA Negotiation Failed.....	643
6.13.2 IPsec SA Negotiation Failed.....	644
6.13.3 Services Are Interrupted After an IPsec Tunnel Is Established.....	646
6.14 FAQ About IPsec.....	649
6.14.1 Private Network Communication Fails After IPsec Is Configured. What Are the Causes?.....	649
6.14.2 How Do I Rectify the Failure to View SA Information by Running the display ipsec sa Command After IPsec Is Configured?.....	649
6.14.3 Does the Interface with a Dynamic IP Address Support IPsec?.....	650
6.14.4 IPsec Does Not Take Effect When Both IPsec and NAT Are Configured on a Device Interface. How This Problem Is Solved?.....	651
6.14.5 Why Cannot an IPsec Tunnel Be Established Until It Is Restarted?.....	651
6.15 References for IPsec.....	651
7 A2A VPN Configuration.....	654
7.1 Overview of A2A VPN.....	654
7.2 Understanding A2A VPN.....	655
7.2.1 Basic Networking.....	655
7.2.2 Implementation.....	657
7.2.2.1 GM Registering with the KS.....	657
7.2.2.2 GM Data Protection.....	658
7.2.2.3 Rekey.....	659
7.3 Application Scenarios for A2A VPN.....	659

7.3.1 Typical A2A VPN Networking.....	659
7.3.2 A2A VPN Redundancy.....	661
7.4 Licensing Requirements and Limitations for A2A VPN.....	662
7.5 Default Settings for A2A VPN.....	663
7.6 Configuring A2A VPN.....	663
7.6.1 Configuring a GM.....	663
7.6.1.1 Configuring IKE.....	664
7.6.1.2 (Optional) Defining Data Flows Not to Be Protected.....	664
7.6.1.3 Configuring a GDOI Policy.....	665
7.6.1.4 Configuring an IP Address for Multicast Rekey Messages.....	666
7.6.1.5 (Optional) Configuring the Receive_Option Mode.....	667
7.6.1.6 (Optional) Configuring the QoS Function for A2A VPN.....	667
7.6.1.7 (Optional) Configuring Fragmentation Before Encryption.....	669
7.6.1.8 Applying a GDOI Policy Group to an Interface.....	669
7.6.1.9 Verifying the GM Configuration.....	670
7.7 Maintaining A2A VPN.....	671
7.7.1 Monitoring the A2A VPN Status.....	671
7.7.2 Clearing A2A VPN Statistics.....	671
7.8 Configuration Examples for A2A VPN.....	672
7.8.1 Example for Configuring a Typical A2A VPN Networking.....	672
7.8.2 Example for Configuring GM Link Redundancy.....	680
7.9 Troubleshooting A2A VPN.....	687
7.9.1 GM Fails to Register with the KS.....	687
7.10 A2A VPN FAQ.....	688
7.10.1 Why Some Service Packets Are Lost After A2A VPN Is Deployed?.....	688
7.11 References for A2A VPN.....	688
8 BGP/MPLS IP VPN Configuration.....	690
8.1 Overview of BGP/MPLS IP VPN.....	691
8.2 Understanding BGP/MPLS IP VPN.....	692
8.2.1 Concepts.....	692
8.2.2 Implementation.....	697
8.2.3 Basic Networking.....	702
8.2.4 Inter-AS VPN.....	704
8.2.5 MCE.....	711
8.2.6 HoVPN.....	714
8.2.7 VPN FRR.....	718
8.2.8 VPN GR.....	720
8.2.9 VPN NSR.....	721
8.2.10 VPN Tunnel Policy.....	721
8.3 Application Scenarios for BGP/MPLS IP VPN.....	724
8.3.1 BGP/MPLS IP VPN Application.....	724
8.3.2 Hub and Spoke Networking Application.....	725

8.3.3 Interconnection Between VPNs and the Internet.....	727
8.4 Summary of BGP/MPLS IP VPN Configuration Tasks.....	730
8.5 Licensing Requirements and Limitations for BGP/MPLS IP VPN.....	734
8.6 Default Settings for BGP/MPLS IP VPN.....	734
8.7 Configuring BGP/MPLS IP VPN.....	735
8.7.1 Configuring Basic BGP/MPLS IP VPN Functions.....	735
8.7.1.1 Configuration Tasks.....	735
8.7.1.2 Establishing MP-IBGP Peer Relationships Between PE Devices.....	737
8.7.1.3 Configuring a VPN Instance on a PE Device.....	737
8.7.1.4 Binding a VPN Instance to an Interface.....	740
8.7.1.5 Configuring Route Exchange Between PE and CE Devices.....	741
8.7.1.6 Verifying the Configuration of Basic BGP/MPLS IP VPN Functions.....	755
8.7.2 Configuring Hub and Spoke.....	756
8.7.2.1 Configuring MP-IBGP Between Hub-PE and Spoke-PE.....	756
8.7.2.2 Configuring VPN Instances on PE Devices.....	757
8.7.2.3 Binding a VPN Instance to an Interface.....	759
8.7.2.4 Configuring Route Exchange Between PE device and CE Devices.....	760
8.7.2.5 Verifying the Hub and Spoke Configuration.....	761
8.7.3 Configuring Inter-AS VPN Option A.....	762
8.7.4 Configuring Inter-AS VPN Option B.....	762
8.7.4.1 Configuring MP-IBGP Between PE and ASBR in the Same AS.....	763
8.7.4.2 Configuring MP-EBGP Between ASBRs in Different ASs.....	764
8.7.4.3 Disabling an ASBR from Filtering VPNv4 Routes by VPN Targets.....	765
8.7.4.4 (Optional) Configuring Routing Policies to Control VPN Route Advertisement and Acceptance.....	766
8.7.4.5 (Optional) Enabling Next-Hop-based Label Allocation on the ASBR.....	767
8.7.4.6 Verifying the Inter-AS VPN Option B Configuration.....	768
8.7.5 Configuring Inter-AS VPN Option C (Solution 1).....	769
8.7.5.1 Enabling the Labeled IPv4 Route Exchange.....	770
8.7.5.2 Configuring a Routing Policy to Control Label Distribution.....	771
8.7.5.3 Establishing an MP-EBGP Peer Relationship Between PE Devices.....	773
8.7.5.4 Verifying the Inter-AS VPN Option C Configuration (Solution 1).....	775
8.7.6 Configuring Inter-AS VPN Option C (Solution 2).....	775
8.7.6.1 Establishing the EBGP Peer Relationship Between ASBRs.....	776
8.7.6.2 Advertising the Routes of the PE in the Local AS to the Remote PE.....	777
8.7.6.3 Enabling the Capability of Exchanging Labeled IPv4 Routes.....	778
8.7.6.4 Establishing an LDP LSP for the Labeled BGP Routes of the Public Network.....	779
8.7.6.5 Establishing the MP-EBGP Peer Relationship Between PEs.....	779
8.7.6.6 Verifying the Inter-AS VPN Option C Configuration (Solution 2).....	780
8.7.7 Configuring an MCE Device.....	781
8.7.7.1 Configure Route Exchange Between an MCE Device and VPN Sites.....	781
8.7.7.2 Configure Route Exchange Between an MCE Device and a PE Device.....	786
8.7.7.3 Verifying the MCE Configuration.....	790

8.7.8 Configuring HoVPN.....	790
8.7.9 Configuring PBR to an LSP for VPN Packets.....	792
8.7.10 Configuring an OSPF Sham Link.....	793
8.7.11 Configuring Route Reflection to Optimize the VPN Backbone Layer.....	797
8.7.11.1 Configuring the Client PEs to Establish MP IBGP Connections with the RR.....	797
8.7.11.2 Configuring the RR to Establish MP IBGP Connections with the Client PEs.....	798
8.7.11.3 Configuring Route Reflection for BGP IPv4 VPN Routes.....	799
8.7.11.4 Verifying the Configuration of Route Reflection to Optimize the VPN Backbone Layer.....	800
8.7.12 Configuring IP FRR for VPN Routes.....	800
8.7.13 Configuring VPN FRR.....	802
8.7.14 Configuring VPN GR.....	805
8.7.15 Configuring Tunnel Policies.....	806
8.7.15.1 Configuring and Applying a Tunnel Policy.....	807
8.7.15.2 Configuring and Applying a Tunnel Selector.....	809
8.7.16 Connecting a VPN to the Internet.....	811
8.8 Maintaining BGP/MPLS IP VPN.....	813
8.8.1 Collecting Statistics About L3VPN Traffic.....	813
8.8.2 Checking L3VPN Traffic.....	813
8.8.3 Clearing L3VPN Traffic.....	814
8.8.4 Displaying BGP/MPLS IP VPN Information.....	814
8.8.5 Checking Network Connectivity and Reachability.....	815
8.8.6 Viewing the Integrated Route Statistics of IPv4 VPN Instances.....	816
8.8.7 Resetting BGP Statistics of a VPN Instance IPv4 Address Family.....	816
8.8.8 Resetting BGP Connections.....	816
8.8.9 Monitoring the Running Status of VPN Tunnels.....	817
8.9 Configuration Examples for BGP/MPLS IP VPN.....	817
8.9.1 Example for Configuring BGP/MPLS IP VPN.....	818
8.9.2 Example for Configuring BGP/MPLS IP VPNs with Overlapping Address Spaces.....	829
8.9.3 Example for Configuring Communication Between Local VPNs.....	839
8.9.4 Example for Configuring Hub and Spoke.....	843
8.9.5 Example for Configuring Inter-AS VPN Option A.....	852
8.9.6 Example for Configuring Inter-AS VPN Option B.....	863
8.9.7 Example for Configuring Inter-AS VPN Option C (Solution 1).....	872
8.9.8 Example for Configuring Inter-AS VPN Option C (Solution 2).....	883
8.9.9 Example for Configuring MCE.....	895
8.9.10 Example for Configuring PBR to an LSP for VPN Packets.....	905
8.9.11 Example for Configuring HoVPN.....	911
8.9.12 Example for Configuring an OSPF Sham Link.....	920
8.9.13 Example for Configuring BGP AS Number Substitution.....	930
8.9.14 Example for Configuring the BGP SoO Attribute.....	936
8.9.15 Example for Configuring CE Dual-homing.....	945
8.9.16 Example for Configuring VPN FRR.....	960

8.9.17 Example for Configuring IP FRR for VPN Routes.....	969
8.9.18 Example for Configuring VPN GR.....	974
8.9.19 Example for Configuring Double RRs to Optimize the VPN Backbone Layer.....	985
8.9.20 Example for Connecting a VPN to the Internet.....	995
8.9.21 Example for Configuring BGP/MPLS IP VPN to Use a GRE Tunnel.....	1003
8.9.22 Example for Configuring L3VPN Using LDP Signaling over GRE.....	1011
8.9.23 Example for Configuring L3VPN with LDP Signals Carried by DSVPN.....	1020
8.9.24 Example for Configuring L3VPN with LDP Signals Carried by DSVPN and Protected by IPSec.....	1034
8.9.25 Example for Configuring a Tunnel Policy for an L3VPN.....	1052
8.10 FAQ About BGP/MPLS IP VPN.....	1063
8.10.1 Why Routes Cannot Be Imported When AS Numbers on the BGP/MPLS IP VPN Are the Same?.....	1064
8.11 References for BGP/MPLS IP VPN.....	1064
9 MCE IPv6 Configuration.....	1065
9.1 Overview of MCE IPv6.....	1065
9.2 Licensing Requirements and Limitations for MCE IPv6.....	1067
9.3 Configuring an MCE Device.....	1067
9.3.1 Configuring a VPN Instance.....	1067
9.3.2 Configure Route Exchange Between an MCE Device and VPN Sites.....	1069
9.3.3 Configure Route Exchange Between an MCE Device and a PE Device.....	1073
9.3.4 Verifying the MCE Configuration.....	1076
9.4 Configuration Examples for MCE IPv6.....	1076
9.4.1 Example for Configuring an MCE IPv6 Device.....	1076
10 EVPN Configuration.....	1093
10.1 Overview of EVPN.....	1093
10.2 Understanding EVPN.....	1094
10.2.1 Implementation.....	1094
10.3 Application Scenarios for EVPN.....	1097
10.3.1 EVPN Applications.....	1097
10.4 Licensing Requirements and Limitations for EVPN.....	1098
10.5 Configuring EVPN Functions.....	1099
10.5.1 Before You Start.....	1099
10.5.2 Configuring a VPN Instance.....	1100
10.5.3 Binding an Interface to a VPN Instance.....	1101
10.5.4 Configuring an EVPN BGP Peer Relationship.....	1102
10.5.5 Verifying the EVPN Configuration.....	1103
10.6 Maintaining EVPN.....	1106
10.6.1 Configuring EVPN BGP Soft Reset.....	1106
10.6.2 Resetting EVPN BGP Connections.....	1106
10.7 Configuration Examples for EVPN.....	1107
10.7.1 Example for Dynamically Establishing a VXLAN Tunnel in BGP EVPN Mode to Implement Communication Between Users in Different Network Segments.....	1107
10.8 References for EVPN.....	1114

11 VLL Configuration.....	1115
11.1 Overview of VLL.....	1116
11.2 Understanding VLL.....	1117
11.2.1 Implementation.....	1117
11.2.2 VLL Modes.....	1120
11.2.2.1 VLL in CCC Mode.....	1120
11.2.2.2 VLL in Martini Mode.....	1121
11.2.2.3 VLL in SVC Mode.....	1126
11.2.2.4 Comparison of VLL Modes.....	1127
11.2.3 Inter-AS VLL.....	1128
11.2.4 VLL FRR.....	1129
11.3 Application Scenarios for VLL.....	1132
11.3.1 Point-to-Point Layer 2 Connection Between Sites in Different Cities.....	1132
11.3.2 Multi-service Transparent Transmission over PWs on a MAN.....	1132
11.4 Summary of VLL Configuration Tasks.....	1133
11.5 Licensing Requirements and Limitations for VLL.....	1135
11.6 Default Settings for VLL.....	1135
11.7 Configuring VLL.....	1135
11.7.1 Configuring the CCC VLL.....	1136
11.7.2 Configuring the Martini VLL.....	1136
11.7.3 Configuring the SVC VLL.....	1138
11.7.4 Configuring Inter-AS VLL.....	1139
11.7.5 Configuring VLL FRR.....	1140
11.7.5.1 Configuring Primary and Secondary PWs.....	1141
11.7.5.2 (Optional) Configuring Fast Fault Notification - OAM Mapping.....	1142
11.7.5.3 (Optional) Configuring BFD for PW.....	1143
11.7.5.4 (Optional) Configuring a Revertive Switchover Policy.....	1143
11.7.5.5 Verifying the VLL FRR Configuration.....	1144
11.7.6 Configuring the Access of VLL to L3VPN.....	1145
11.7.6.1 Before You Start.....	1145
11.7.6.2 Creating an L2VE Interface.....	1147
11.7.6.3 Creating an L3VE Interface.....	1147
11.7.6.4 Associating the L2VE Interface with a VLL.....	1148
11.7.6.5 Configuring the Access of a User to L3VPN.....	1149
11.7.6.6 Verifying the configuration of Martini VLL to Access L3VPN.....	1150
11.7.7 Configuring and Applying a Tunnel Policy.....	1150
11.7.8 Configuring the Alarm Report Function.....	1153
11.8 Maintaining VLL.....	1154
11.8.1 Monitoring the Running Status of VLL.....	1154
11.8.2 Checking Connectivity of the VLL Network.....	1154
11.9 Configuration Examples for VLL.....	1156
11.9.1 Example for Configuring a Local CCC Connection.....	1156

11.9.2 Example for Configuring a VLL Connection in SVC Mode.....	1158
11.9.3 Example for Configuring a VLL Connection in Martini Mode.....	1163
11.9.4 Example for Configuring Inter-AS Martini VLL (Option A).....	1169
11.9.5 Example for Configuring Martini VLL FRR (Asymmetrically Connected CEs).....	1176
11.9.6 Example for Configuring VLL to Use a GRE Tunnel.....	1192
11.9.7 Example for Configuring a VLL Using an MPLS TE Tunnel.....	1198
11.10 Troubleshooting VLL.....	1206
11.10.1 The VC of a Martini VLL Connection Cannot Go Up.....	1206
11.11 References for VLL.....	1208
12 PWE3 Configuration.....	1209
12.1 Overview of PWE3.....	1210
12.2 Relationship Between PWE3 and L2VPN.....	1211
12.2.1 Extensions to the Control Plane.....	1211
12.2.2 Extensions at the Data Plane.....	1211
12.3 Understanding PWE3.....	1212
12.3.1 Implementation.....	1212
12.3.2 Control Word.....	1216
12.3.3 VCCV.....	1217
12.3.4 PWE3 FRR.....	1218
12.3.5 Inter-AS Technology.....	1220
12.4 Application Scenarios for PWE3.....	1221
12.4.1 PWE3 Carrying Enterprise Leased Line Services on a MAN.....	1221
12.5 Summary of PWE3 Configuration Tasks.....	1222
12.6 Licensing Requirements and Limitations for PWE3.....	1224
12.7 Default Settings for PWE3.....	1225
12.8 Configuring PWE3.....	1225
12.8.1 Configuring a Static PW.....	1225
12.8.1.1 Enabling MPLS L2VPN.....	1226
12.8.1.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template	1226
12.8.1.3 Creating a Static PW.....	1227
12.8.1.4 Verifying the Static PW Configuration.....	1228
12.8.2 Configuring a Dynamic PW.....	1229
12.8.2.1 Enabling MPLS L2VPN.....	1229
12.8.2.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template	1230
12.8.2.3 Creating a Dynamic PW.....	1231
12.8.2.4 Verifying the Dynamic PW Configuration.....	1232
12.8.3 Configuring PW Switching.....	1232
12.8.4 Configuring TDM PWE3.....	1234
12.8.4.1 Configuring an AC Interface to Transparently Transmit TDM Cells.....	1235
12.8.4.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template.....	1236
12.8.4.3 Configuring PW.....	1238
12.8.4.4 Verifying the TDM PWE3 Configuration.....	1240

12.8.5 Configuring Static BFD for PWs.....	1241
12.8.5.1 Enabling BFD Globally.....	1241
12.8.5.2 Configuring BFD for PWs.....	1242
12.8.5.3 Verifying the Configuration of Static BFD for PWs.....	1243
12.8.6 Configuring PWE3 FRR.....	1243
12.8.6.1 Configuring Primary and Secondary PWs.....	1244
12.8.6.2 (Optional) Configuring Fast Fault Notification - OAM Mapping.....	1245
12.8.6.3 (Optional) Configuring BFD for PW.....	1246
12.8.6.4 (Optional) Configuring a Revertive Switchover Policy.....	1246
12.8.6.5 Verifying the PWE3 FRR Configuration.....	1247
12.8.7 Configuring Inter-AS PWE3.....	1248
12.8.8 Configuring and Applying a Tunnel Policy.....	1249
12.9 Maintaining PWE3.....	1252
12.9.1 Verifying Connectivity of a PW.....	1252
12.9.2 Locating a Fault on a PW.....	1254
12.10 Configuration Examples for PWE3.....	1255
12.10.1 Example for Configuring a Dynamic Single-Segment PW.....	1255
12.10.2 Example for Configuring a Static Multi-Segment PW.....	1260
12.10.3 Example for Configuring a Dynamic Multi-Segment PW.....	1267
12.10.4 Example for Configuring a Mixed Multi-Segment PW.....	1275
12.10.5 Example for Configuring Inter-AS PWE3 Option A.....	1283
12.10.6 Example for Configuring TDM PWE3 (Using the 8E1T1-M Interface Card).....	1289
12.10.7 Example for Configuring TDM PWE3 (Using the 8SA interface card).....	1297
12.11 References for PWE3.....	1303
13 VPLS Configuration.....	1305
13.1 Overview of VPLS.....	1306
13.2 Understanding VPLS.....	1307
13.2.1 Implementation.....	1307
13.2.2 PW Signaling Protocols.....	1311
13.2.3 Packet Encapsulation.....	1312
13.2.4 MAC Address Management.....	1316
13.2.5 Loop Prevention.....	1319
13.2.6 Inter-AS VPLS.....	1319
13.3 Application Scenarios for VPLS.....	1321
13.3.1 VPLS Application in Individual Services.....	1321
13.3.2 VPLS Application in Enterprise Services.....	1323
13.4 Licensing Requirements and Limitations for VPLS.....	1325
13.5 Default Settings for VPLS.....	1325
13.6 Configuring Martini VPLS.....	1326
13.6.1 Creating a VSI and Configuring LDP Signaling.....	1326
13.6.2 Binding VSIs to AC Interfaces.....	1328
13.6.3 Verifying the Martini VPLS Configuration.....	1330

13.7 (Optional) Configuring Inter-AS Martini VPLS.....	1330
13.7.1 Configuring Inter-AS Martini VPLS in OptionA Mode.....	1331
13.7.2 Configuring Inter-AS Martini VPLS in OptionC Mode.....	1332
13.8 (Optional) Setting Related Parameters for a VSI.....	1335
13.8.1 Configuring a PE to Send MAC Withdraw Messages to Remove MAC Address Entries.....	1335
13.8.2 Configuring MAC Withdraw Loop Detection.....	1336
13.8.3 Configuring MAC Address Learning.....	1336
13.8.4 Configuring a VSI to Ignore the AC Status.....	1338
13.9 Maintaining VPLS.....	1339
13.9.1 Collecting Traffic Statistics on a VPLS PW.....	1339
13.9.2 Clearing the Traffic Statistics.....	1340
13.9.3 Checking Traffic Statistics on a VPLS PW.....	1340
13.9.4 Enabling or Disabling VSI.....	1341
13.9.5 Clearing MAC Address Entries.....	1341
13.9.6 Checking Connectivity of the VPLS Network.....	1342
13.9.7 Configuring the Upper and Lower Alarm Thresholds for VPLS VCs.....	1342
13.9.8 Checking MPLS L2VPN Usage Information.....	1343
13.10 Configuration Examples for VPLS.....	1343
13.10.1 Example for Configuring Martini VPLS.....	1343
13.10.2 Example for Configuring Inter-AS Martini VPLS in OptionA Mode.....	1350
13.11 Troubleshooting VPLS.....	1357
13.11.1 VSI Cannot Go Up in Martini VPLS Mode.....	1357
13.12 References for VPLS.....	1358
14 VXLAN Configuration.....	1359
14.1 Overview of VXLANs.....	1359
14.2 Understanding VXLANs.....	1361
14.2.1 VXLAN Network Architecture.....	1361
14.2.2 Packet Encapsulation Format.....	1365
14.2.3 VXLAN Implementation.....	1367
14.2.3.1 Packet Identification.....	1367
14.2.3.2 Tunnel Establishment.....	1371
14.2.3.3 Packet Forwarding.....	1372
14.3 Application Scenario.....	1379
14.4 Licensing Requirements and Limitations for VXLAN.....	1380
14.5 Configuring VXLAN (in Static Mode).....	1381
14.5.1 Configuring Deployment Mode for VXLAN Access Service.....	1383
14.5.2 Configuring a VXLAN Tunnel.....	1385
14.5.3 Configuring a Layer 3 VXLAN Gateway.....	1387
14.5.4 (Optional) Configuring Static ARP Entries.....	1388
14.5.5 (Optional) Configuring a Static MAC Address Entry.....	1389
14.5.6 Verifying the VXLAN Configuration in Centralized Gateway Mode Using Static Mode.....	1390
14.6 Configuring VXLAN (in BGP EVPN Mode).....	1390

14.6.1 Configuring Deployment Mode for VXLAN Access Service.....	1391
14.6.2 Configuring a VXLAN Tunnel.....	1393
14.6.3 Configuring a Layer 3 VXLAN Gateway.....	1395
14.6.4 Checking the Configuration.....	1397
14.7 Configuration Examples for VXLANs.....	1397
14.7.1 Example for Configuring Communication Within a Network Segment Through a VXLAN Tunnel.....	1397
14.7.2 Example for Configuring a Layer 3 VXLAN Gateway to Enable Communication Between Users in Different Network Segments.....	1402
14.7.3 Example for Dynamically Establishing a VXLAN Tunnel in BGP EVPN Mode to Implement Communication Between Users in Different Network Segments.....	1408
14.7.4 Example for Configuring the Headquarters and Branch to Communicate Using VXLAN over IPSec Tunnels	1415
14.8 References for VXLANs.....	1426
14.9 Further Reading.....	1426
14.9.1 Server Virtualization.....	1426
14.9.2 Large Layer 2 Network.....	1427

1 L2TP Configuration

About This Chapter

L2TP connections are established between the LAC and LNS in several application scenarios so that remote users can access resources in the headquarters using L2TP tunnels.

[1.1 Overview of L2TP](#)

This section describes the definition and functions of L2TP.

[1.2 Understanding L2TP](#)

This section describes the implementation of L2TP.

[1.3 Application Scenarios for L2TP](#)

This section describes the application scenarios for L2TP.

[1.4 Licensing Requirements and Limitations for L2TP](#)

This section describes the notes that need to be taken during L2TP configuration.

[1.5 Default Settings for L2TP](#)

This section provides the default settings for L2TP.

[1.6 Configuring L2TP](#)

This section describes the procedures for configuring L2TP functions.

[1.7 Maintaining L2TP](#)

This section describes how to disconnect an L2TP tunnel forcibly and monitor the running status of L2TP.

[1.8 Configuration Examples for L2TP](#)

This section provides L2TP configuration examples.

[1.9 Troubleshooting L2TP](#)

This section describes common faults caused by incorrect L2TP configurations.

[1.10 FAQ About L2TP](#)

This section describes the FAQ about L2TP.

[1.11 References for L2TP](#)

This section lists references for L2TP.

1.1 Overview of L2TP

This section describes the definition and functions of L2TP.

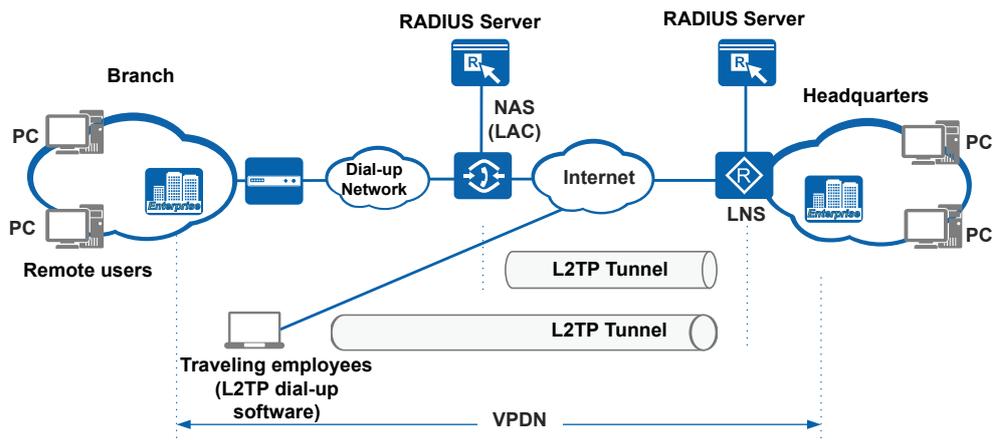
Definition

The Layer 2 Tunneling Protocol (L2TP) is a Virtual Private Dial-up Network (VPDN) tunneling protocol and expands applications of the Point-to-Point Protocol (PPP) to allow remote dial-up users to access the network of an enterprise headquarters.

Based on PPP negotiation, L2TP sets up tunnels between branch users and enterprise headquarters over the dial-up network, so that remote users can access the headquarters network. The PPP over Ethernet (PPPoE) technology further expands the application scale of L2TP and can establish L2TP tunnels between remote users and the headquarters over the Ethernet and Internet.

Figure 1-1 shows a typical networking for constructing a VPDN network using L2TP.

Figure 1-1 Typical networking of L2TP



Purpose

As enterprises develop and services increase, many branches are set up in different locations, and some staff often go on business trips. They require fast, secure, and reliable network connections with the headquarters. On traditional dial-up networks, they use phone lines leased by the Internet Service Provider (ISP) and apply for a dial string or IP addresses from the ISP. This results in high costs. Besides, leased lines cannot provide services for remote users especially the staff on business trips. VPDN, a dial-up network based VPN, is introduced to make a better use of dial-up networks to ease access of remote users. VPDN establishes a point-to-point virtual link between remote users and the headquarters gateway.

VPDN provides the following tunneling technologies:

- Point-to-point tunneling protocol (PPTP)
- Layer 2 forwarding (L2F)
- Layer 2 Tunneling Protocol (L2TP)

L2TP combines advantages of PPTP and L2F and is widely accepted. L2TP enables an individual or a small number of remote users to access the internal network of an enterprise over the public network.

Benefits

L2TP encapsulates PPP packets to transmit private data of an enterprise through virtual links established over the public network. This releases the enterprise from renting expensive physical lines. The enterprise only needs to manage remote access users and users on the private network, reducing maintenance cost due to simplified network architecture.

L2TP provides convenient, secure, and reliable access services for remote users, and brings the following benefits:

- Flexible identity authentication and high security
 - L2TP uses PPP security features such as PAP and CHAP to authenticate user identity.
 - L2TP allows control messages to be transmitted in cipher text and supports tunnel authentication.
 - L2TP works with Internet Protocol Security (IPSec) to ensure high security data transmission, although it does not encrypt data to be transmitted.
- Multi-protocol transmission

L2TP transmits PPP frames, which can be used to encapsulate packets of multiple network layer protocols. Therefore, L2TP can be used on IP networks, Frame Relay (FR) permanent virtual circuit (PVCs), X.25 virtual circuits (VCs), or ATM VCs.
- Remote Authentication Dial-in User Service (RADIUS) authentication

L2TP provides two authentication methods to manage access users: local authentication and RADIUS authentication using the user name and password sent by a dial-up user.
- Internal address allocation

The enterprise headquarters gateway enabled with L2TP dynamically allocates private addresses to remote users.
- Reliability

L2TP supports LNS backup. When the primary LNS is unreachable, an LAC can establish a new connection with a secondary LNS. This enhances reliability of VPN services.

1.2 Understanding L2TP

This section describes the implementation of L2TP.

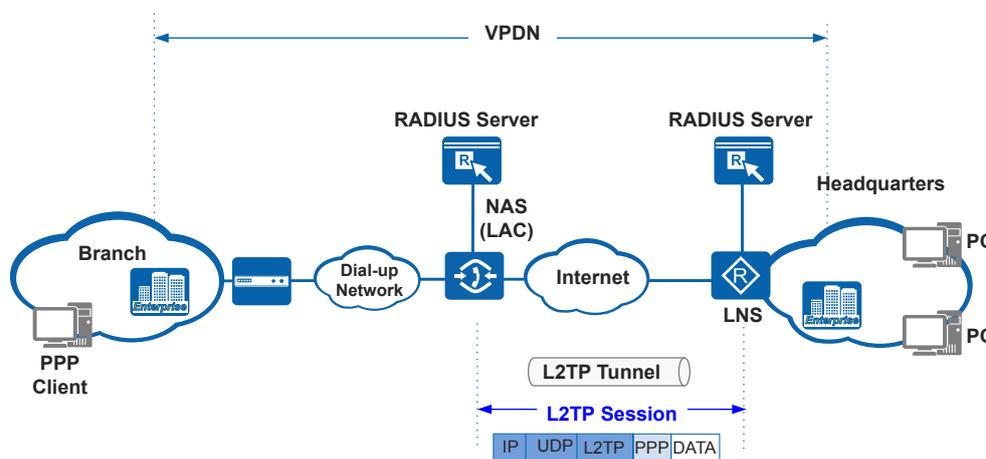
1.2.1 Concepts

Figure 1-2 shows a typical L2TP networking. Basic concepts related to L2TP are listed as follows:

- **VPDN**
- **PPP Terminal**
- **NAS**

- **LAC**
- **LNS**
- **Tunnel and Session**

Figure 1-2 L2TP networking diagram



VPDN

VPDN, as a VPN that carries PPP packets, provides access services for enterprise users, small-scale ISPs, and traveling employees.

The **PPP terminal** accesses a dial-up network and dials up to the **NAS**. After receiving a PPP packet, the NAS implements L2TP encapsulation, and forwards the packet with an outer IP header over the public network to the **LNS**. After receiving the packet, the LNS decapsulates the packet to obtain the original PPP packet, implementing transparent transmission of the PPP packet over the public network. In this manner, a VPDN connection is set up between the PPP terminal and the LNS.

As the Ethernet becomes popular, PPP terminals can be used on the traditional dial-up networks and can also connect to the **LAC** over the Ethernet using the PPPoE technology.

PPP Terminal

In L2TP applications, PPP terminals are the devices that initiate dial-up calls and perform PPP encapsulation on data. For example, the PPP terminal can be a remote PC or a gateway in the branch.

NAS

A network access server (NAS) is maintained by the ISP and connected to a dial-up network. It is an access point geographically closest to the PPP terminal. The NAS works on a traditional dial-up network to provide VPDN services for remote dial-up users to set up tunnel connections with the enterprise headquarters network.

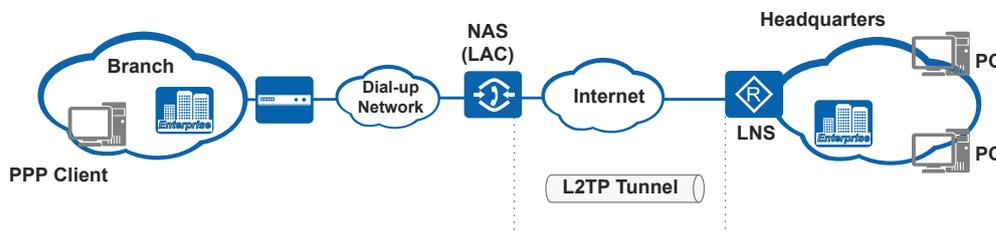
LAC

An L2TP access concentrator (LAC) provides PPP and L2TP processing capabilities on the packet switched network. The LAC establishes an L2TP connection with the L2TP network

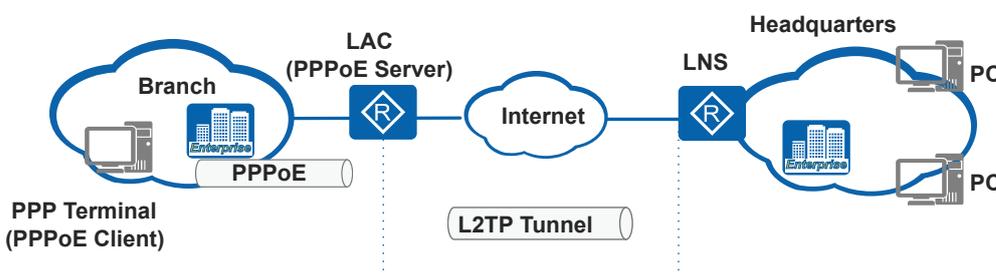
server (LNS) based on the user name or domain name in PPP packets so that PPP frames can be transmitted to the LNS.

The LAC can be deployed on different devices on various networks.

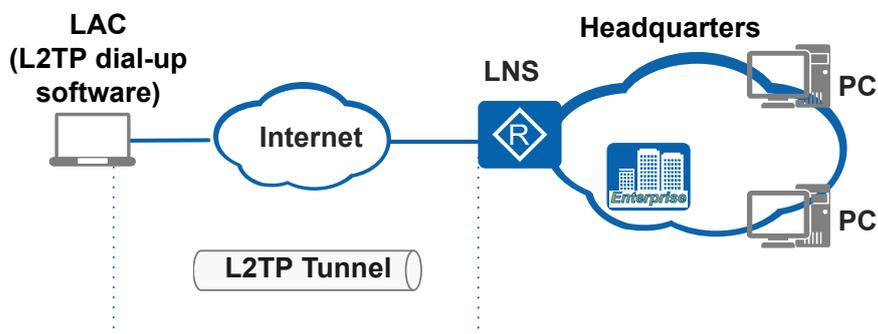
- On a traditional dial-up network, the ISP usually deploys an LAC on the NAS.



- On an Ethernet in an enterprise branch, an LAC is deployed on the gateway for PPP terminals and also functions as a PPPoE server.



- A traveling employee uses a PC to access the Internet. The L2TP dial-up software installed on the PC functions as the LAC.



An LAC can establish different L2TP tunnels to isolate data flows. That is, multiple VPDN connections can be set up on the LAC.

An LAC transmits data between the LNS and PPP terminal. The LAC encapsulates data received from the PPP terminal based on L2TP, sends data to the LNS, decapsulates the data received from the LNS, and sends it to the PPP terminal.

LNS

PPP sessions are initiated by user devices and received by the LNS. After being authenticated by the LNS, remote users successfully set up PPP sessions with the LNS and can access resources in the enterprise headquarters. As the other endpoint of an L2TP tunnel, the LNS is a peer device of the LAC, and sets up an L2TP tunnel with the LAC. Additionally, the LNS is the logical termination point of a PPP session; therefore, the PPP client (user device) and the LNS establish a virtual point-to-point link.

The LNS is located at the border between the headquarters' private network and the public network, and is often used as the gateway of the enterprise headquarters. In addition, the LNS

provides the network address translation (NAT) function to translate private IP addresses in the enterprise headquarters network into public IP addresses.

Tunnel and Session

There are two types of connections during the L2TP tunnel establishment between the LAC and LNS.

- Tunnel connection
Multiple L2TP tunnels can be set up between an LNS and an LAC. A tunnel consists of one or more sessions.
- Session connection
An L2TP session can be set up only after a tunnel is created successfully, and represents a PPP session over the tunnel.

1.2.2 L2TP Implementation

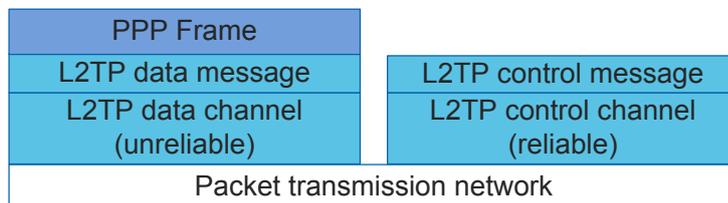
L2TP Architecture

The L2TP protocol defines two message types: control messages and data messages that are transmitted between an LAC and an LNS. L2TP uses these two types of messages to expand PPP applications.

- Control message
Control messages are used to establish, maintain, and tear down tunnels and sessions. L2TP uses retransmission and periodical tunnel connectivity check mechanisms to ensure reliable transmission of control messages. L2TP also supports flow control and congestion control on control messages.
- Data message
Data messages are used to encapsulate PPP frames and are transmitted over tunnels. Data messages are transmitted over an unreliable channel without flow control, congestion control, and retransmission mechanisms.

Figure 1-3 illustrates the relationship between PPP packets, control messages, and data messages.

Figure 1-3 L2TP architecture



Control messages encapsulated with L2TP headers are transmitted over a reliable L2TP control channel on an IP network.

Data messages carrying PPP frames are transmitted over an unreliable data channel. PPP frames are encapsulated using L2TP and then transmitted over an IP network.

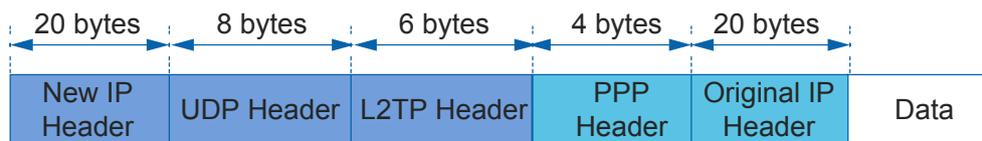
The well-known UDP port for L2TP is 1701, which is only used in initial stage of tunnel setup. The L2TP tunnel initiator randomly selects an idle port to forward packets to port 1701

of the receiver. After receiving the packets, the receiver randomly selects an idle port to forward packets to the port selected by the initiator. Both ends use the selected ports to communicate until the tunnel is disconnected.

L2TP Packet Structure

Figure 1-4 shows the format of an L2TP packet, which is generated by encapsulating a PPP frame initiated by a remote dial-up user.

Figure 1-4 Format of an L2TP packet



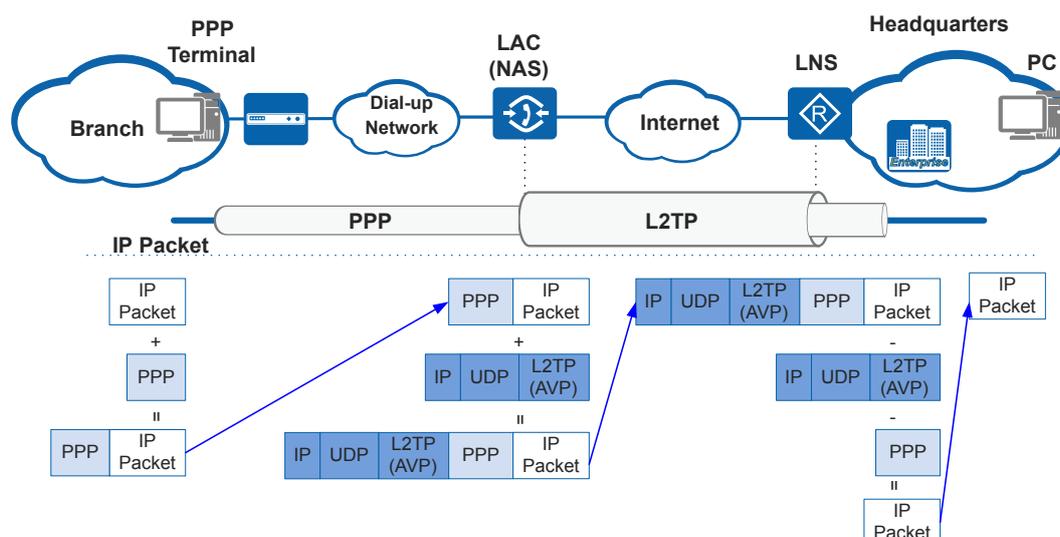
After L2TP encapsulation, an L2TP packet has 38 bytes more than the original packet. (If an L2TP packet carries sequence number information, it has 42 bytes more than the original packet.) If the length of the encapsulated packets exceeds the MTU of the outbound interface, the device must be able to fragment the IP packets because L2TP does not support packet fragmentation. The receiver end reassembles fragmented packets into L2TP packets.

L2TP Packet Encapsulation

As an expansion to PPP, L2TP allows PPP packets to be transmitted through tunnels over the public network.

If only PPP is deployed on the network, dial-up calls initiated by PPP terminals can only reach the edge node NAS of the dial-up network. The NAS is the termination point of PPP sessions. When L2TP is deployed, PPP packets can be transparently transmitted over the public network and reach the LNS in the enterprise headquarters. In this case, the LNS is the termination point of PPP sessions.

Figure 1-5 L2TP packet encapsulation



As shown in **Figure 1-5**, packets are sent from a branch to the headquarters following the process as follows:

1. PPP terminal: encapsulates IP packets with PPP at the link layer and sends the packets.
2. LAC: receives PPP packets and determines whether access users are VPDN users based on user names or domain names carried in the packets.
 - If they are VPDN users, the LAC adds L2TP headers to PPP packets and then adds UDP and IP headers to the packets based on the public network address of the LNS. The outer layer of the encapsulated packets is the IP address of the public network address. The packets are forwarded over the public network to the LNS.
 - If they are non-VPDN users, the LAC decapsulates PPP packets. In this case, the LAC is the termination point of PPP sessions.
3. LNS: receives L2TP packets and removes IP, L2TP, and PPP headers to obtain IP packets sent by PPP terminals. The LNS searches the routing table for the destination host in the headquarters based on the destination address contained in the packets.

When the destination host sends response packets to the branch device, the LNS searches the routing table for the outbound interface and encapsulated the packet with L2TP in a similar process.

L2TP Packet Transmission

L2TP tunnel connections and session connections must be set up before PPP packets can be transmitted. L2TP connections are initiated for the first time according to the following procedure:

1. Setting an L2TP tunnel connection
After receiving a PPP negotiation request from a remote user, the LAC initiates an L2TP connection request to the LNS. The LAC and LNS exchange control messages to negotiate the tunnel ID and tunnel authentication information. After negotiation succeeds, an L2TP tunnel is set up and it is identified by a tunnel ID.
2. Setting an L2TP session connection
After an L2TP tunnel is set up, the LAC and LNS exchange control messages to negotiate the session ID. The L2TP session carries LCP negotiation information and authentication information. After authenticating such information, the LNS informs the LAC that a session is set up. An L2TP session connection is identified by a session ID.
3. Transmitting PPP packets
After an L2TP session connection is set up successfully, the PPP terminal sends data packets to the LAC. The LAC encapsulates the packets based on the tunnel ID and session ID and sends the packets to the LNS. The LNS decapsulates the packets and sends the packets to the destination host by searching for the host address in the routing table.

1.2.3 Working Procedure

VPDN connections are set up between the remote user and LNS. ISPs deploy the NAS that is geographically closest to the remote user as the LAC. L2TP tunnel connections are set up between the LAC and LNS.

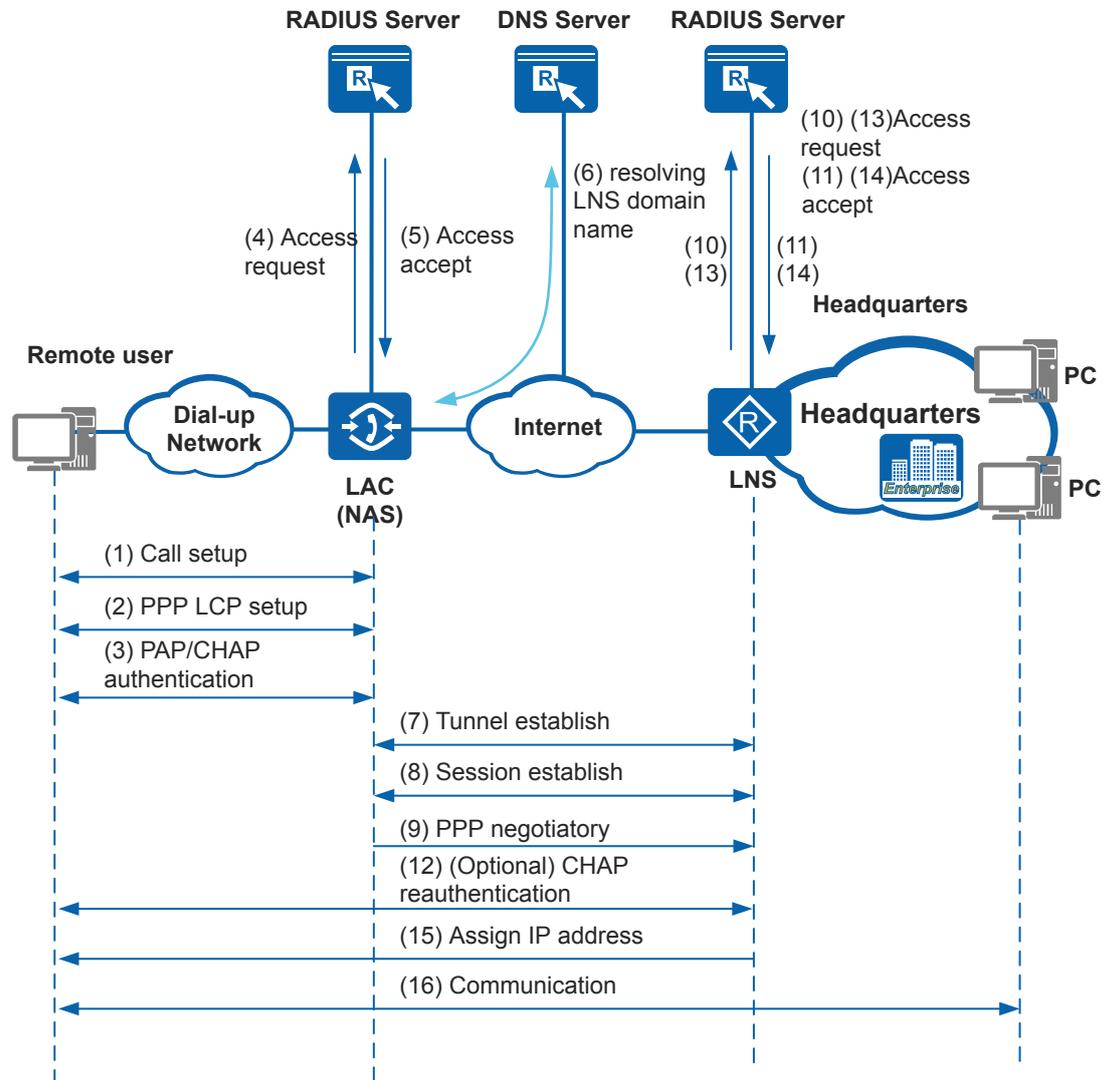
1. Remote users dial up on a PSTN or an ISDN to initiate PPP connections to a local NAS deployed by an ISP.

2. The NAS accepts calls from remote users and performs PPP negotiation.
3. As the LAC, the NAS determines whether remote users are VPDN users based on user names or domain names. If remote users are VPDN users, the L2TP module encapsulates PPP packets from them and sends the packets through the L2TP tunnel to the LNS. If remote users are not VPDN users, PPP packets from them are processed and forwarded normally.
4. Upon receiving call connection requests sent through the L2TP tunnel, the LNS authenticates remote users and assigns and sends IP addresses to remote users.
5. Remote users obtain IP addresses and send packets to hosts in the headquarters to communicate.
6. The LNS receives packets transmitted through the tunnel and forwards the packets to destination hosts according to the routing table.

After L2TP encapsulation, remote users set up point-to-point connections to the LNS, and the LAC and Internet are transparent to users. The LAC and LNS use remote authentication.

Figure 1-6 shows the L2TP call setup procedure in details.

Figure 1-6 L2TP call setup procedure



1. The PC of a remote user initiates a request for a call connection to the LAC.
2. The PC and the LAC perform PPP LCP negotiation.
3. The LAC authenticates the PC user using the Challenge Handshake Authentication Protocol (CHAP).
4. The LAC sends authentication information including the user name and password to the RADIUS server for authentication.
5. After authenticating the user, the RADIUS server sends the authentication result to the user.
6. If the LNS domain name is specified on the LAC, the LAC checks whether the LNS domain name is parsed. If the LNS domain name is not parsed, the LAC requests the corresponding IP address based on the domain name from the DNS server. If an IP address is parsed from the domain name, the tunnel setup process is triggered. If no IP address is parsed from the domain name, the user cannot go online.

7. An L2TP tunnel connection is set up between the LAC and LNS.
8. An L2TP session connection is set up between the LAC and LNS.
9. The LNS processes PPP negotiation information contained in the session connection request.
10. The LNS sends an access request to its RADIUS server for authentication.
11. The RADIUS server sends a response packet after the authentication succeeds. If the Frame-IP and Frame-Route attributes or address pool name is specified on the RADIUS server, the response packet carries the Frame-IP and Frame-Route attributes or the specified address pool name.
12. (Optional) The LNS performs secondary CHAP authentication on the remote user.
13. The LNS sends secondary authentication information to its RADIUS server for authentication.
14. The RADIUS server sends a response packet after the authentication succeeds.
15. The LNS saves the Frame-IP and Frame-Route attributes or the specified address pool name carried in the response packet. An L2TP connection is set up and the LNS assigns IP addresses to remote users.
16. The remote user can communicate with devices in the headquarters and the LNS functions as a gateway.

 **NOTE**

If you run step 12, step 13 and 14 are mandatory.

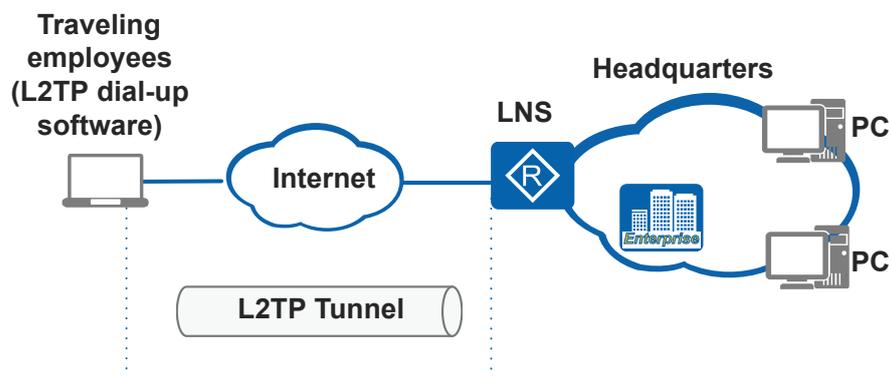
1.3 Application Scenarios for L2TP

This section describes the application scenarios for L2TP.

1.3.1 Client-Initiated L2TP Connection

Traveling employees need to communicate with employees in the headquarters and access the headquarters gateway through the Internet to use internal resources. However, the headquarters gateway cannot identify and manage access users. To solve this problem, configure the headquarters gateway as the LNS to establish a virtual point-to-point connection between the traveling employees and the headquarters gateway when the employees use the L2TP dial-up software on the PC to initiate L2TP connections. The LNS can authenticate access users and assign private network addresses. The LNS can use ACLs to manage users' access rights.

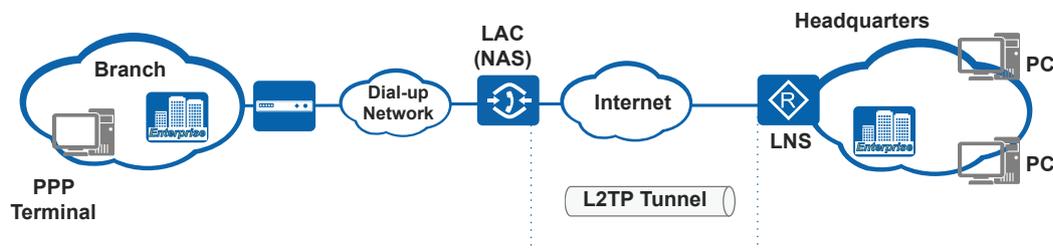
Figure 1-7 Client-initiated L2TP connection



1.3.2 LAC-Initiated L2TP Connection upon Receiving a Call Connection Request

An enterprise has a branch located in another city, and the branch network is a traditional dial-up network. Branch users need to establish VPDN connections with users in the headquarters. Therefore, the branch users apply for the L2TP service from the ISP. The ISP configures the NAS as the LAC to send call connection requests to the LNS through the Internet. The gateway in the headquarters is configured as the LNS to establish VPDN connections between the branch and the headquarters.

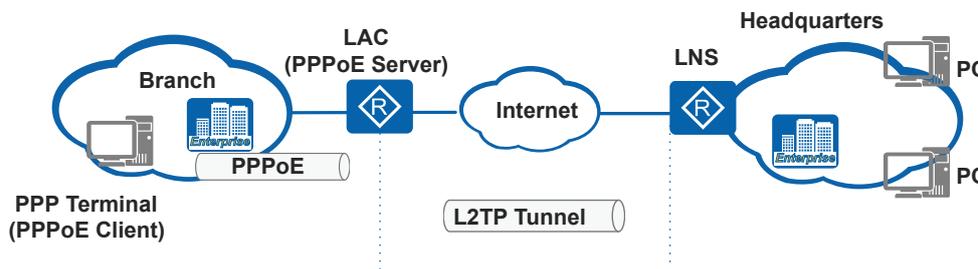
Figure 1-8 LAC-initiated L2TP connection upon receiving a call connection request



1.3.3 LAC-Initiated L2TP Connection upon Receiving a Call from a PPPoE User

An enterprise has a branch located in another city, and its branch network uses the Ethernet and has a gateway deployed, so branch hosts can access the Internet. Hosts in the headquarters need to communicate with hosts in the branch. You can use L2TP to configure the headquarter gateway as an LNS that manages access requests from branch hosts in a centralized manner. Dial-up data of a branch host cannot be transmitted directly over the Ethernet network. When PPPoE dial-up software is deployed on the branch host, the host functions as the PPPoE client, and the branch gateway functions as the PPPoE server and the LAC. Dial-up data can then be transmitted to the headquarters.

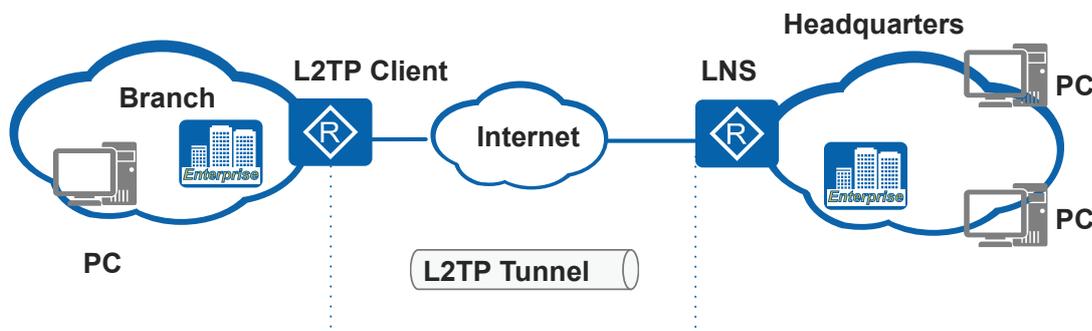
Figure 1-9 LAC-initiated L2TP connection upon receiving a call from a PPPoE user



1.3.4 L2TP Client-Initiated L2TP Connection

An enterprise has a branch located in another city, and its branch network uses the Ethernet and has a gateway deployed, so branch hosts can access the Internet. A VPN connection must be established between the headquarters gateway and the branch so that the headquarters can provide access services for users. Any user from the branch is allowed to access the headquarters gateway which only authenticates the branch gateway. Configure the headquarters gateway as the LNS and the branch gateway as the L2TP Client. Configure virtual users on the branch gateway to initiate L2TP connections to the headquarters. In L2TP Client-Initiated mode, virtual point-to-point connections are established between the L2TP Client and LNS. After receiving IP packets from branch users, the L2TP Client forwards the packets through a virtual dial-up interface and the L2TP tunnel to the LNS, which forwards the packets to the destination host.

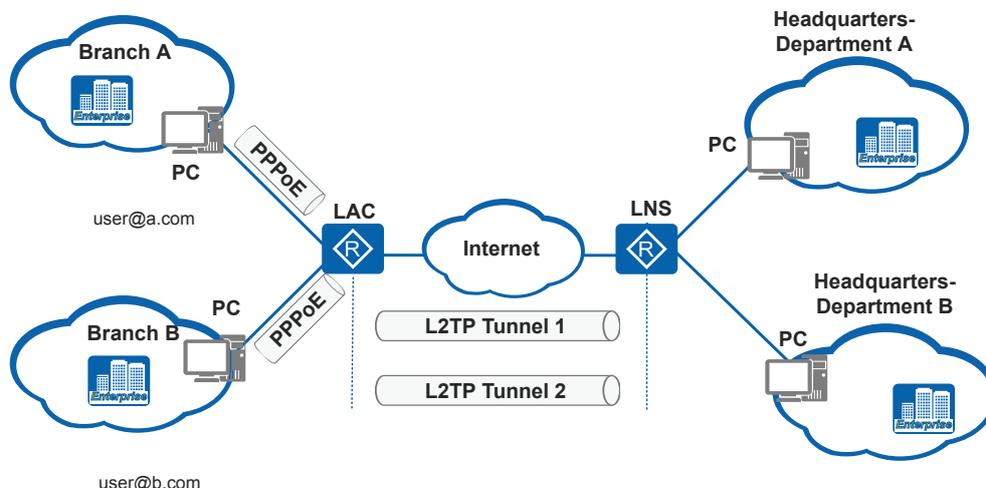
Figure 1-10 L2TP Client-Initiated L2TP connection



1.3.5 LAC-Initiated L2TP Connection When Users from Multiple Domains Are Connected

Different enterprise branches are allowed to access resources in different departments of the enterprise headquarters. The headquarters provides access services for branch staff. The headquarters establishes VPN connections with branches using L2TP. The branch gateway determines users based on domain names when there are many users, which facilitates VPN user management. Each branch uses a separate L2TP tunnel and obtains private addresses on different segments. Because source and destination addresses are allocated by the headquarters, you can configure an ACL to allow the headquarters to manage access rights of branches.

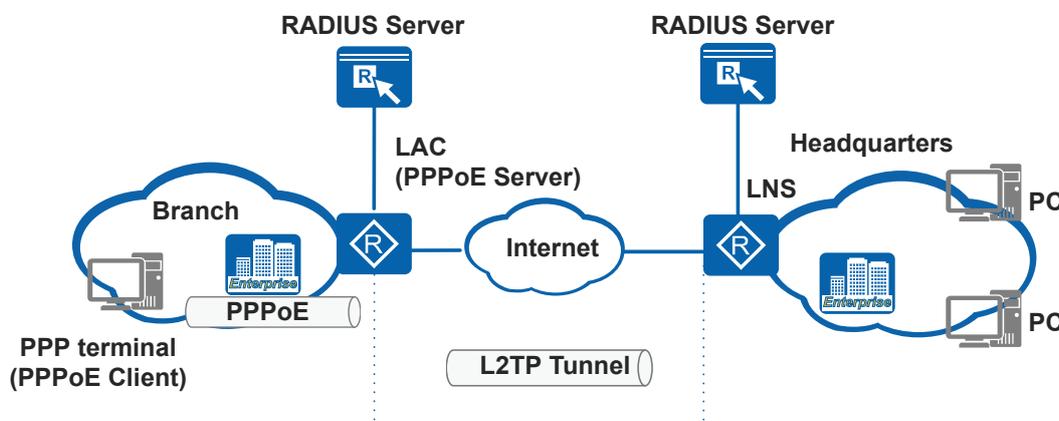
Figure 1-11 LAC-initiated L2TP connection when users from multiple domains are connected



1.3.6 Authenticating VPDN Users Using the RADIUS Server

An enterprise has a branch located in another city, and its branch network uses the Ethernet and has a gateway deployed, so branch hosts can access the Internet. A VPDN connection must be established between the headquarters gateway and the branch so that the headquarters can provide access services for users. Configure the branch gateway as the LAC and the headquarters gateway as the LNS to set up a VPDN connection. In addition, the branch gateway is configured as a PPPoE server to exchange PPP packets with the PPP client (branch user) over Ethernet. The LAC and LNS can authenticate branch users. However, when there are many users, a RADIUS server can be deployed to authenticate users remotely. The RADIUS server on the LAC side must support L2TP authentication. The RADIUS server determines whether remote users are VPDN users and sends the authentication result to the LAC.

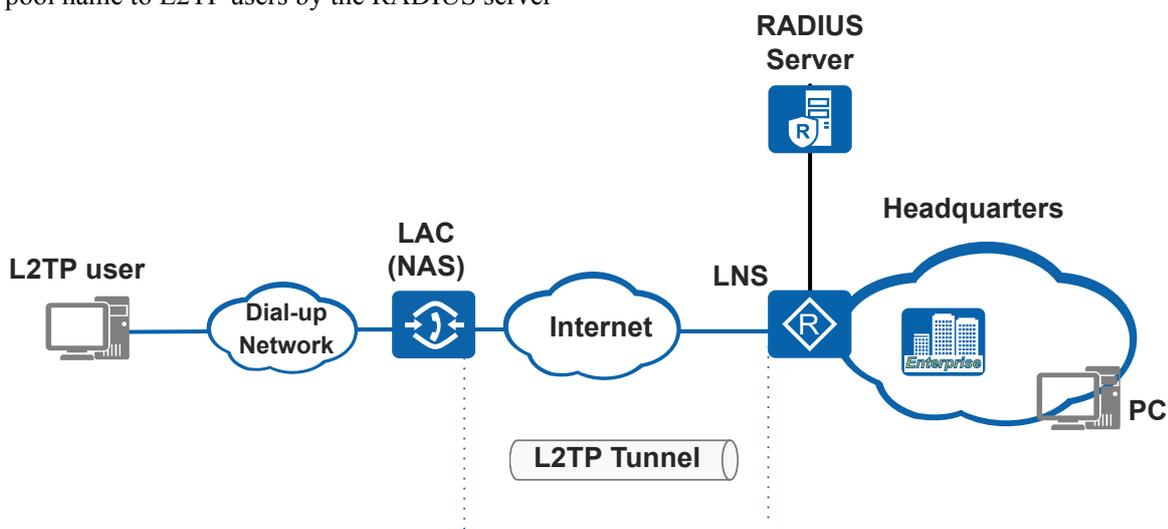
Figure 1-12 Authenticating VPDN users using the RADIUS server



1.3.7 Allocating the Frame-IP and Frame-Route Attributes and the Specified Address Pool Name to L2TP Users by the RADIUS Server

To facilitate management and reduce maintenance costs on user information, many enterprises use the RADIUS server to uniformly manage user information. By specifying the Frame-IP and Frame-Route attributes or the address pool name on the RADIUS server, the LNS allocates planned IP addresses to each user after the user goes online.

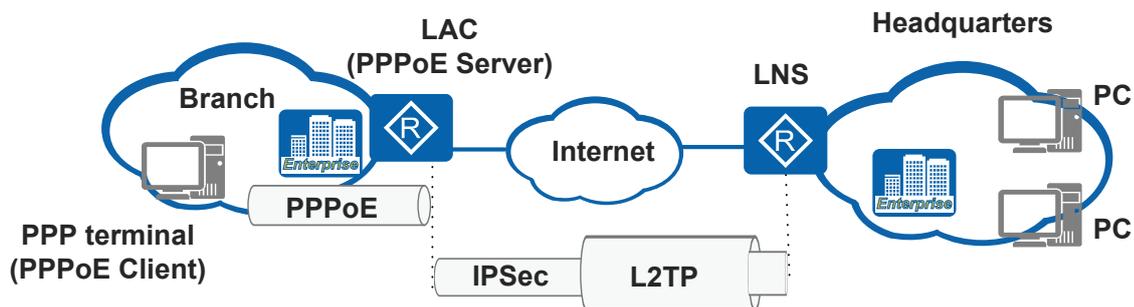
Figure 1-13 Allocating the Frame-IP and Frame-Route attributes and the specified address pool name to L2TP users by the RADIUS server



1.3.8 Setting Up a Secure Tunnel Connection Using L2TP over IPsec Encapsulation

An enterprise has a branch located in another city, and its branch network uses the Ethernet and has a gateway deployed, so branch hosts can access the Internet. L2TP is enabled on the headquarters and branch devices to set up VPDN connections so that branch users can access resources in the headquarters. When the enterprise requires high security, L2TP is not enough to provide protection on packet transmission. In this case, use L2TP together with IPsec to prevent data from being intercepted or attacked. Data packets are encapsulated using IPsec and L2TP on the LAC before being forwarded to the headquarters. Configure an IPsec policy on the headquarters gateway to decapsulate the packets. L2TP over IPsec encapsulation protects the original data packets and provides protection on access users as required.

Figure 1-14 Setting up a secure tunnel connection using L2TP over IPsec encapsulation



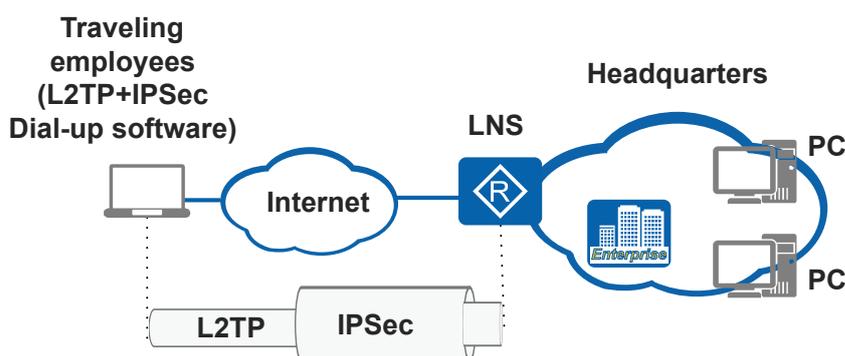
Related Topics

[Example for Configuring L2TP over IPsec for Remote Dial-Up Users to Connect to the Headquarters](#)

1.3.9 Setting Up a Secure Tunnel Connection Using IPsec over L2TP Encapsulation

Traveling employees need to communicate with staff in the headquarters. Therefore, L2TP is implemented to set up VPDN connections between them and the LNS in the headquarters authenticates access users. When traveling employees need to transmit confidential information to the headquarters, L2TP is not enough to provide protection on packet transmission. In this case, use L2TP together with IPsec to protect data transmission. Run the dial-up software on the traveling employee's PC. Data packets are encapsulated using L2TP and then IPsec on the PC before being forwarded to the headquarters. Configure an IPsec policy on the headquarters gateway to decapsulate the packets. IPsec over L2TP encapsulation protects all packets whose source and destination addresses are addresses of the LAC and LNS.

Figure 1-15 Setting up a secure tunnel connection using IPsec over L2TP encapsulation



Related Topics

[Example for Configuring L2TP over IPsec for Remote Dial-Up Users to Connect to the Headquarters](#)

1.4 Licensing Requirements and Limitations for L2TP

This section describes the notes that need to be taken during L2TP configuration.

Involved Network Elements

None

License Requirements

L2TP is a basic feature of the device and is not under license control.

Feature Limitations

None

1.5 Default Settings for L2TP

This section provides the default settings for L2TP.

Table 1-1 Default settings for L2TP

Parameter	Default Setting
l2tp enable	Disabled
tunnel authentication	Enabled
tunnel password	Not configured
tunnel name	Device name
tunnel avp-hidden	Disabled
mandatory-chap	Disabled
mandatory-lcp	Disabled
tunnel timer hello	60s

1.6 Configuring L2TP

This section describes the procedures for configuring L2TP functions.

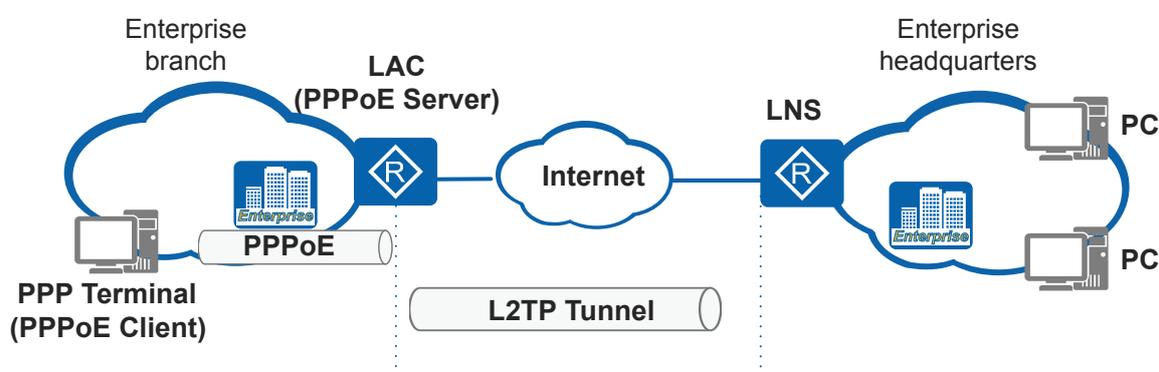
1.6.1 Configuring the LAC to Initiate Call-Triggered L2TP Connections

The LAC functions as a PPPoE server to authenticate call connecting requests from remote users, and initiates L2TP connections to the LNS based on the user information contained in the request packets.

Context

An enterprise has some branches located in other cities, and its branches use the Ethernet and have gateways deployed, so that branch hosts can access the Internet. Hosts at the headquarters need to communicate with hosts at branches. You can use L2TP to configure the headquarter gateway as an LNS that uniformly manages access requests from branch hosts. Dial-up data of a branch host cannot be transmitted directly over the Ethernet network. However, when PPPoE dial-up software is deployed on the branch host, the host functions as the PPPoE client, and the branch gateway functions as the PPPoE server and the LAC. Dial-up data can then be transmitted to the headquarters.

Figure 1-16 Networking diagram for the LAC to initiate an L2TP connection request on receiving a dial-up call



Prerequisite

A reachable route has been configured between the LNS and the LAC.

Configuration Process

[Table 1-2](#) shows the configuration process on the LAC. [Table 1-3](#) shows the configuration process on the LNS.

Table 1-2 Configuration process on the LAC

Configuration	Procedure	Description
Configure local or remote AAA authentication.	Configure local authentication.	Store user information, including the user name, password, and service type, on the local device.

Configuration	Procedure	Description
	Configure remote authentication.	Configure RADIUS server parameters to enable the RADIUS server to store user information, including the user name, password, and service type, and authenticate access users.
Configure the LAC to initiate an L2TP connection.	Enable L2TP.	Enable L2TP globally.
	Configure PPP negotiation.	Set the PPP negotiation mode to PAP or CHAP on the virtual tunnel interface. Assign an IP address for the VT interface to make the configuration take effect. Configure a PPPoE server on the physical interface at the user side.
	Create an L2TP group.	Configure L2TP parameters, including the LAC tunnel name and password, LNS address, and VPDN user name. You can also configure the Attribute Value Pair (AVP) data to be transmitted in cipher text, primary and secondary LNSs, and an interval for sending Hello packets.

Table 1-3 Configuration process on the LNS

Configuration	Procedure	Description
Configure local or remote AAA authentication.	Configure local authentication.	Store user information, including the user name, password, and service type, on the local device. You can also enable LCP renegotiation or mandatory CHAP authentication to implement second authentication on remote users.
	Configure remote authentication.	Configure RADIUS server parameters to enable the RADIUS server to store user information, including the user name, password, and service type, and authenticate access users. You can also enable LCP renegotiation or mandatory CHAP authentication to implement second authentication on remote users.
Configure the LNS to respond to the L2TP connection request.	Enable L2TP.	Enable L2TP globally.
	Configure an IP address pool.	Assign an IP address dynamically for the remote user after the user is authenticated. This step is not required when a static IP address is assigned to the remote user.

Configuration	Procedure	Description
	Configure PPP negotiation.	<p>Set PPP negotiation mode to PAP or CHAP on the VT interface.</p> <p>Configure an IP address and use this address as the private network gateway address of the L2TP tunnel.</p> <p>Import an IP address pool to dynamically allocate IP addresses for remote users.</p> <p>If mandatory CHAP authentication is configured, the PPP authentication mode must be CHAP.</p>
	Create an L2TP group.	<p>Configure L2TP parameters, including the LNS tunnel name and password, number of the VT, and LAC tunnel name.</p> <p>You can also configure the AVP data to be transmitted in cipher text, and an interval for sending Hello packets.</p>

1.6.1.1 Configuring AAA Authentication and Accounting

Context

The AAA provides authentication, authorization, and accounting security functions to manage access users and ensure secure connections. You can configure local or remote authentication on the LAC and LNS to authenticate remote users.

When users can access the Internet through only the LNS, you can configure the accounting function on the LNS to manage the online duration and traffic of the users.

The LAC checks the user name or domain name of the users to determine whether to establish a tunnel to the LNS. The user name and domain name are described as follows:

- **User name:** applies to the scenario where there are few users and each user is managed independently. In this scenario, each user exclusively occupies an L2TP tunnel.
 If remote users are authenticated based on user names, the device uses the default domain named **default**, the authentication scheme named **default**. The authentication scheme named **default** uses the default authentication mode **local**.
- **Domain name:** applies to the scenario where there are many access users and the users are managed uniformly. In this scenario, users with the same domain name occupy an L2TP tunnel.
 If remote users are authenticated based on the domain name, you need to configure a domain and an authentication scheme for the domain.

The LAC and LNS must have the same AAA authentication configurations.

For details about how to configure AAA authentication, see *AAA Configuration* in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide*.

 **NOTE**

If the LAC is trusted by the LNS, you can run the **authentication-mode none** command in the authentication scheme view of the LNS to set the authentication mode to non-authentication. If the command is configured, the LNS does not perform second authentication on remote users.

Procedure

- Configuring local authentication
 - a. Run **system-view**
Enter the system view.
 - b. Run **aaa**
Enter the AAA view.
 - c. Run **authentication-scheme** *authentication-scheme-name*
Create an authentication scheme, and enter the authentication scheme view.
By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
 - d. Run **authentication-mode local**
Set the authentication mode to **local**.
By default, local authentication is used.
 - e. Run **quit**
Return to the AAA view.
 - f. Run **domain** *domain-name*
Create a domain, and enter the domain view.
By default, the device has a domain named **default**, and its authentication mode is **local**.
 - g. Run **authentication-scheme** *authentication-scheme-name*
Specify an authentication scheme for the domain.
 - h. Run **quit**
Return to the AAA view.
 - i. Run **local-user** *user-name* **password cipher** *password*
Configure a user name and password for the local user, and store the user name and password on the device as the VPDN user information. The information is used to verify remote users.
The password is stored in cipher text mode.
 **NOTE**
To fully ensure the safety of the equipment, the users needs change the password on a regular basis.
 - j. Run **local-user** *user-name* **service-type ppp**
Configure a service type for the local user. The service type must be set to **ppp** because L2TP uses PPP negotiation.
 - k. Run **return**
Return to the user view.
- Configuring remote authentication and accounting

- a. Run **system-view**
Enter the system view.
- b. Run **radius-server template** *template-name*
Create a RADIUS server template, and enter the RADIUS server template view.
You can configure RADIUS server parameters in the RADIUS server template view.
- c. Run **radius-server authentication** *ip-address port*
Configure an IP address and a port number for the RADIUS server.
- d. Run **radius-server accounting** *ip-address port*
Configure a RADIUS accounting server.
By default, no RADIUS accounting server is configured.
- e. Run **radius-server shared-key cipher** *key-string*
Configure a shared key for connecting to the RADIUS server.
By default, the shared key is **huawei** in cipher text.
- f. Run **quit**
Return to the system view.
- g. Run **aaa**
Enter the AAA view.
- h. Run **authentication-scheme** *authentication-scheme-name*
Create an authentication scheme, and enter the authentication scheme view.
By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
- i. Run **authentication-mode radius**
Set the authentication mode to **radius**.
By default, local authentication is used.
- j. (Optional)Run **accounting-scheme** *accounting-scheme-name*
Create an accounting scheme and enter the accounting scheme view.
A default accounting scheme named **default** is available on the device. The **default** scheme can only be modified but cannot be deleted.
- k. (Optional)Run **accounting-mode radius**
Set the accounting mode to RADIUS.
By default, non-accounting is used.
- l. (Optional)Run **accounting start-fail** { **online** | **offline** }
Configure a policy for accounting-start failures.
By default, users cannot go online if accounting-start fails.
- m. (Optional)Run **accounting realtime** *interval*
Enable real-time accounting and set a real-time accounting interval.
- n. (Optional)Run **accounting interim-fail** [**max-times** *times*] { **online** | **offline** }
Specify the maximum number of real-time accounting requests and a policy for real-time accounting failures.
- o. Run **quit**
Return to the AAA view.

- p. Run **domain** *domain-name*
Create a domain, and enter the domain view.
By default, the device has a domain named **default**, and its authentication mode is **local**.
- q. Run **authentication-scheme** *authentication-scheme-name*
Specify an authentication scheme for the domain.
By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
- r. Run **radius-server** *template-name*
Specify RADIUS server template for users in the domain.
- s. (Optional)Run **accounting-scheme** *accounting-scheme-name*
Apply the accounting scheme to the domain.
By default, the accounting scheme **default** is applied to a domain. In this accounting scheme, non-accounting is used and the real-time accounting function is disabled.
- t. (Optional)Run **statistic enable**
If traffic-based accounting is used, enable traffic statistics collection in the domain.
By default, traffic statistics collection is disabled for a domain.
- u. Run **return**
Return to the user view.

----End

1.6.1.2 Configuring the LAC to Accept Dial-Up Calls and Initiate L2TP Connections

Context

Configure the LAC to accept dial-up calls for users and implement PPP negotiation with these users. Configure L2TP parameters to enable the LAC to initiate L2TP connections to the LNS based on the user name or domain name.

When configuring the LAC, note the following:

- When the user initiates a call connection request, the authentication mode must be the same as that configured for the virtual interface template on the LAC.
- Assign an IP address for the interface that connects the LAC to the user, to make the IP protocol on the interface take effect.
- Tunnel authentication is enabled by default, and no authentication password is configured.
 - If tunnel authentication is used, configure the same authentication password for the LAC and LNS.
 - If tunnel authentication is not used, disable tunnel authentication on the LAC and LNS.

Procedure

- Configure the LAC.

- a. Run **system-view**
The system view is displayed.
- b. Run **l2tp enable**
L2TP is enabled globally.
- c. Run **interface virtual-template vt-number**
A virtual interface template is created, and the virtual template view is displayed.
You can configure PPP negotiation parameters for the interface that functions as the PPPoE service interface.
 **NOTE**
PPPoE and L2TP services cannot be configured on the same VT interface simultaneously.
- d. Run **ppp authentication-mode { pap | chap }**
The PPP authentication mode is set to **pap** or **chap**.
The LAC and LNS must have the same authentication mode.
 **NOTE**
In PAP authentication, passwords are transmitted in plain text on the network, bringing potential security risks. CHAP authentication is recommended.
- e. Run **mtu size**
The MTU of the interface is set.
When the device interconnects with a non-Huawei device, set an MTU value on the virtual template interface to prevent an interconnection failure, for example, failure of the non-Huawei device to reassemble data packets after they are fragmented on a physical outbound interface of the Huawei device. The MTU value must be less than or equal to the encapsulation header length of L2TP packets (the encapsulation header length of an L2TP packet is 38 bytes but is 42 bytes when it carries sequence number information) subtracted from the MTU value on the physical outbound interface (1500 bytes by default). For example, when the MTU value on the physical outbound interface is 1500 bytes and the encapsulation header length of an L2TP packet is 42 bytes, the value of *size* in this step must be less than or equal to 1458.
If a physical interface performs packet fragmentation again after the packet is fragmented on the corresponding VT interface, device performance degrades. To prevent this case, you are advised to set the MTU value of the VT interface to the range of 1400 to 1450.
- f. Run **quit**
Return to the system view.
- g. Run **interface interface-type interface-number**
The view of the physical interface connected to remote users is displayed.
- h. Run **pppoe-server bind virtual-template vt-number**
The interface is configured to function as a PPPoE service interface, and bound to a virtual interface template.
- i. Run **quit**
Return to the system view.
- j. Run **l2tp-group group-number**
An L2TP group is created, and the L2TP group view is displayed.

You can configure L2TP connection parameters to enable the LAC to initiate L2TP connections to the LNS if the user information matches the configuration.

- k. Run **tunnel password** { **simple** | **cipher** } *password*

The password of the L2TP tunnel is configured. The password must be the same as that of the tunnel on the LNS.

Tunnel authentication is enabled by default, and no authentication password is configured.

It is recommended that you enable the tunnel authentication function. If the tunnel authentication function is not required, run the **undo tunnel authentication** command to disable the function.



If **simple** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

-
- l. Run **tunnel name** *tunnel-name*

A tunnel name is configured to enable the LNS to accept L2TP connections based on the LAC tunnel name.

By default, the device name is used as the tunnel name when no tunnel name is specified.

- m. Run either of the following commands to configure a public network address or domain name for the LNS, which specifies the destination address of control messages.

■ **start l2tp ip** *ip-address* &<1-4> { **domain** *domain-name* | **fullusername** *user-name* | **interface** *interface-type interface-number* | **vpn-instance** *vpn-instance-name* **fullusername** *user-name* }

■ **start l2tp host** *hostname* { **domain** *domain-name* | **fullusername** *user-name* }

The keywords define VPDN users.

■ **fullusername**: specifies a name for VPDN users. L2TP connections can be established for remote users with the same user name.

■ **domain**: specifies a domain name for VPDN users. L2TP connections can be established for remote users with the same domain name.

■ **vpn-instance**: specifies the VPN instance to which the IP address of the L2TP connections of a specific L2TP group belongs.

- n. Run **return**

Return to the user view.

----End

1.6.1.3 Configuring the LNS to Respond to the L2TP Connection Request

Context

Configure L2TP parameters to enable the LNS to respond to L2TP connection requests to the LAC based on the LAC tunnel name.

When configuring the LNS, note the following:

- When you configure PPP negotiation parameters on the virtual interface template, the authentication mode must be the same as that configured on the LAC.
- If the L2TP group number is not **1**, you must specify an LAC tunnel name.
- Tunnel authentication is enabled by default, and no authentication password is configured.
 - If tunnel authentication is used, configure the same authentication password for the LAC and LNS.
 - If tunnel authentication is not used, disable tunnel authentication on the LAC and LNS.
- If RADIUS authentication is used and Frame-IP and Frame-Route attributes are specified by the RADIUS server for users, the LNS delivers the Frame-IP and Frame-Route attributes to users and does not allocate IP addresses from the local address pool. The Frame-IP must be included in the local address pool.
- If the VPN instance attributes are configured for users on the RADIUS server when RADIUS authentication is used, you cannot bind the VPN instance to the VT interface of the LNS.

 **NOTE**

The LNS does not know users' real MAC addresses because user terminals use virtual MAC addresses allocated by the device. These virtual MAC addresses change randomly and cannot be bound with static IP addresses.

Procedure

- Configuring the LNS
 - a. Run **system-view**

The system view is displayed.
 - b. Run **l2tp enable**

L2TP is enabled globally.
 - c. Run **ip pool ip-pool-name**

A global IP address pool is created, and the global IP address pool view is displayed. The global IP address pool is used to allocate IP addresses to remote users.

This step is not required if you have manually configured a static IP address for the user.

 **NOTE**

L2TP can only allocate IP addresses of the address pool configured using the **ip pool** command but not attributes of other address pools to users.

If you want to allocate the DNS server address to users, add the **service-scheme** command to the AAA configuration.

- d. Run **network ip-address [mask { mask | mask-length }]**

A network segment is configured to allocate IP addresses dynamically from the largest to the smallest.
- e. Run **gateway-list ip-address <1-8>**

A gateway address is configured, and allocated to the remote user.

f. **Run quit**

Return to the system view.

g. **Run interface virtual-template vt-number**

A virtual interface template is created, and the virtual template view is displayed.

You can configure PPP negotiation parameters on the interface that functions as the private network gateway interface to accept L2TP connections of remote users.

 **NOTE**

PPPoE and L2TP services cannot be configured on the same VT interface simultaneously.

h. **Run ip address ip-address { mask | mask-length }**

An IP address is configured for the gateway in the headquarters.

i. **Run remote address { ip-address | pool pool-name }**

An IP address pool is configured to allocate IP addresses dynamically for remote users.

This step is not required if you have configured static IP addresses for remote users.

This step is not required if RADIUS authentication is used and an address pool name or Frame-IP attribute is specified by the RADIUS server. The LNS allocates IP addresses for remote users from the address pool specified by the RADIUS server.

When L2TP supports multiple address pools, omit this step if the **service-scheme** command has been run to specify an address pool. The LNS allocates IP addresses to remote users from the address pool specified by the **service-scheme** command.

 **NOTE**

If multiple users dial up using the same static IP address, users can go online but their service packets may fail to be forwarded if forcible address allocation is not configured on the LNS. Customers need to correctly plan static IP addresses. If the device must identify users and allow only one user terminal to connect to it, the planned address for the user terminal must be in the address pool and the **ppp ipcp remote-address forced** command must be configured.

j. **Run ppp authentication-mode { pap | chap }**

The PPP authentication mode is set to **pap** or **chap** to authenticate remote users.

The LAC and LNS must have the same authentication mode.

 **NOTE**

In PAP authentication, passwords are transmitted in plain text on the network, bringing potential security risks. CHAP authentication is recommended.

k. **Run mtu size**

The MTU of the interface is set.

When the device interconnects with a non-Huawei device, set an MTU value on the virtual template interface to prevent an interconnection failure, for example, failure of the non-Huawei device to reassemble data packets after they are fragmented on a physical outbound interface of the Huawei device. The MTU value must be less than or equal to the encapsulation header length of L2TP packets (the encapsulation header length of an L2TP packet is 38 bytes but is 42 bytes when it carries sequence number information) subtracted from the MTU value on the physical outbound interface (1500 bytes by default). For example, when the MTU value on the physical outbound interface is 1500 bytes and the encapsulation header length of an L2TP packet is 42 bytes, the value of *size* in this step must be less than or equal to 1458.

If a physical interface performs packet fragmentation again after the packet is fragmented on the corresponding VT interface, device performance degrades. To prevent this case, you are advised to set the MTU value of the VT interface to the range of 1400 to 1450.

l. Run **quit**

Return to the system view.

m. Run **l2tp-group** *group-number*

An L2TP group is created, and the L2TP group view is displayed.

You can configure L2TP connection parameters to accept connections initiated by the LAC.

When the L2TP group number is 1, the LNS accepts all the L2TP connections.

n. Run **tunnel password** { **simple** | **cipher** } *password*

The password of the L2TP tunnel is configured. The password must be the same as that of the tunnel on the LAC.

Tunnel authentication is enabled by default, and no authentication password is configured.

It is recommended that you enable the tunnel authentication function. If the tunnel authentication function is not required, run the **undo tunnel authentication** command to disable the function.

 **NOTICE**

If **simple** is selected, the password is saved in the configuration file in plain text. In this case, users at a lower level can easily obtain the password by viewing the configuration file. This brings security risks. Therefore, it is recommended that you select **cipher** to save the password in cipher text.

o. Run **tunnel name** *tunnel-name*

A tunnel name is configured. The tunnel name is used for PPP negotiation during tunnel establishment.

By default, the device name is used as the tunnel name when no tunnel name is specified.

p. Run **allow l2tp virtual-template** *virtual-template-number* [**remote** *remote-name* [**vpn-instance** *vpn-instance-name*]]

The L2TP group is configured as the LNS to respond to L2TP connection requests initiated by the LAC.

You must specify a virtual interface template and an LAC tunnel name.

When the L2TP group number is 1, the LNS accepts any L2TP connection requests from the LAC. You can choose not to specify the remote tunnel name.

q. Run **return**

Return to the user view.

----End

1.6.2 Configuring L2TP Client-Initiated L2TP Connections

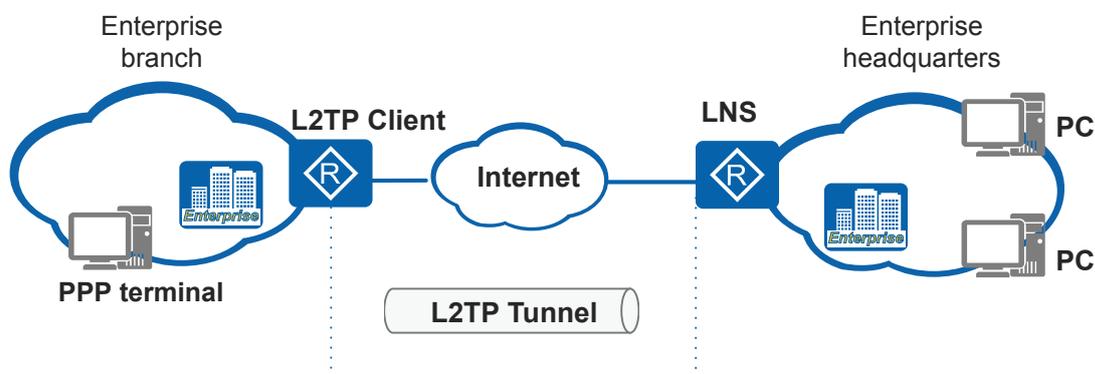
Access users do not need to dial up and they can access the L2TP Client in any means. The L2TP Client uses a virtual dial-up interface to initiate PPP sessions and L2TP connection requests to the LNS.

Context

An enterprise has some branches located in other cities, and its branches use the Ethernet and have gateways deployed, so that branch hosts can access the Internet.

A VPDN connection must be established between the headquarters gateway and branches to enable the headquarters to provide access services for the users. Any user from branches is allowed to access the headquarters gateway which only authenticates the gateways in branches. Configure the headquarters gateway as the LNS and the branch gateway as the L2TP Client. Configure virtual users on the branch gateway which dial up to initiate L2TP connections to the headquarters. In L2TP Client-Initiated mode, virtual point-to-point connections are established between the L2TP Client and LNS. After receiving IP packets from branch users, the L2TP Client forwards the packets through the virtual dial-up interface and the packets pass through L2TP tunnels to the LNS, which forwards the packets to the destination host.

Figure 1-17 Networking diagram for establishing L2TP Client-Initiated L2TP connections



Prerequisites

A reachable route has been configured between the LNS and the L2TP Client.

A LAN connection is established between branch users and the L2TP Client which functions as the gateway.

Configuration Process

Table 1-4 shows the configuration process on the L2TP Client. **Table 1-5** shows the configuration process on the LNS.

Table 1-4 Configuration process on the L2TP Client

Configuration	Procedure	Description
Configure the L2TP Client to initiate an L2TP connection.	Enable L2TP.	Enable L2TP globally.
	Configure PPP negotiation.	Configure parameters for a virtual interface template and use this template as a virtual dial-up interface. Assign an IP address for the VT interface to make the configuration take effect.
	Create an L2TP group.	Configure L2TP parameters, including the LAC tunnel name and password, LNS address, and VPDN user name. You can also configure the AVP data to be transmitted in cipher text, primary and secondary LNSs, and an interval for sending Hello packets.

Table 1-5 Configuration process on the LNS

Configuration	Procedure	Description
Configure local or remote AAA authentication.	Configure local authentication.	Store user information, including the user name, password, and service type, on the local device. You can also enable LCP renegotiation or mandatory CHAP authentication to implement second authentication on remote users.
	Configure remote authentication.	Configure RADIUS server parameters to enable the RADIUS server to store user information, including the user name, password, and service type, and authenticate access users. You can also enable LCP renegotiation or mandatory CHAP authentication to implement second authentication on remote users.
Configure the LNS to respond to the L2TP connection request.	Enable L2TP.	Enable L2TP globally.
	Configure an IP address pool.	Assign an IP address dynamically for the remote user after the user is authenticated. This step is not required when a static IP address is assigned to the remote user.
	Configure PPP negotiation.	Set PPP negotiation mode to PAP or CHAP on the VT interface. Configure an IP address and use this address as the private network gateway address of the L2TP tunnel. Import an IP address pool to dynamically allocate IP addresses for remote users. If mandatory CHAP authentication is configured, the PPP authentication mode must be CHAP.

Configuration	Procedure	Description
	Create an L2TP group.	Configure L2TP parameters, including the LNS tunnel name and password, number of the VT, and L2TP Client tunnel name. You can also configure the AVP data to be transmitted in cipher text, and an interval for sending Hello packets.

1.6.2.1 Configuring AAA Authentication and Accounting

Context

The AAA authentication provides authentication, authorization, and accounting security functions to manage remote access users and ensure secure connections. In L2TP Client initiated mode, you can configure local or remote authentication on the LNS to authenticate the L2TP Client.

After users are authenticated by the LNS, they can access the Internet. If you want to charge the users on their accessed network resources, you can configure the AAA accounting function on the LNS.

For details about how to configure AAA authentication, see *AAA Configuration* in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide*.

NOTE

If the L2TP Client is trusted by the LNS, you can run the **authentication-mode none** command in the authentication scheme view of the LNS to set the authentication mode to non-authentication. If the command is configured, the LNS does not perform second authentication on remote users.

Procedure

- Configuring local authentication
 - a. Run **system-view**
Enter the system view.
 - b. Run **aaa**
Enter the AAA view.
 - c. Run **authentication-scheme authentication-scheme-name**
Create an authentication scheme, and enter the authentication scheme view.
By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
 - d. Run **authentication-mode local**
Set the authentication mode to **local**.
By default, local authentication is used.
 - e. Run **quit**
Return to the AAA view.

- f. Run **domain** *domain-name*
Create a domain, and enter the domain view.
By default, the device has a domain named **default**, and its authentication mode is **local**.
- g. Run **authentication-scheme** *authentication-scheme-name*
Specify an authentication scheme for the domain.
- h. Run **quit**
Return to the AAA view.
- i. Run **local-user** *user-name* **password cipher** *password*
Configure a user name and password for the local user, and store the user name and password on the device as the VPDN user information. The information is used to verify remote users.
The password is stored in cipher text mode.
 **NOTE**
To fully ensure the safety of the equipment, the users needs change the password on a regular basis.
- j. Run **local-user** *user-name* **service-type ppp**
Configure a service type for the local user. The service type must be set to **ppp** because L2TP uses PPP negotiation.
- k. Run **return**
Return to the user view.
- Configuring remote authentication and accounting
 - a. Run **system-view**
Enter the system view.
 - b. Run **radius-server template** *template-name*
Create a RADIUS server template, and enter the RADIUS server template view. You can configure RADIUS server parameters in the RADIUS server template view.
 - c. Run **radius-server authentication** *ip-address port*
Configure an IP address and a port number for the RADIUS server.
 - d. Run **radius-server accounting** *ip-address port*
Configure a RADIUS accounting server.
By default, no RADIUS accounting server is configured.
 - e. Run **radius-server shared-key cipher** *key-string*
Configure a shared key for connecting to the RADIUS server.
By default, the shared key is **huawei** in cipher text.
 - f. Run **quit**
Return to the system view.
 - g. Run **aaa**
Enter the AAA view.
 - h. Run **authentication-scheme** *authentication-scheme-name*
Create an authentication scheme, and enter the authentication scheme view.

- By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
- i. Run **authentication-mode radius**
Set the authentication mode to **radius**.
By default, local authentication is used.
 - j. (Optional)Run **accounting-scheme** *accounting-scheme-name*
Create an accounting scheme and enter the accounting scheme view.
A default accounting scheme named **default** is available on the device. The **default** scheme can only be modified but cannot be deleted.
 - k. (Optional)Run **accounting-mode radius**
Set the accounting mode to RADIUS.
By default, non-accounting is used.
 - l. (Optional)Run **accounting start-fail** { **online** | **offline** }
Configure a policy for accounting-start failures.
By default, users cannot go online if accounting-start fails.
 - m. (Optional)Run **accounting realtime** *interval*
Enable real-time accounting and set a real-time accounting interval.
 - n. (Optional)Run **accounting interim-fail** [**max-times** *times*] { **online** | **offline** }
Specify the maximum number of real-time accounting requests and a policy for real-time accounting failures.
 - o. Run **quit**
Return to the AAA view.
 - p. Run **domain** *domain-name*
Create a domain, and enter the domain view.
By default, the device has a domain named **default**, and its authentication mode is **local**.
 - q. Run **authentication-scheme** *authentication-scheme-name*
Specify an authentication scheme for the domain.
By default, the device has an authentication scheme named **default**, and its authentication mode is **local**.
 - r. Run **radius-server** *template-name*
Specify RADIUS server template for users in the domain.
 - s. (Optional)Run **accounting-scheme** *accounting-scheme-name*
Apply the accounting scheme to the domain.
By default, the accounting scheme **default** is applied to a domain. In this accounting scheme, non-accounting is used and the real-time accounting function is disabled.
 - t. (Optional)Run **statistic enable**
If traffic-based accounting is used, enable traffic statistics collection in the domain.
By default, traffic statistics collection is disabled for a domain.
 - u. Run **return**

Return to the user view.

----End

1.6.2.2 Configuring the L2TP Client to Dial Up and Initiate L2TP Connections

Context

Create a virtual dial-up interface on the L2TP Client. The virtual users automatically dial up to initiate L2TP connections to the LNS.

When configuring the L2TP Client, note the following:

- As a PPP dial-up client, the L2TP Client can obtain an IP address for its virtual tunnel (VT) interface through PPP negotiation from the LNS. Or you can manually specify an IP address for the VT interface.
- Dial-up parameters, including the user name, password, and authentication mode, of the VT interface on the L2TP Client must be the same as those on the LNS.
- Tunnel authentication is enabled by default, and no authentication password is configured.
 - If tunnel authentication is used, configure the same authentication password for the L2TP Client and LNS.
 - If tunnel authentication is not used, disable tunnel authentication on the L2TP Client and LNS.

Procedure

- Configure the L2TP Client.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **l2tp enable**
L2TP is enabled globally.
 - c. Run **interface virtual-template vt-number**
A virtual interface template is created, and the virtual template view is displayed. You can configure dial-up parameters for the VT interface.
 **NOTE**
PPPoE and L2TP services cannot be configured on the same VT interface simultaneously.
 - d. Run **ip address ppp-negotiate**
The L2TP Client is configured to obtain an IP address from the LNS. You can run either of the following commands to make the IP protocol take effect.
 - Run the **ip address ip-address { mask | mask-length }** command to assign an IP address for the interface.
 - Run the **ip address unnumbered interface interface-type interface-number** command to use one IP address of the other interfaces.
 - e. Run **ppp pap local-user username password { cipher | simple } password**
 **NOTE**
When you specify **simple**, the password is saved in plain text in the configuration, which brings potential security risks. You are advised to specify **cipher** to save the password in the cipher text.

The PPP negotiation mode is set to **pap** and a user name and password are specified.

If the authentication mode is set to **chap**, run the following commands:

- **ppp chap user** *username*
- **ppp chap password** { **cipher** | **simple** } *password*

 **NOTE**

In PAP authentication, the password is transmitted in plain text on the network, which brings potential security risks.

- f. Run **l2tp-auto-client enable**

The device dial-up function is enabled.

- g. Run **mtu size**

The MTU of the interface is set.

When the device interconnects with a non-Huawei device, set an MTU value on the virtual template interface to prevent an interconnection failure, for example, failure of the non-Huawei device to reassemble data packets after they are fragmented on a physical outbound interface of the Huawei device. The MTU value must be less than or equal to the encapsulation header length of L2TP packets (the encapsulation header length of an L2TP packet is 38 bytes but is 42 bytes when it carries sequence number information) subtracted from the MTU value on the physical outbound interface (1500 bytes by default). For example, when the MTU value on the physical outbound interface is 1500 bytes and the encapsulation header length of an L2TP packet is 42 bytes, the value of *size* in this step must be less than or equal to 1458.

If a physical interface performs packet fragmentation again after the packet is fragmented on the corresponding VT interface, device performance degrades. To prevent this case, you are advised to set the MTU value of the VT interface to the range of 1400 to 1450.

- h. Run **quit**

Return to the system view.

- i. Run **interface** *interface-type interface-number*

The view of the physical interface connected to remote users is displayed.

- j. Run **ip address** *ip-address* { *mask* | *mask-length* }

An IP address is configured for the interface and used as the gateway address.

- k. Run **quit**

Return to the system view.

- l. Run **l2tp-group** *group-number*

An L2TP group is created, and the L2TP group view is displayed.

You can configure L2TP connection parameters to enable the L2TP Client to initiate L2TP connections to the LNS if the user information matches the configuration.

- m. Run **tunnel password** { **simple** | **cipher** } *password*

The password of the L2TP tunnel is configured. The password must be the same as that of the tunnel on the LNS.

Tunnel authentication is enabled by default, and no authentication password is configured.

It is recommended that you enable the tunnel authentication function. If the tunnel authentication function is not required, run the **undo tunnel authentication** command to disable the function.



NOTICE

If **simple** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

n. Run **tunnel name** *tunnel-name*

A tunnel name is configured to enable the LNS to accept L2TP connections based on the LAC tunnel name.

By default, the device name is used as the tunnel name when no tunnel name is specified.

o. Run either of the following commands to configure a public network address or domain name for the LNS, which specifies the destination address of control messages.

- **start l2tp ip** *ip-address* &<1-4> { **domain** *domain-name* | **fullusername** *user-name* | **interface** *interface-type interface-number* | **vpn-instance** *vpn-instance-name fullusername user-name* }
- **start l2tp host** *hostname* { **domain** *domain-name* | **fullusername** *user-name* }

The keywords define VPDN users.

- **fullusername**: specifies a name for VPDN users. L2TP connections can be established for remote users with the same user name.
- **domain**: specifies a domain name for VPDN users. L2TP connections can be established for remote users with the same domain name.
- **vpn-instance**: specifies the VPN instance to which the IP address of the L2TP connections of a specific L2TP group belongs.

p. Run **return**

Return to the user view.

----End

1.6.2.3 Configuring the LNS to Respond to the L2TP Connection Request

Context

Configure L2TP parameters to enable the LNS to respond to L2TP connection requests to the LAC based on the LAC tunnel name.

When configuring the LNS, note the following:

- When you configure PPP negotiation parameters on the virtual interface template, the authentication mode must be the same as that configured on the LAC.
- If the L2TP group number is not **1**, you must specify an LAC tunnel name.
- Tunnel authentication is enabled by default, and no authentication password is configured.

- If tunnel authentication is used, configure the same authentication password for the LAC and LNS.
- If tunnel authentication is not used, disable tunnel authentication on the LAC and LNS.
- If RADIUS authentication is used and Frame-IP and Frame-Route attributes are specified by the RADIUS server for users, the LNS delivers the Frame-IP and Frame-Route attributes to users and does not allocate IP addresses from the local address pool. The Frame-IP must be included in the local address pool.
- If the VPN instance attributes are configured for users on the RADIUS server when RADIUS authentication is used, you cannot bind the VPN instance to the VT interface of the LNS.

 **NOTE**

The LNS does not know users' real MAC addresses because user terminals use virtual MAC addresses allocated by the device. These virtual MAC addresses change randomly and cannot be bound with static IP addresses.

Procedure

- Configuring the LNS
 - a. Run **system-view**

The system view is displayed.
 - b. Run **l2tp enable**

L2TP is enabled globally.
 - c. Run **ip pool ip-pool-name**

A global IP address pool is created, and the global IP address pool view is displayed. The global IP address pool is used to allocate IP addresses to remote users.

This step is not required if you have manually configured a static IP address for the user.

 **NOTE**

L2TP can only allocate IP addresses of the address pool configured using the **ip pool** command but not attributes of other address pools to users.

If you want to allocate the DNS server address to users, add the **service-scheme** command to the AAA configuration.
 - d. Run **network ip-address [mask { mask | mask-length }]**

A network segment is configured to allocate IP addresses dynamically from the largest to the smallest.
 - e. Run **gateway-list ip-address <1-8>**

A gateway address is configured, and allocated to the remote user.
 - f. Run **quit**

Return to the system view.
 - g. Run **interface virtual-template vt-number**

A virtual interface template is created, and the virtual template view is displayed.

You can configure PPP negotiation parameters on the interface that functions as the private network gateway interface to accept L2TP connections of remote users.

 **NOTE**

PPPoE and L2TP services cannot be configured on the same VT interface simultaneously.

- h. Run **ip address** *ip-address* { *mask* | *mask-length* }

An IP address is configured for the gateway in the headquarters.

- i. Run **remote address** { *ip-address* | **pool** *pool-name* }

An IP address pool is configured to allocate IP addresses dynamically for remote users.

This step is not required if you have configured static IP addresses for remote users.

This step is not required if RADIUS authentication is used and an address pool name or Frame-IP attribute is specified by the RADIUS server. The LNS allocates IP addresses for remote users from the address pool specified by the RADIUS server.

When L2TP supports multiple address pools, omit this step if the **service-scheme** command has been run to specify an address pool. The LNS allocates IP addresses to remote users from the address pool specified by the **service-scheme** command.

 **NOTE**

If multiple users dial up using the same static IP address, users can go online but their service packets may fail to be forwarded if forcible address allocation is not configured on the LNS. Customers need to correctly plan static IP addresses. If the device must identify users and allow only one user terminal to connect to it, the planned address for the user terminal must be in the address pool and the **ppp ipcp remote-address forced** command must be configured.

- j. Run **ppp authentication-mode** { **pap** | **chap** }

The PPP authentication mode is set to **pap** or **chap** to authenticate remote users.

The LAC and LNS must have the same authentication mode.

 **NOTE**

In PAP authentication, passwords are transmitted in plain text on the network, bringing potential security risks. CHAP authentication is recommended.

- k. Run **mtu size**

The MTU of the interface is set.

When the device interconnects with a non-Huawei device, set an MTU value on the virtual template interface to prevent an interconnection failure, for example, failure of the non-Huawei device to reassemble data packets after they are fragmented on a physical outbound interface of the Huawei device. The MTU value must be less than or equal to the encapsulation header length of L2TP packets (the encapsulation header length of an L2TP packet is 38 bytes but is 42 bytes when it carries sequence number information) subtracted from the MTU value on the physical outbound interface (1500 bytes by default). For example, when the MTU value on the physical outbound interface is 1500 bytes and the encapsulation header length of an L2TP packet is 42 bytes, the value of *size* in this step must be less than or equal to 1458.

If a physical interface performs packet fragmentation again after the packet is fragmented on the corresponding VT interface, device performance degrades. To prevent this case, you are advised to set the MTU value of the VT interface to the range of 1400 to 1450.

- l. Run **quit**

Return to the system view.

- m. Run **l2tp-group** *group-number*
An L2TP group is created, and the L2TP group view is displayed.
You can configure L2TP connection parameters to accept connections initiated by the LAC.
When the L2TP group number is 1, the LNS accepts all the L2TP connections.
- n. Run **tunnel password** { **simple** | **cipher** } *password*
The password of the L2TP tunnel is configured. The password must be the same as that of the tunnel on the LAC.
Tunnel authentication is enabled by default, and no authentication password is configured.
It is recommended that you enable the tunnel authentication function. If the tunnel authentication function is not required, run the **undo tunnel authentication** command to disable the function.

 **NOTICE**

If **simple** is selected, the password is saved in the configuration file in plain text. In this case, users at a lower level can easily obtain the password by viewing the configuration file. This brings security risks. Therefore, it is recommended that you select **cipher** to save the password in cipher text.

-
- o. Run **tunnel name** *tunnel-name*
A tunnel name is configured. The tunnel name is used for PPP negotiation during tunnel establishment.
By default, the device name is used as the tunnel name when no tunnel name is specified.
 - p. Run **allow l2tp virtual-template** *virtual-template-number* [**remote** *remote-name* [**vpn-instance** *vpn-instance-name*]]
The L2TP group is configured as the LNS to respond to L2TP connection requests initiated by the LAC.
You must specify a virtual interface template and an LAC tunnel name.
When the L2TP group number is 1, the LNS accepts any L2TP connection requests from the LAC. You can choose not to specify the remote tunnel name.
 - q. Run **return**
Return to the user view.

---End

1.6.3 Configuring Other L2TP Functions

L2TP provides some other functions to ensure better L2TP services, such as LCP Renegotiation, AVP Parameter Encryption and Tunnel Authentication.

Prerequisites

Basic L2TP functions have been configured, and L2TP connections have been established between the LAC and the LNS.

Pre-Configuration Tasks

The L2TP functions supported by the device are as follows, and these configurations are optional.

1.6.3.1 Configuring LCP Renegotiation

Context

The LAC authenticates access users. After the users are authenticated, the LAC sends authentication information to the LNS that determines whether the users are authorized users.

If the LNS does not trust the LAC, you can configure LCP renegotiation to implement second authentication on the users. The users renegotiate with the LNS, and L2TP connections can be established only after renegotiation succeeds.

LCP renegotiation and mandatory CHAP authentication cannot be configured simultaneously. LCP renegotiation has a higher priority than mandatory CHAP authentication. Therefore, when both of them are configured, the device performs LCP renegotiation.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **l2tp-group group-number**

The L2TP group view is displayed.

Step 3 Run **mandatory-lcp**

LCP renegotiation is enabled.

----End

1.6.3.2 Configuring CHAP Mandatory Authentication

Context

You can configure mandatory CHAP authentication after the LNS receives authentication information transmitted from the LAC, when the LNS demands high security. After mandatory CHAP authentication is configured, the LNS performs only CHAP authentication for remote users. If the authentication mode is set to PAP authentication on the LAC, the LNS authentication fails and L2TP sessions cannot be established.

LCP renegotiation and mandatory CHAP authentication cannot be configured simultaneously. LCP renegotiation has a higher priority than mandatory CHAP authentication. Therefore, when both of them are configured, the device performs LCP renegotiation.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **l2tp-group** *group-number*

The L2TP group view is displayed.

Step 3 Run **mandatory-chap**

Mandatory CHAP authentication is enabled.

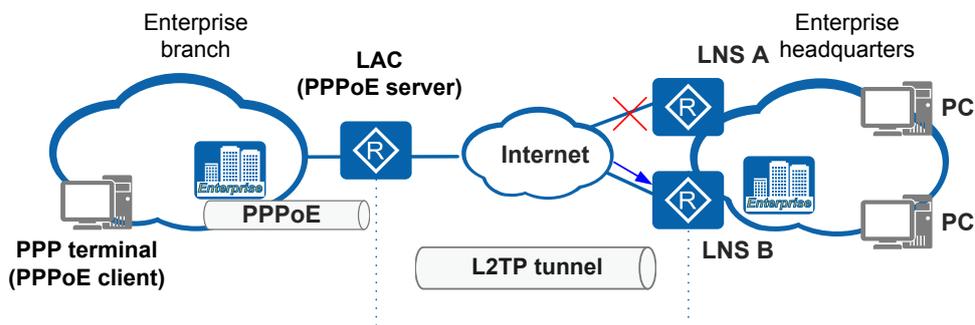
----End

1.6.3.3 Configuring Primary and Secondary LNSs

Context

You can configure double gateways (one primary and one secondary) in an enterprise headquarters if the enterprise demands high reliability. When the primary gateway fails, services are switched to the secondary gateway. However, L2TP connection requests initiated by the LAC cannot reach the primary LNS. To ensure that L2TP connection requests are sent to the secondary LNS, you must configure the IP address of the secondary gateway on the LAC, so that when the first IP address is unreachable, the LAC sends the request packets to the address of the secondary gateway.

Figure 1-18 Networking diagram of primary and secondary LNSs



Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **l2tp-group** *group-number*

The L2TP group view is displayed.

Step 3 Run **start l2tp ip** *ip-address* &<1-4> { **domain** *domain-name* | **fullusername** *user-name* }

An L2TP connection is initiated. A maximum of four LNS addresses can be configured.

----End

1.6.3.4 Configuring AVP Parameter Encryption

Context

An L2TP connection is established so that control messages can be exchanged between the LAC and LNS. Control messages carry various types of AVP parameters, including the user name and password. When the AVP parameter encryption is configured, AVP parameters are encrypted and the key information is hidden to improve communication security.

You must configure the L2TP tunnel authentication function before enabling the AVP parameter encryption.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **l2tp-group group-number**

The L2TP group view is displayed.

Step 3 Run **tunnel authentication**

The L2TP tunnel authentication is enabled.

Step 4 Run **tunnel password { simple | cipher } password**

An authentication password is configured. This password can also be used for encrypting AVP parameters.



NOTICE

If **simple** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

Step 5 Run **tunnel avp-hidden**

The AVP parameter encryption is enabled to hide key AVP parameters in L2TP packets.

----End

1.6.3.5 Configuring L2TP Tunnel Authentication

Context

You can configure the L2TP tunnel authentication when a network has a high security requirement. Configure the same authentication password on the LAC and LNS.

The LAC and LNS check the authentication password configured for each other. If the authentication password is the same, an L2TP tunnel can be established.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **l2tp-group** *group-number*

The L2TP group view is displayed.

Step 3 Run **tunnel authentication**

The L2TP tunnel authentication is enabled.

Step 4 Run **tunnel password** { **simple** | **cipher** } *password*

An authentication password is configured.



NOTICE

If **simple** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

----End

1.6.3.6 Configuring L2TP Tunnel Connectivity

Context

Hello packets are used to detect tunnel connectivity between the LAC and LNS.

When Hello packets time out, L2TP tunnels are automatically deleted to release network resources. The interval for sending Hello packets is set based on network requirements.

- If the network is stable, you can set a longer interval for sending Hello packets to reduce network burdens.
- If the network is unstable, you can set a shorter interval for sending Hello packets to detect tunnel status. When the primary and secondary LNSs are deployed, L2TP connection requests are sent to the IP address of the secondary LNS if the tunnel disconnection is detected.

When the device attempts to set up a tunnel to an LNS but the LNS runs abnormally, the device marks the LNS as unusable and does not set up a tunnel to the LNS in a period. This period is the LNS locking duration. After the locking duration, the device attempts to set up a tunnel to the LNS again.

Procedure

- Configuring interval for sending Hello packets
 - a. Run **system-view**

The system view is displayed.
 - b. Run **l2tp-group** *group-number*

The L2TP group view is displayed.

- c. Run **tunnel timer hello** *interval*
The interval for sending Hello packets is configured.
By default, Hello packets are sent at intervals of 60s.
- Configuring LNS locking duration
 - a. Run **system-view**
The system view is displayed.
 - b. Run **l2tp aging** *time*
The LNS locking duration is configured.
By default, the LNS locking duration is 30 seconds.
Perform this step on the LAC only.

----End

1.6.4 Verifying the L2TP Configuration

After configurations of other L2TP functions are complete, run the following commands on the LAC or the LNS to view the configurations, tunnel status, and session status.

Prerequisites

L2TP sessions have been established successfully between the LAC and LNS.

Procedure

- Run the **display l2tp tunnel** command to view information about L2TP tunnel IDs, session IDs, and public network addresses of the local end and the remote end.
- Run the **display l2tp session** command to view information about L2TP session IDs of the local end and the remote end and the local tunnel ID.
- Run the **display l2tp-group** [*group-number*] command to view configurations of the specified L2TP group.

----End

1.7 Maintaining L2TP

This section describes how to disconnect an L2TP tunnel forcibly and monitor the running status of L2TP.

1.7.1 Disconnecting an L2TP Tunnel Manually

Context

When there are no access users, a network fault occurs, or the administrator needs to disconnect an L2TP tunnel, you can manually disconnect the L2TP tunnel and the sessions in the tunnel. After the L2TP tunnel is disconnected manually, all the L2TP users who use the tunnel or sessions will go offline.

Before you disconnect an L2TP tunnel, run the **display l2tp tunnel** command to view the tunnel ID or the tunnel name of the remote end that you want to disconnect. Run the **display l2tp session** command to view its local session ID.

After you forcibly disconnect an L2TP tunnel, all control and session connections in the tunnel are removed. The tunnel can be re-established when a new user dials in.

Procedure

- Run the **reset l2tp tunnel** { **peer-name** *remote-name* | **local-id** *tunnel-id* } command in the user view to forcibly disconnect an L2TP tunnel based on the local tunnel ID or the remote tunnel name.
- Run the **reset l2tp session session-id** *session-id* command in the user view to forcibly disconnect a session connection based on the local session ID.

----End

1.7.2 Monitoring the Running Status of L2TP

Context

In routine maintenance, you can run the following commands in any view to view the running status of L2TP.

Procedure

- Run the **display l2tp-group** command to view the L2TP group number existing on the device.
- Run the **display l2tp tunnel** [**tunnel-item** *tunnel-item* | **tunnel-name** *tunnel-name*] command to view the detailed connection parameters of a specified tunnel based on the local tunnel ID or the remote tunnel name.
- Run the **display l2tp session** [**destination-ip** *d-ip-address* | **session-item** *session-item* | **source-ip** *s-ip-address*] command to view the detailed connection parameters of a specified session based on the local session ID and to view a session ID corresponding to a source or destination public network address of the L2TP tunnel.

----End

1.7.3 Collecting L2TP Packet Statistics

Context

When L2TP users go online, there are packet exchanges of multiple protocols. You can run the **display l2tp statistics tunnel** command to display L2TP packet statistics.

Procedure

- Run the **display l2tp statistics tunnel** [**local-id** *tunnel-id*] command in the user view to check L2TP packet statistics.
- Run the **reset l2tp statistics tunnel** [**local-id** *tunnel-id*] command in the user view to reset L2TP packet statistics.

----End

1.8 Configuration Examples for L2TP

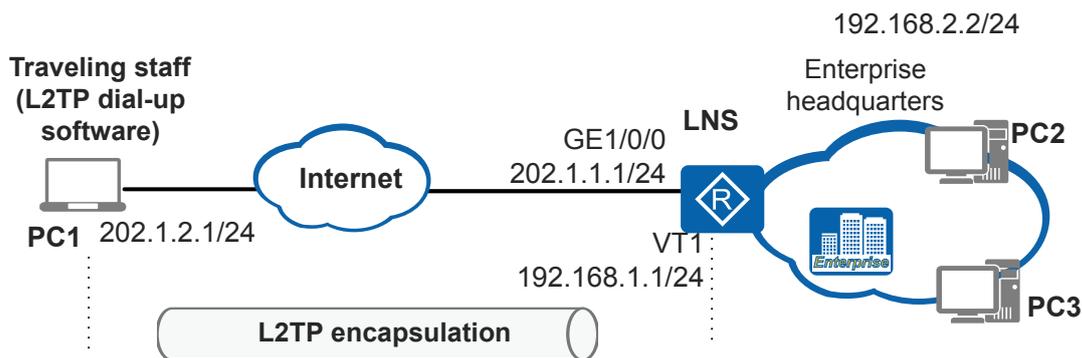
This section provides L2TP configuration examples.

1.8.1 Example for Configuring Client-Initiated L2TP Connections

Networking Requirements

As shown in [Figure 1-19](#), traveling employees need to communicate with the headquarters and access the headquarters gateway through the Internet to use internal resources. However, the headquarters gateway cannot identify and manage access users. To solve this problem, configure the headquarters gateway as the LNS to establish a virtual point-to-point connection between the traveling employees and the headquarters gateway when the employees use the L2TP dialup software on the PC to initiate L2TP connections. A PC running Windows 7 operating system is used in this example.

Figure 1-19 Networking diagram for establishing client-initiated L2TP connections



Configuration Roadmap

The configuration roadmap is as follows:

1. Connect the headquarters gateway to the Internet, and configure the gateway as the LNS to respond to L2TP connection requests sent by a traveling employee.
2. Connect the employee to the Internet, and enable the employee to initiate L2TP connections to the LNS using the L2TP dialup software.

Procedure

Step 1 Configure the LNS.

Configure an IP address and a route to the Internet. For example, set the next hop address to the Internet to 202.1.1.2.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] interface gigabitethernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 202.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
[LNS] ip route-static 0.0.0.0 0 202.1.1.2
```

Set the user name, password, and service type to **huawei**, **Huawei@1234**, and **ppp** respectively.

```
[LNS] aaa
[LNS-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

Configure an IP address pool used to assign addresses to dialup users.

```
[LNS] ip pool lns
[LNS-ip-pool-lns] network 192.168.1.0 mask 24
[LNS-ip-pool-lns] gateway-list 192.168.1.1
[LNS-ip-pool-lns] quit
```

Configure a virtual interface template.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ip address 192.168.1.1 255.255.255.0
[LNS-Virtual-Template1] ppp authentication-mode chap
[LNS-Virtual-Template1] remote address pool lns
[LNS-Virtual-Template1] quit
```

Enable L2TP and create an L2TP group numbered 1.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Disable the tunnel authentication function. The PC running Windows 7 operating system does not support tunnel authentication.

```
[LNS-l2tp1] undo tunnel authentication
```

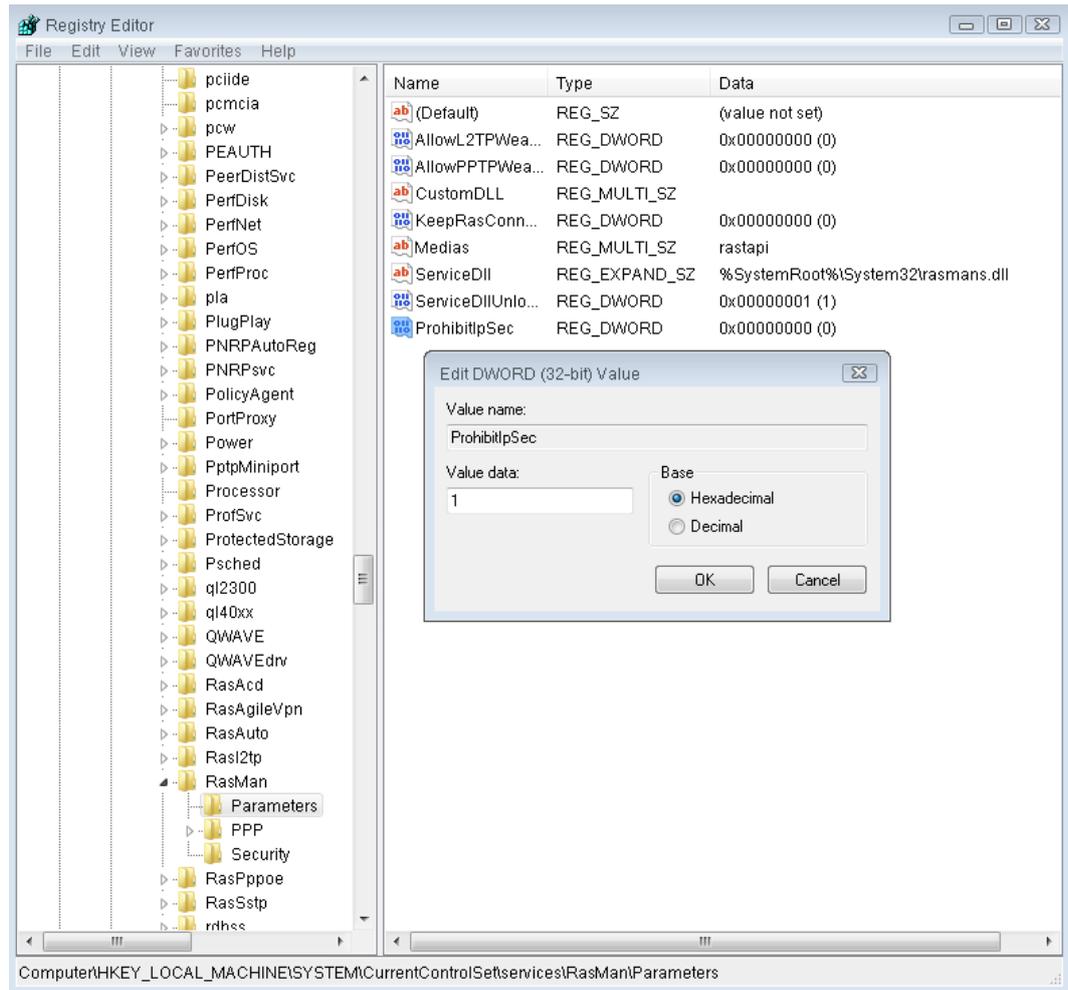
Bind the LNS to the virtual interface template.

```
[LNS-l2tp1] allow l2tp virtual-template 1
```

Step 2 Configure the Windows 7 operating system.

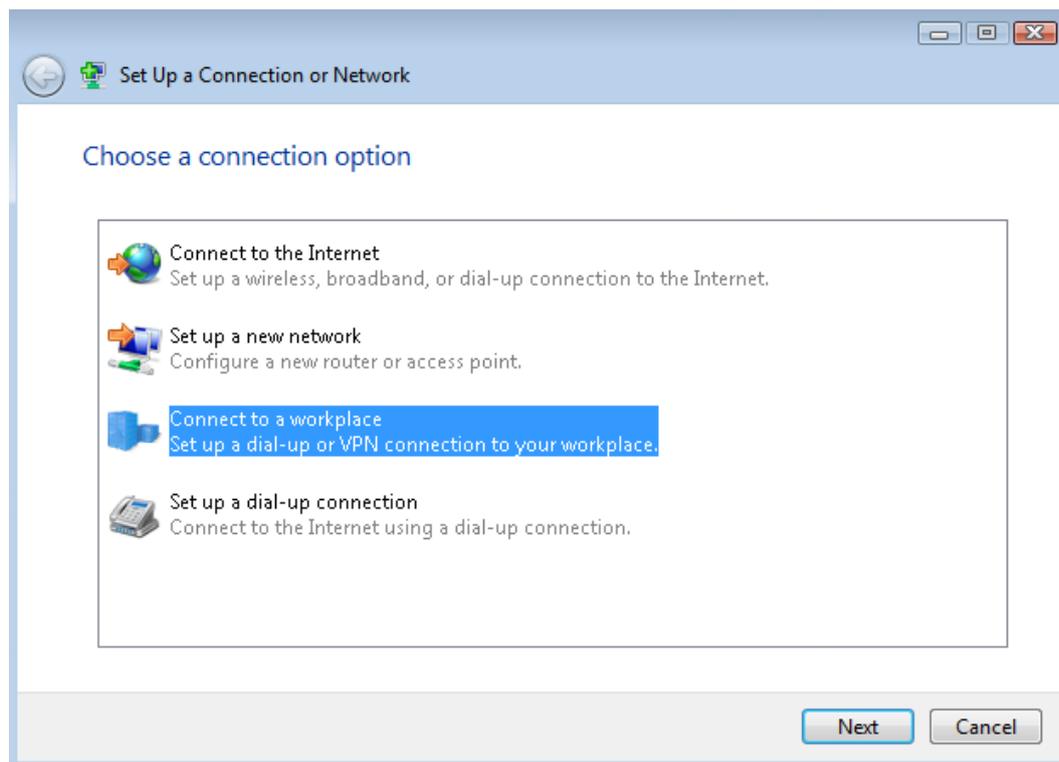
Modify the Windows registry and disable the digital certificate authentication function.

Choose **Start > Run** and enter **regedit** to open the Registry Editor. Open **Parameters** in **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rasman**, create **DWORD** and set the name and value to **ProhibitIpSec** and **1** respectively. After modifying the parameters, restart the PC.

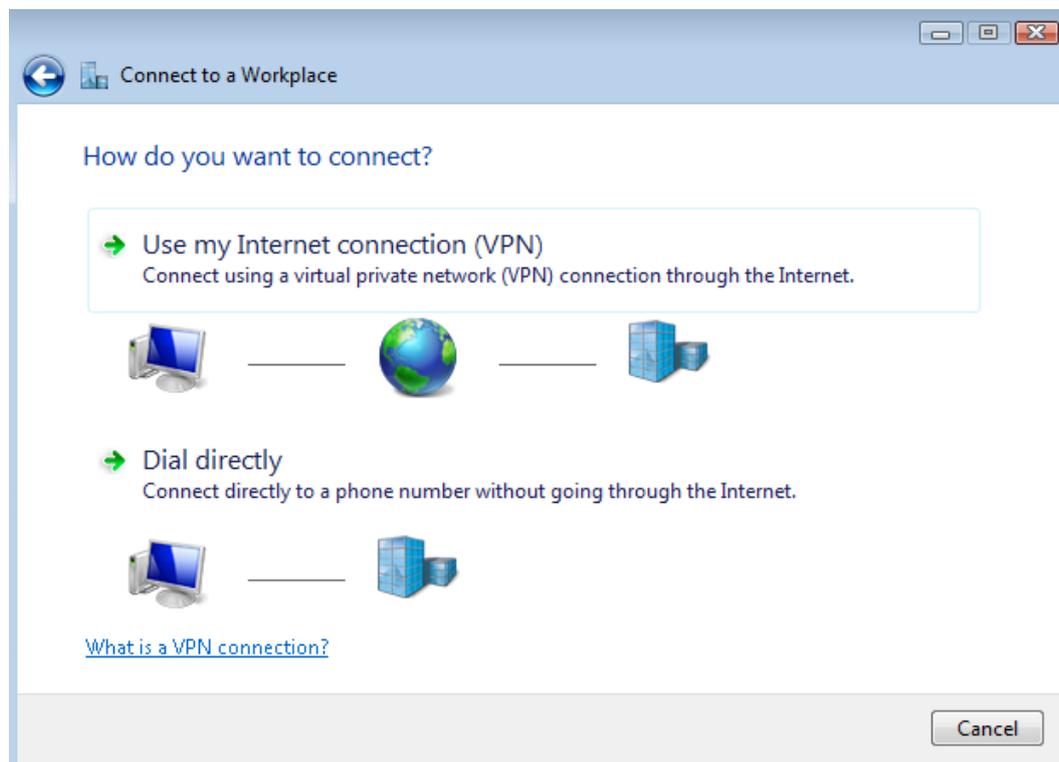


Create an L2TP network connection.

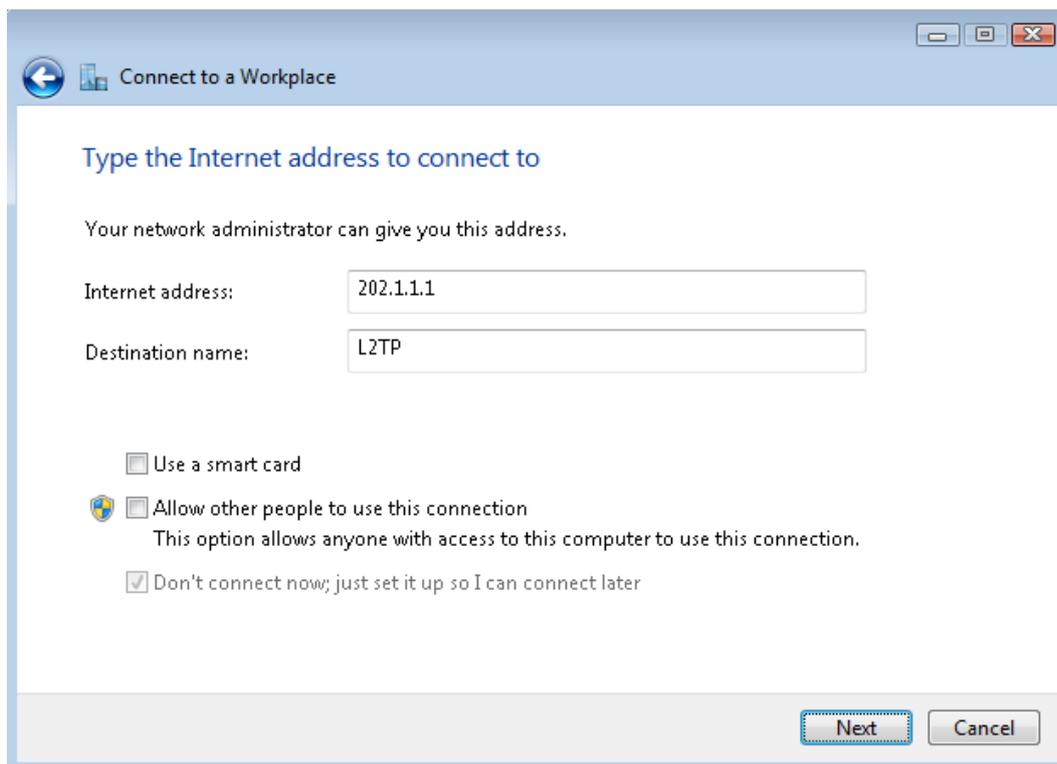
Choose **Start > Run > Network and Sharing Center**, click **Set Up a Connection or Network**, choose **Connect to a workplace**, and click **Next**.



Click **Use my Internet connection (VPN)**.



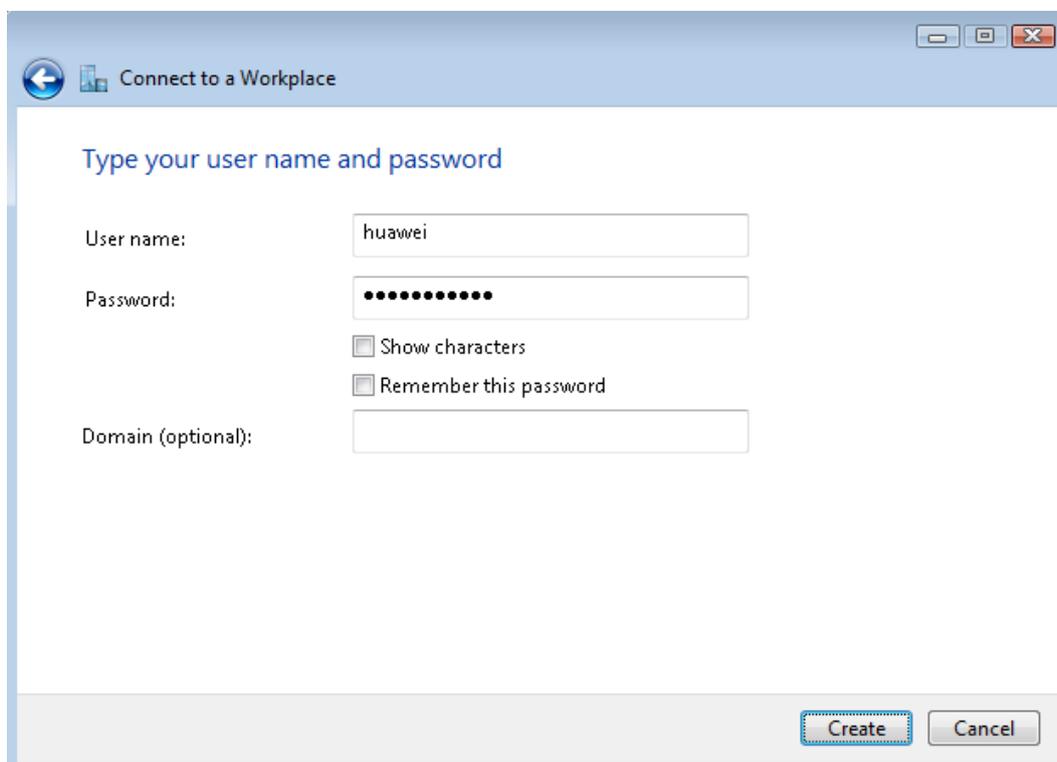
Enter an Internet address which is the IP address of the LNS (**202.1.1.1**), enter a destination name (for example, **L2TP**) as the network connection name, and click **Next**. You can customize a destination name.



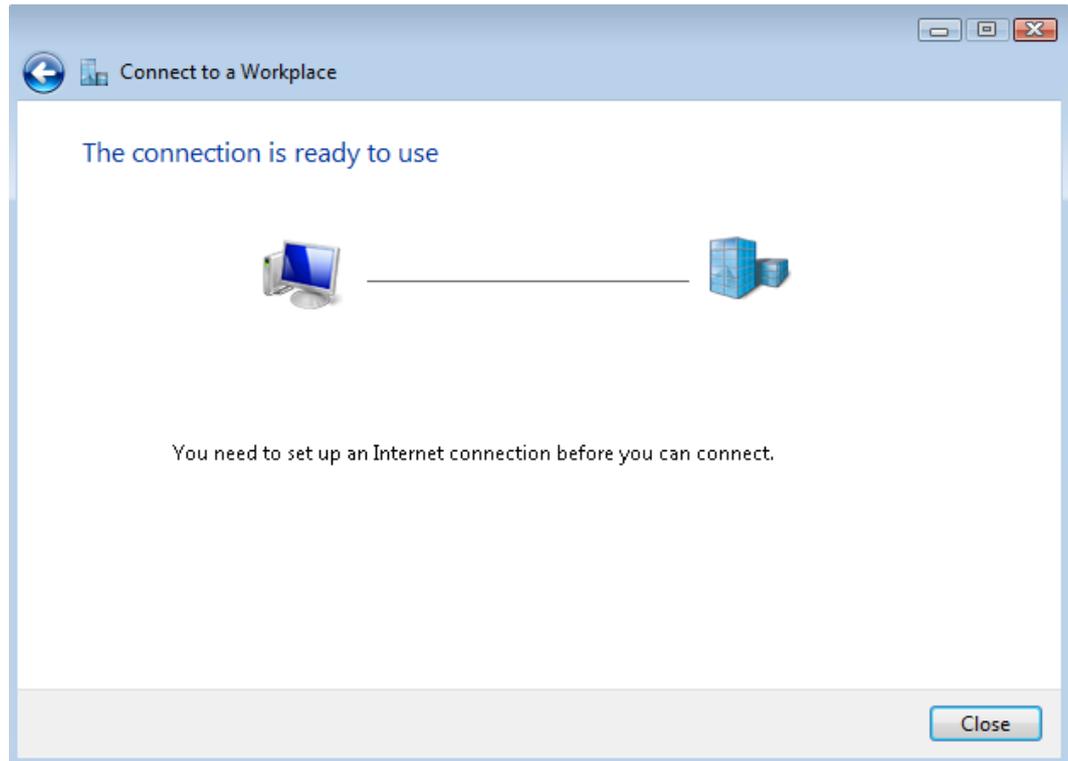
Enter the user name **huawei** and password **Huawei@1234** and click **Create**.

 **NOTE**

You do not need to set the domain.



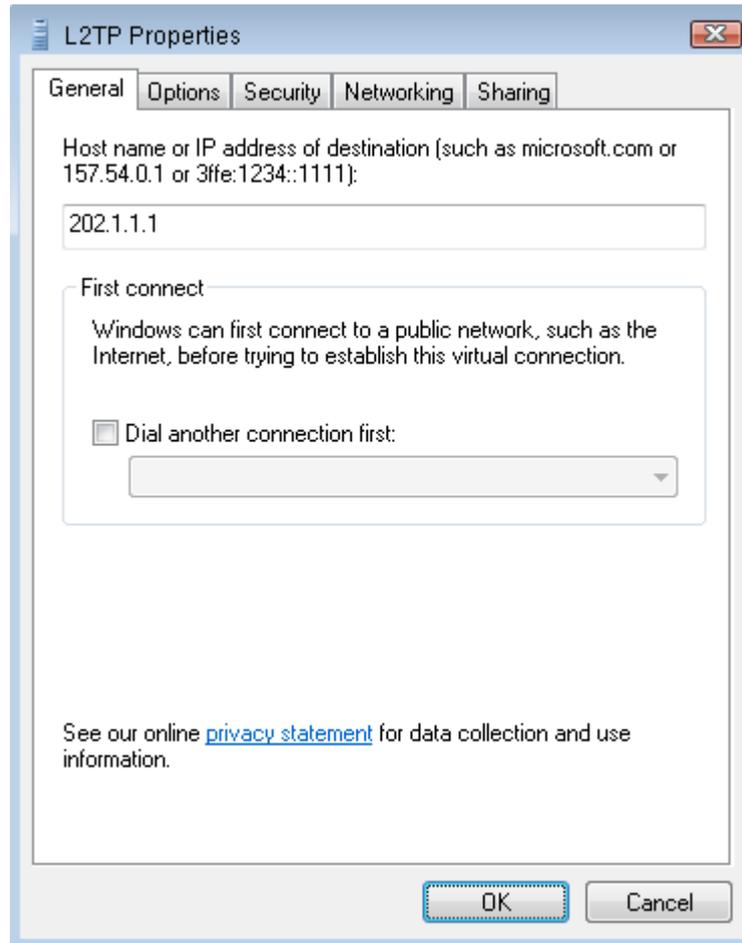
Click **Close**.



Set authentication parameters for the L2TP connection.

Choose **Start > Run > Network and Sharing Center** and click **Connect to a network**. The created **L2TP** connection is displayed. Right-click **L2TP** and choose **Properties** to set connection parameters.

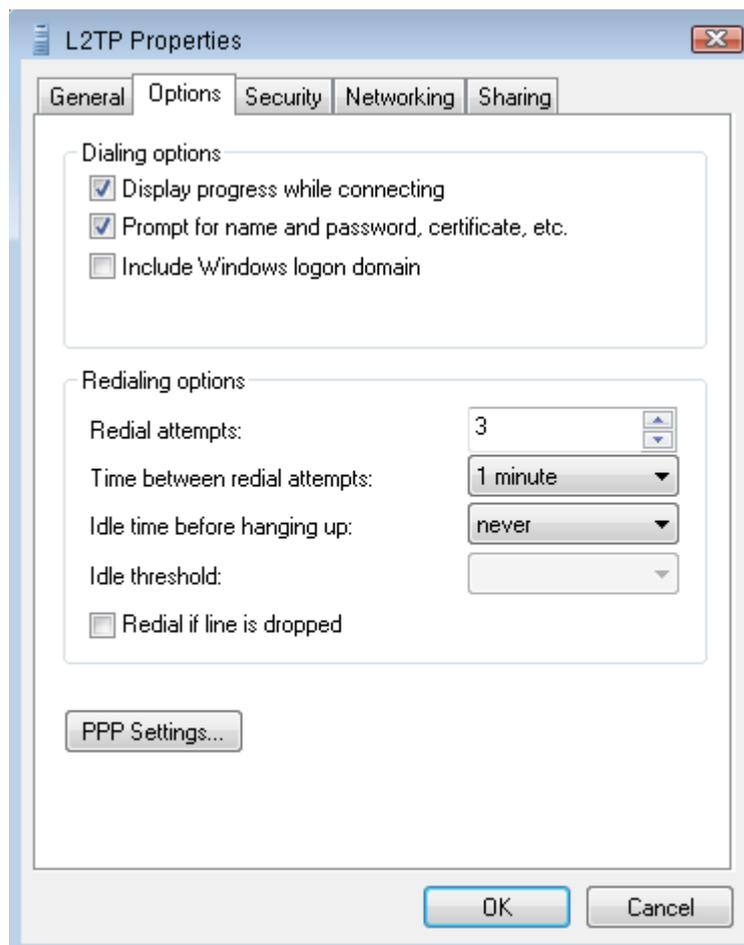
You do not need to modify parameters on the **General** tab.



Select **Display progress while connecting** and **Prompt for name and password certificate, etc** on the **Options** tab.

 **NOTE**

Do not change the parameters that are displayed after you click **PPP Settings**.

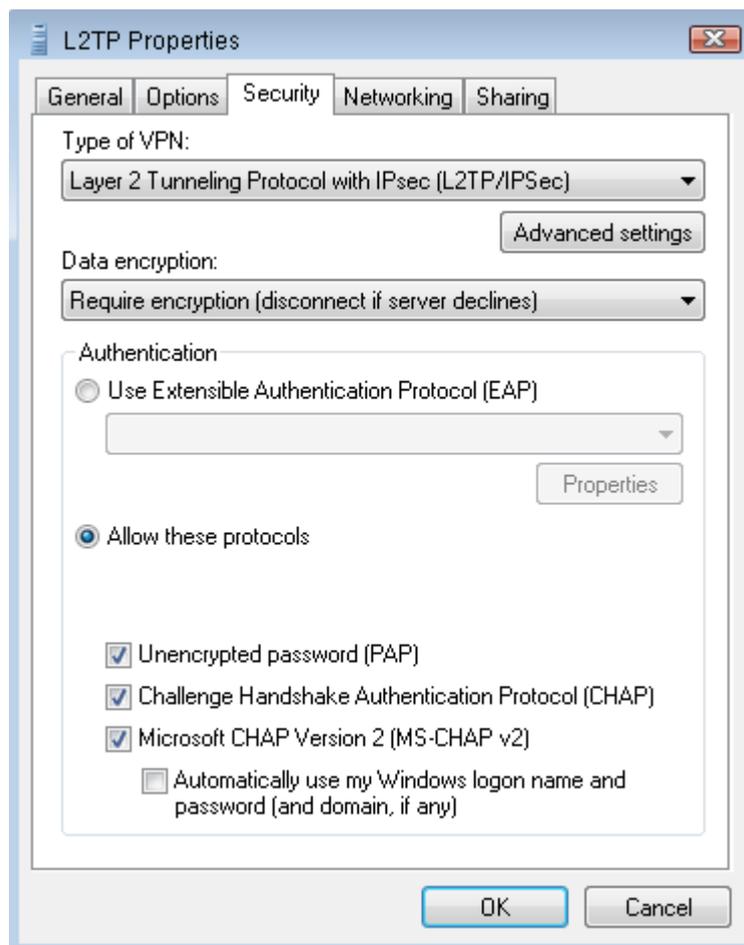


On the **Security** tab, select **Automatic** or **Layer 2 Tunneling Protocol with IPsec** for **Type of VPN**.

Select **Unencrypted password [PAP]**, **Challenge Handshake Authentication Protocol [CHAP]**, and **Microsoft CHAP Version 2 [MS-CHAP v2]** in **Allow these protocols**.

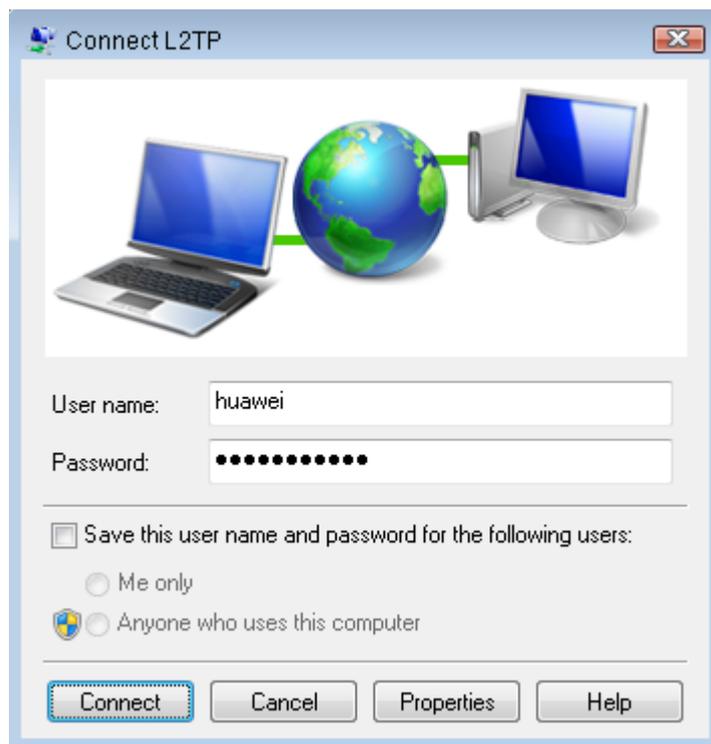
NOTE

If you click **Advanced settings**, a dialog box is displayed on which you can set the IPsec pre-shared key. Do not set the IPsec pre-shared key here.



You do not need to modify settings on the **Networking** and **Sharing** tabs.

Choose **Start > Run > Network and Sharing Center** and click **Connect to a network**. The created **L2TP** connection is displayed. Right-click **L2TP**, enter the user name and password, and click **Connect**.



Step 3 Verify the configuration.

After the configurations are complete, PC 1 obtains a private network address **192.168.1.254** for the **L2TP** connection, and PC 1 can communicate with the PC in the headquarters.

---End

Configuration File

Configuration file of the LNS

```
#
 sysname LNS
#
 l2tp enable
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.1 255.255.255.0
#
aaa
 local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F\H0Qv8B]KAZja3}3q'RNx;VI%^
 %#
 local-user huawei privilege level 0
 local-user huawei server-type ppp
#
 l2tp-group 1
 undo tunnel authentication
 allow l2tp virtual-template 1
#
interface Virtual-Template1
 ppp authentication-mode chap
 remote address pool lns
 ip address 192.168.1.1 255.255.255.0
#
 ip pool lns
 network 192.168.1.0 mask 255.255.255.0
```

```
gateway-list 192.168.1.1
#
ip route-static 0.0.0.0 0.0.0.0 202.1.1.2
#
return
```

1.8.2 Example for Configuring the LAC to Initiate Call-Triggered L2TP Connections (Dial-Up Users)

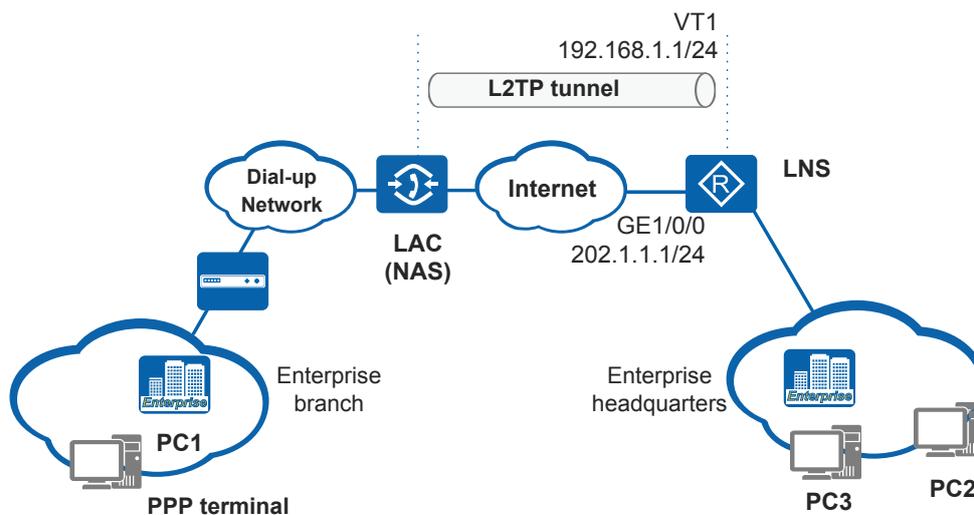
Networking Requirements

As shown in **Figure 1-20**, an enterprise has a branch located in another city, and the branch is located in a traditional dial-up network.

Branch users need to establish VPDN connections with users at the headquarters. Therefore, the branch users apply for the L2TP service from the ISP. The ISP configures the NAS as the LAC to send call connecting requests to the LNS through the Internet.

The gateway in the headquarters is configured as the LNS to establish L2TP connections between the branch and the headquarters.

Figure 1-20 Networking diagram for configuring the LAC to initiate call-triggered L2TP connections (dial-up users)



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure local AAA authentication for the LNS to authenticate dial-up users.
2. Create an IP address pool and allocate IP addresses to users, so that the LNS can manage the users.
3. Configure negotiation parameters using the virtual interface template, so that the LNS can implement PPP negotiation with the users.

4. Configure an L2TP group and create a tunnel between the LAC and LNS, so that the LNS can accept L2TP connection requests.

Procedure

- Step 1** Configure AAA authentication, and set the user name and password to **huawei** and **Huawei@1234**.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] aaa
[LNS-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

- Step 2** Configure a private IP address pool.

```
[LNS] ip pool 1
[LNS-ip-pool-1] network 192.168.1.0 mask 24
[LNS-ip-pool-1] gateway-list 192.168.1.1
[LNS-ip-pool-1] quit
```

- Step 3** Set PPP negotiation parameters.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ip address 192.168.1.1 255.255.255.0
[LNS-Virtual-Template1] ppp authentication-mode chap
[LNS-Virtual-Template1] remote address pool 1
[LNS-Virtual-Template1] quit
```

- Step 4** Configure the LNS to accept L2TP connection requests.

Enable L2TP and configure an L2TP group.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Configure an LNS tunnel name and LAC tunnel name.

```
[LNS-l2tp1] tunnel name LNS
[LNS-l2tp1] allow l2tp virtual-template 1 remote LAC
```

Enable the tunnel authentication function, and configure an authentication password.

```
[LNS-l2tp1] tunnel authentication
[LNS-l2tp1] tunnel password cipher huawei
[LNS-l2tp1] quit
```

Configure an IP address and a route to the Internet. For example, set the next hop address to the Internet to 202.1.1.2.

```
[LNS] interface gigabitethernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 202.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
[LNS] ip route-static 0.0.0.0 0 202.1.1.2
```

- Step 5** Verify the configuration.

After PC 1 goes online, run the **display l2tp tunnel** command on the LNS. The tunnel and session are established.

```
[LNS] display l2tp tunnel
```

```
Total tunnel : 1
LocalTID RemoteTID RemoteAddress Port Sessions RemoteName
1 1 202.1.2.1 1701 1 LAC
```

Check that PC 1 can communicate with hosts in the enterprise headquarters.

----End

Configuration File

Configuration file of the LNS

```
#
 sysname LNS
#
 l2tp enable
#
 ip pool 1
 network 192.168.1.0 mask 255.255.255.0
 gateway-list 192.168.1.1
#
 aaa
 local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F\H0Qv8B]KAZja3}3q'RNx;VI%^
 %#
 local-user huawei privilege level 0
 local-user huawei service-type ppp
#
 interface Virtual-Template1
 ppp authentication-mode chap
 remote address pool 1
 ip address 192.168.1.1 255.255.255.0
#
 interface
 GigabitEthernet1/0/0
 ip address 202.1.1.1 255.255.255.0
#
 l2tp-group 1
 allow l2tp virtual-template 1 remote LAC
 tunnel password cipher %@@@/-#)Iq[S4F:#2~ZNvqa$]\DL%@@@
 tunnel name LNS
#
 ip route-static 0.0.0.0 0.0.0.0 202.1.1.2
#
return
```

1.8.3 Example for Configuring the LAC to Initiate Call-Triggered L2TP Connections (PPPoE Users)

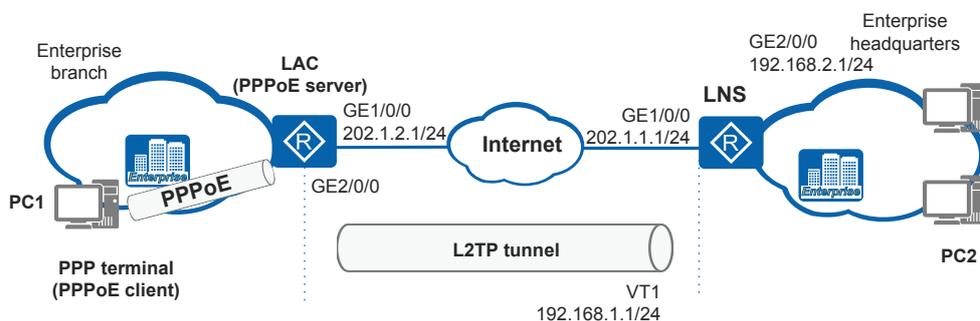
Networking Requirements

As shown in [Figure 1-21](#), an enterprise has some branches located in other cities, and branches use the Ethernet network.

The branch staff need to establish VPDN connections with the headquarters. L2TP is deployed between the branch and the headquarters. The branch has no dial-up network, and its gateway functions as a PPPoE server to allow dial-up data to be transmitted over the Ethernet. The branch gateway also functions as the LAC to establish L2TP tunnels with the headquarters.

The gateway at the enterprise headquarters is configured as the LNS to establish L2TP connections between the branch and headquarters.

Figure 1-21 Networking diagram for the LAC to initiate call-triggered L2TP connections (PPPoE users)



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the LAC as a PPPoE server and enable CHAP authentication so that the LAC can accept dial-up data from branch users over the Ethernet.
2. Configure local AAA authentication for the LNS to authenticate dial-up users.
3. Configure the LAC to establish L2TP connections to the headquarters for dial-up users that are authenticated.
4. Configure local AAA authentication for the LNS to authenticate dial-up users.
5. Create an IP address pool and allocate IP addresses to users, so that the LNS can manage the users.
6. Configure negotiation parameters using the virtual interface template, so that the LNS can implement PPP negotiation with the users.
7. Configure an L2TP group and create a tunnel between the LAC and LNS, so that the LNS can accept L2TP connection requests.

Procedure

Step 1 Configure the LAC as a PPPoE server.

Create a virtual interface template and configure PPP negotiation mode.

```
<Huawei> system-view
[Huawei] sysname LAC
[LAC] interface virtual-template 1
[LAC-Virtual-Template1] ppp authentication-mode chap
[LAC-Virtual-Template1] quit
```

Configure the PPPoE service on the physical interface at the user side and bind the interface to a virtual interface template.

```
[LAC] interface gigabitethernet 2/0/0
[LAC-GigabitEthernet2/0/0] pppoe-server bind virtual-template 1
[LAC-GigabitEthernet2/0/0] quit
```

Step 2 Configure the AAA authentication, and set the user name and password to **huawei** and **Huawei@1234** on the LAC.

```
[LAC] aaa
[LAC-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LAC-aaa] local-user huawei service-type ppp
[LAC-aaa] quit
```

Step 3 Configure the LAC to initiate an L2TP connection.

Enable L2TP and configure an L2TP group.

```
[LAC] l2tp enable
[LAC] l2tp-group 1
```

Configure a tunnel name for the LAC local end and specify a public IP address for the LNS.

```
[LAC-l2tp1] tunnel name lac
[LAC-l2tp1] start l2tp ip 202.1.1.1 fullusername huawei
```

Enable the tunnel authentication function, and configure an authentication password. The password must be the same as that on the LNS.

```
[LAC-l2tp1] tunnel authentication
[LAC-l2tp1] tunnel password cipher huawei
[LAC-l2tp1] quit
```

Configure an IP address for the public-network-side interface.

```
[LAC] interface gigabitethernet 1/0/0
[LAC-GigabitEthernet1/0/0] ip address 202.1.2.1 255.255.255.0
[LAC-GigabitEthernet1/0/0] quit
```

Configure a static route to the LNS. For example, set the next hop IP address to 202.1.2.2.

```
[LAC] ip route-static 202.1.1.1 32 202.1.2.2
```

Step 4 Configure the AAA authentication on the LNS.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] aaa
[LNS-aaa] local-user huawei password cipher Huawei@1234
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

Step 5 Configure a private IP address pool for the LNS.

```
[LNS] ip pool 1
[LNS-ip-pool-1] network 192.168.1.0 mask 24
[LNS-ip-pool-1] gateway-list 192.168.1.1
[LNS-ip-pool-1] quit
```

Step 6 Set PPP negotiation parameters for the LNS.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ip address 192.168.1.1 255.255.255.0
[LNS-Virtual-Template1] ppp authentication-mode chap
[LNS-Virtual-Template1] remote address pool 1
[LNS-Virtual-Template1] quit
```

Step 7 Configure the LNS to respond to the L2TP connection request.

Enable L2TP and configure an L2TP group.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Configure an LNS tunnel name and LAC tunnel name.

```
[LNS-12tp1] tunnel name lns
[LNS-12tp1] allow l2tp virtual-template 1 remote lac
```

Enable the tunnel authentication function, and configure an authentication password.

```
[LNS-12tp1] tunnel authentication
[LNS-12tp1] tunnel password cipher huawei
[LNS-12tp1] quit
```

Configure an IP address for the public-network-side interface.

```
[LNS] interface gigabitEthernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 202.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
```

Configure a static route to the LAC. For example, set the next hop IP address to 202.1.1.2.

```
[LNS] ip route-static 202.1.2.1 32 202.1.1.2
```

Configure a private IP address.

```
[LNS] interface gigabitEthernet 2/0/0
[LNS-GigabitEthernet2/0/0] ip address 192.168.2.1 255.255.255.0
[LNS-GigabitEthernet2/0/0] quit
```

Step 8 Verify the configuration.

After PC 1 goes on line, run the **display pppoe-server session all** command on the LAC to view the PPPoE sessions.

```
[LAC] display pppoe-server session all
SID Intf          State OIntf          RemMAC
LocMAC
1 Virtual-Templat1:0 UP GE2/0/0 5489.98f7.2fcb 5489.9872.366f
```

Run the **display l2tp tunnel** command on the LAC or LNS to view L2TP tunnel and session information. The command output for the LNS is shown as an example.

```
[LNS] display l2tp tunnel

Total tunnel : 1
LocalTID RemoteTID RemoteAddress Port Sessions RemoteName
1 1 202.1.2.1 1701 1 lac
```

Check that PC 1 can communicate with PC 2 in the enterprise headquarters.

----End

Configuration Files

- Configuration file of the LAC

```
#
sysname LAC
#
l2tp enable
#
aaa
local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F
\H0Qv8B]KAZja3}3q'RNx;VI%^%#
local-user huawei privilege level 0
local-user huawei service-type ppp
#
interface Virtual-Templat1
```

```
ppp authentication-mode chap
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 pppoe-server bind Virtual-Template 1
#
l2tp-group 1
 tunnel password cipher %@%@/-#)Lg[S4F:#2~ZNvqa$]\DL%@@@
 tunnel name lac
 start l2tp ip 202.1.1.1 fullusername huawei
#
ip route-static 202.1.1.1 255.255.255.255 202.1.2.2
#
return
```

- Configuration file of the LNS

```
#
sysname LNS
#
l2tp enable
#
ip pool 1
 network 192.168.1.0 mask 255.255.255.0
 gateway-list 192.168.1.1
#
aaa
 local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F
 \H0Qv8B]KAZja3}3q'RNx;VI%^%#
 local-user huawei privilege level 0
 local-user huawei service-type ppp
#
interface Virtual-Template1
 ppp authentication-mode chap
 remote address pool 1
 ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 192.168.2.1 255.255.255.0
#
l2tp-group 1
 allow l2tp virtual-template 1 remote lac
 tunnel password cipher %@%@EB~j7Je>;@>uNr''D=J<]\WL%@@@
 tunnel name lns
#
ip route-static 202.1.2.1 255.255.255.255 202.1.1.2
#
return
```

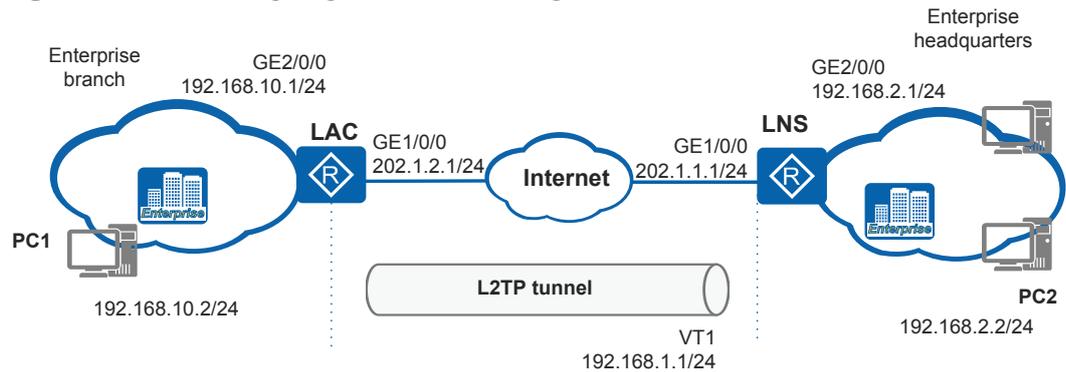
1.8.4 Example for Configuring an L2TP Client-Initiated L2TP Connection

Networking Requirements

As shown in [Figure 1-22](#), an enterprise has some branches located in other cities, and branches use the Ethernet network.

The headquarters network provides VPDN services for the branch staff to allow them to access the network of the headquarters. The LNS only authenticates the L2TP Client. The L2TP Client dials up to establish an L2TP connection to the LNS.

Figure 1-22 Networking diagram for establishing an L2TP Client-Initiated L2TP connection



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable L2TP on the L2TP Client. The virtual PPP user sends a connection request to the server in the headquarters over an L2TP tunnel. After the PPP user is authenticated, a tunnel is set up.
2. On the L2TP Client, configure a reachable route to the LNS and enable the dial-up function.
3. On the LNS, configure L2TP, a virtual PPP user, and a route to the public network segment.

Procedure

Step 1 Configure the L2TP Client.

Configure an IP address for the public-network-side interface.

```
<Huawei> system-view
[Huawei] sysname L2TP Client
[L2TP Client] interface gigabitethernet 1/0/0
[L2TP Client-GigabitEthernet1/0/0] ip address 202.1.2.1 255.255.255.0
[L2TP Client-GigabitEthernet1/0/0] quit
```

Configure an IP address for the user-side interface.

```
[L2TP Client] interface gigabitethernet 2/0/0
[L2TP Client-GigabitEthernet2/0/0] ip address 192.168.10.1 255.255.255.0
[L2TP Client-GigabitEthernet2/0/0] quit
```

Enable L2TP globally, create an L2TP group, and configure the user **huawei** to establish an L2TP connection to the LNS.

```
[L2TP Client] l2tp enable
[L2TP Client] l2tp-group 1
[L2TP Client-l2tp1] tunnel name L2TP Client
[L2TP Client-l2tp1] start l2tp ip 202.1.1.1 fullusername huawei
```

Enable tunnel authentication and set the tunnel password.

```
[L2TP Client-l2tp1] tunnel authentication
[L2TP Client-l2tp1] tunnel password cipher huawei
[L2TP Client-l2tp1] quit
```

Configure the user name and password, authentication mode, and IP address for the virtual PPP user.

```
[L2TP Client] interface virtual-template 1
[L2TP Client-Virtual-Template1] ppp chap user huawei
[L2TP Client-Virtual-Template1] ppp chap password cipher Huawei@1234
[L2TP Client-Virtual-Template1] ip address ppp-negotiate
[L2TP Client-Virtual-Template1] quit
```

On the LNS, configure a static route to the public network. For example, set the next hop address to 202.1.2.2.

```
[L2TP Client] ip route-static 202.1.1.1 255.255.255.255 202.1.2.2
```

Enable the L2TP Client to dial up and establish an L2TP tunnel.

```
[L2TP Client] interface virtual-template 1
[L2TP Client-Virtual-Template1] l2tp-auto-client enable
[L2TP Client-Virtual-Template1] quit
```

Configure private routes so that branches can communicate with the headquarters through the private network.

```
[L2TP Client] ip route-static 192.168.2.0 255.255.255.0 virtual-template 1
```

Step 2 Configure the LNS.

Configure an IP address for the public-network-side interface.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] interface GigabitEthernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 202.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
```

Configure an IP address for the user-side interface.

```
[LNS] interface GigabitEthernet 2/0/0
[LNS-GigabitEthernet2/0/0] ip address 192.168.2.1 255.255.255.0
[LNS-GigabitEthernet2/0/0] quit
```

Configure AAA authentication, and set the user name and password to **huawei** and **Huawei@1234** on the LNS.

```
[LNS] aaa
[LNS-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

Configure an IP address pool for the LNS and allocate an IP address to the dial-up interface of the L2TP Client.

```
[LNS] ip pool 1
[LNS-ip-pool-1] network 192.168.1.0 mask 24
[LNS-ip-pool-1] gateway-list 192.168.1.1
[LNS-ip-pool-1] quit
```

Create a virtual interface template and configure PPP negotiation parameters.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ppp authentication-mode chap
```

```
[LNS-Virtual-Template1] remote address pool 1
[LNS-Virtual-Template1] ip address 192.168.1.1 255.255.255.0
[LNS-Virtual-Template1] quit
```

Enable L2TP and configure an L2TP group.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Configure an LNS tunnel name and L2TP Client tunnel name.

```
[LNS-l2tp1] tunnel name lns
[LNS-l2tp1] allow l2tp virtual-template 1 remote L2TP Client
```

Enable the tunnel authentication function, and configure an authentication password.

```
[LNS-l2tp1] tunnel authentication
[LNS-l2tp1] tunnel password cipher huawei
[LNS-l2tp1] quit
```

On the LNS, configure a static route to the public network. For example, set the next hop address to 202.1.1.2.

```
[LNS] ip route-static 202.1.2.1 255.255.255.255 202.1.1.2
```

Configure private routes so that the headquarters can communicate with branches through the private network.

```
[LNS] ip route-static 192.168.10.0 255.255.255.0 virtual-template 1
```

Step 3 Verify the configuration.

Run the **display l2tp tunnel** command on the L2TP Client or LNS to view L2TP tunnel and session information. The command output for the LNS is shown as an example.

```
[LNS] display l2tp tunnel

Total tunnel : 1
LocalTID RemoteTID RemoteAddress      Port   Sessions RemoteName
1          1          202.1.2.1          1701   1         L2TP Client
```

Check that PC 1 can communicate with PC 2 in the enterprise headquarters.

----End

Configuration Files

- Configuration file of the L2TP Client

```
#
sysname L2TP Client
#
l2tp enable
#
interface Virtual-Template1
 ppp chap user huawei
 ppp chap password cipher
 %^%#'&=6Q(|7-#|.]EB`mK$(h7[CY`2m)-YT)Q=Oh2~2%^%#
 ip address ppp-negotiate
 l2tp-auto-client enable
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 192.168.10.1 255.255.255.0
#
l2tp-group 1
 tunnel password cipher %@@@/-#)Lg[S4F:#2~ZNvqa$]\DL%@@@
```

```
tunnel name L2TP Client
start l2tp ip 202.1.1.1 fullusername huawei
#
ip route-static 192.168.2.0 255.255.255.0 Virtual-Template1
ip route-static 202.1.1.1 255.255.255.255 202.1.2.2
#
return
```

- Configuration file of the LNS

```
#
sysname LNS
#
l2tp enable
#
ip pool 1
network 192.168.1.0 mask 255.255.255.0
gateway-list 192.168.1.1
#
aaa
local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F
\H0Qv8B]KAZja3}3q'RNx;VI%^%#
local-user huawei privilege level 0
local-user huawei service-type ppp
#
interface Virtual-Template1
ppp authentication-mode chap
remote address pool 1
ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet1/0/0
ip address 202.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 192.168.2.1 255.255.255.0
#
l2tp-group 1
allow l2tp virtual-template 1 remote L2TP Client
tunnel password cipher %@@EB~j7Je>;@>uNr''D=J<] \WL%@@
tunnel name lns
#
ip route-static 192.168.10.0 255.255.255.0 Virtual-Template1
ip route-static 202.1.2.1 255.255.255.255 202.1.1.2
#
return
```

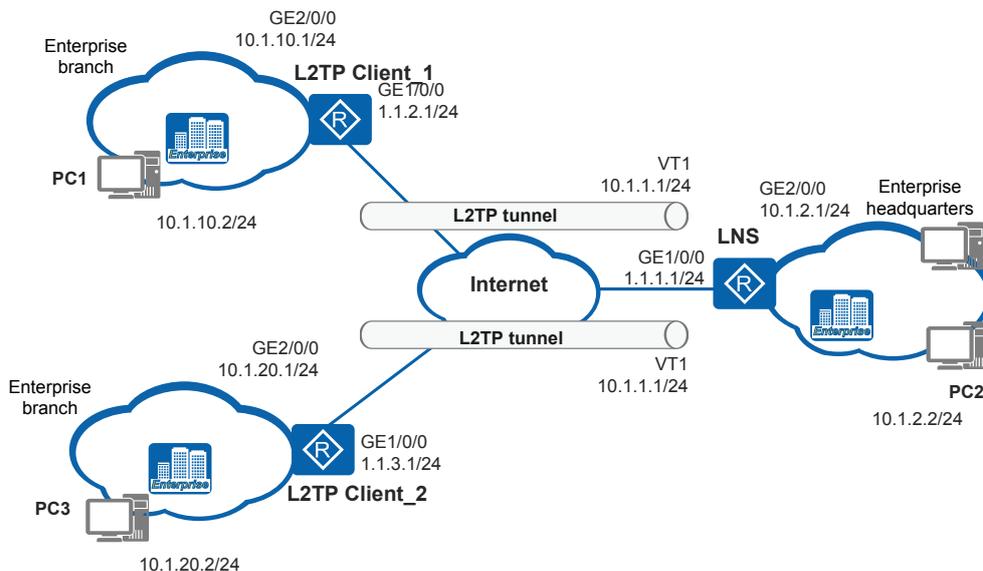
1.8.5 Example for Configuring L2TP Client-Initiated L2TP Connections

Networking Requirements

As shown in [Figure 1-23](#), an enterprise has some branches located in other cities, and branches use the Ethernet network.

The headquarters network provides VPDN services for the branch staff to allow them to access the network of the headquarters. The LNS only authenticates the L2TP Client. The L2TP Client dials up to establish L2TP connections to the LNS.

Figure 1-23 Networking diagram for establishing L2TP Client-Initiated L2TP connections



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable L2TP on the L2TP Client. The virtual PPP user sends a connection request to the server in the headquarters over an L2TP tunnel. After the PPP user is authenticated, a tunnel is set up.
2. On the L2TP Client, configure a reachable route to the LNS and the enable the dial-up function.
3. On the LNS, configure L2TP, a virtual PPP user, and a route to the public network segment.

Procedure

Step 1 Configure the L2TP Client_1.

Configure an IP address for the public-network-side interface.

```
<Huawei> system-view
[Huawei] sysname L2TP Client_1
[L2TP Client_1] interface gigabitethernet 1/0/0
[L2TP Client_1-GigabitEthernet1/0/0] ip address 1.1.2.1 255.255.255.0
[L2TP Client_1-GigabitEthernet1/0/0] quit
```

Configure an IP address for the user-side interface.

```
[L2TP Client_1] interface gigabitethernet 2/0/0
[L2TP Client_1-GigabitEthernet2/0/0] ip address 10.1.10.1 255.255.255.0
[L2TP Client_1-GigabitEthernet2/0/0] quit
```

Enable L2TP globally, create an L2TP group, and configure the user **huawei** to establish an L2TP connection to the LNS.

```
[L2TP Client_1] l2tp enable
[L2TP Client_1] l2tp-group 1
[L2TP Client_1-12tp1] tunnel name L2TP Client_1
[L2TP Client_1-12tp1] start l2tp ip 1.1.1.1 fullusername huawei
```

Enable tunnel authentication and set the tunnel password.

```
[L2TP Client_1-12tp1] tunnel authentication
[L2TP Client_1-12tp1] tunnel password cipher huawei
[L2TP Client_1-12tp1] quit
```

Configure the user name and password, authentication mode, and IP address for the virtual PPP user.

```
[L2TP Client_1] interface virtual-template 1
[L2TP Client_1-Virtual-Template1] ppp chap user huawei
[L2TP Client_1-Virtual-Template1] ppp chap password cipher Huawei@1234
[L2TP Client_1-Virtual-Template1] ip address ppp-negotiate
[L2TP Client_1-Virtual-Template1] ospf p2mp-mask-ignore
[L2TP Client_1-Virtual-Template1] quit
```

On the LNS, configure a static route to the public network. For example, set the next hop address to 1.1.2.2.

```
[L2TP Client_1] ip route-static 1.1.1.1 255.255.255.255 1.1.2.2
```

Enable the L2TP Client to dial up and establish an L2TP tunnel.

```
[L2TP Client_1] interface virtual-template 1
[L2TP Client_1-Virtual-Template1] l2tp-auto-client enable
[L2TP Client_1-Virtual-Template1] quit
```

Configure private routes so that branches can communicate with the headquarters through the private network.

```
[L2TP Client_1] ospf 10
[L2TP Client_1-ospf-10] area 0
[L2TP Client_1-ospf-10-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[L2TP Client_1-ospf-10-area-0.0.0.0] network 10.1.10.0 0.0.0.255
[L2TP Client_1-ospf-10-area-0.0.0.0] quit
[L2TP Client_1-ospf-10] quit
```

Step 2 Configure the L2TP Client_2.

Configure an IP address for the public-network-side interface.

```
<Huawei> system-view
[Huawei] sysname L2TP Client_2
[L2TP Client_2] interface gigabitethernet 1/0/0
[L2TP Client_2-GigabitEthernet1/0/0] ip address 1.1.3.1 255.255.255.0
[L2TP Client_2-GigabitEthernet1/0/0] quit
```

Configure an IP address for the user-side interface.

```
[L2TP Client_2] interface gigabitethernet 2/0/0
[L2TP Client_2-GigabitEthernet2/0/0] ip address 10.1.20.1 255.255.255.0
[L2TP Client_2-GigabitEthernet2/0/0] quit
```

Enable L2TP globally, create an L2TP group, and configure the user **huawei** to establish an L2TP connection to the LNS.

```
[L2TP Client_2] l2tp enable
[L2TP Client_2] l2tp-group 1
[L2TP Client_2-12tp1] tunnel name L2TP Client_2
[L2TP Client_2-12tp1] start l2tp ip 1.1.1.1 fullusername huawei
```

Enable tunnel authentication and set the tunnel password.

```
[L2TP Client_2-12tp1] tunnel authentication
[L2TP Client_2-12tp1] tunnel password cipher huawei
[L2TP Client_2-12tp1] quit
```

Configure the user name and password, authentication mode, and IP address for the virtual PPP user.

```
[L2TP Client_2] interface virtual-template 1
[L2TP Client_2-Virtual-Template1] ppp chap user huawei
[L2TP Client_2-Virtual-Template1] ppp chap password cipher Huawei@1234
[L2TP Client_2-Virtual-Template1] ip address ppp-negotiate
[L2TP Client_2-Virtual-Template1] ospf p2mp-mask-ignore
[L2TP Client_2-Virtual-Template1] quit
```

On the LNS, configure a static route to the public network. For example, set the next hop address to 1.1.3.2.

```
[L2TP Client_2] ip route-static 1.1.1.1 255.255.255.255 1.1.3.2
```

Enable the L2TP Client to dial up and establish an L2TP tunnel.

```
[L2TP Client_2] interface virtual-template 1
[L2TP Client_2-Virtual-Template1] l2tp-auto-client enable
[L2TP Client_2-Virtual-Template1] quit
```

Configure private routes so that branches can communicate with the headquarters through the private network.

```
[L2TP Client_2] ospf 10
[L2TP Client_2-ospf-10] area 0
[L2TP Client_2-ospf-10-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[L2TP Client_2-ospf-10-area-0.0.0.0] network 10.1.20.0 0.0.0.255
[L2TP Client_2-ospf-10-area-0.0.0.0] quit
[L2TP Client_2-ospf-10] quit
```

Step 3 Configure the LNS.

Configure an IP address for the public-network-side interface.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] interface gigabitethernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 1.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
```

Configure an IP address for the user-side interface.

```
[LNS] interface gigabitethernet 2/0/0
[LNS-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[LNS-GigabitEthernet2/0/0] quit
```

Configure AAA authentication, and set the user name and password to **huawei** and **Huawei@1234** on the LNS.

```
[LNS] aaa
[LNS-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

Configure an IP address pool for the LNS and allocate an IP address to the dial-up interface of the L2TP Client.

```
[LNS] ip pool 1
[LNS-ip-pool-1] network 10.1.1.0 mask 24
[LNS-ip-pool-1] gateway-list 10.1.1.1
[LNS-ip-pool-1] quit
```

Create a virtual interface template and configure PPP negotiation parameters.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ppp authentication-mode chap
[LNS-Virtual-Template1] remote address pool 1
[LNS-Virtual-Template1] ip address 10.1.1.1 255.255.255.0
[LNS-Virtual-Template1] ospf network-type p2mp
[LNS-Virtual-Template1] ospf timer hello 10
[LNS-Virtual-Template1] ospf p2mp-mask-ignore
[LNS-Virtual-Template1] quit
```

Enable L2TP and configure an L2TP group.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Configure an LNS tunnel name and L2TP Client tunnel name.

```
[LNS-l2tp1] tunnel name lns
[LNS-l2tp1] allow l2tp virtual-template 1
```

Enable the tunnel authentication function, and configure an authentication password.

```
[LNS-l2tp1] tunnel authentication
[LNS-l2tp1] tunnel password cipher huawei
[LNS-l2tp1] quit
```

On the LNS, configure a static route to the public network. For example, set the next hop address to 1.1.1.2.

```
[LNS] ip route-static 1.1.2.1 255.255.255.255 1.1.1.2
[LNS] ip route-static 1.1.3.1 255.255.255.255 1.1.1.2
```

Configure private routes so that the headquarters can communicate with branches through the private network.

```
[LNS] ospf 10
[LNS-ospf-10] area 0
[LNS-ospf-10-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[LNS-ospf-10-area-0.0.0.0] network 10.1.2.0 0.0.0.255
[LNS-ospf-10-area-0.0.0.0] quit
[LNS-ospf-10] quit
```

Step 4 Verify the configuration.

Run the **display l2tp tunnel** command on the L2TP Client or LNS to view L2TP tunnel and session information in **RemoteName** and **Sessions**. The command output for the LNS is shown as an example.

```
[LNS] display l2tp tunnel

Total tunnel : 2
LocalTID RemoteTID RemoteAddress Port Sessions RemoteName
1 1 1.1.2.1 1701 1 L2TP Client_1
2 1 1.1.3.1 1701 1 L2TP Client_2
```

Check that PC1 and PC3 can communicate with PC2 in the enterprise headquarters.

----End

Configuration Files

- Configuration file of the L2TP Client_1

```
#
sysname L2TP Client_1
#
l2tp enable
#
interface Virtual-Template1
ppp chap user huawei
ppp chap password cipher %^%#'&=6Q(|7-#|.]EB`mK$(h7[CY`2m)-YT)Q=Oh2~2%^%#
ip address ppp-negotiate
l2tp-auto-client enable
ospf p2mp-mask-ignore
#
interface GigabitEthernet1/0/0
ip address 1.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.10.1 255.255.255.0
#
l2tp-group 1
tunnel password cipher %@@@/-#)Lg[S4F:#2~ZNvqa$]\DL%@@@
tunnel name L2TP Client_1
start l2tp ip 1.1.1.1 fullusername huawei
#
ospf 10
area 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.1.10.0 0.0.0.255
#
ip route-static 1.1.1.1 255.255.255.255 1.1.2.2
#
return
```

● Configuration file of the L2TP Client_2

```
#
sysname L2TP Client_2
#
l2tp enable
#
interface Virtual-Template1
ppp chap user huawei
ppp chap password cipher %^%#'&=6Q(|7-#|.]EB`mK$(h7[CY`2m)-YT)Q=Oh2~2%^%#
ip address ppp-negotiate
l2tp-auto-client enable
ospf p2mp-mask-ignore
#
interface GigabitEthernet1/0/0
ip address 1.1.3.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.20.1 255.255.255.0
#
l2tp-group 1
tunnel password cipher %@@@6Za[BAw}f$WX`sX`]:QP1%.t%@@@
tunnel name L2TP Client_2
start l2tp ip 1.1.1.1 fullusername huawei
#
ospf 10
area 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.1.20.0 0.0.0.255
#
ip route-static 1.1.1.1 255.255.255.255 1.1.3.2
#
return
```

● Configuration file of the LNS

```
#
sysname LNS
#
```

```
l2tp enable
#
ip pool 1
 gateway-list 10.1.1.1
 network 10.1.1.0 mask 255.255.255.0
#
aaa
 local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F
 \H0Qv8B]KAZja3}3q'RNx;VI%^%#
 local-user huawei privilege level 0
 local-user huawei service-type ppp
#
interface Virtual-Template1
 ppp authentication-mode chap
 remote address pool 1
 ip address 10.1.1.1 255.255.255.0
 ospf network-type p2mp
 ospf timer hello 10
 ospf p2mp-mask-ignore
#
interface GigabitEthernet1/0/0
 ip address 1.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
l2tp-group 1
 allow l2tp virtual-template 1
 tunnel password cipher %@@@EB~j7Je>;@>uNr' 'D=J<] \WL%@@@
 tunnel name lns
#
ospf 10
 area 0.0.0.0
 network 10.1.1.0 0.0.0.255
 network 10.1.2.0 0.0.0.255
#
ip route-static 1.1.2.1 255.255.255.255 1.1.1.2
ip route-static 1.1.3.1 255.255.255.255 1.1.1.2
#
return
```

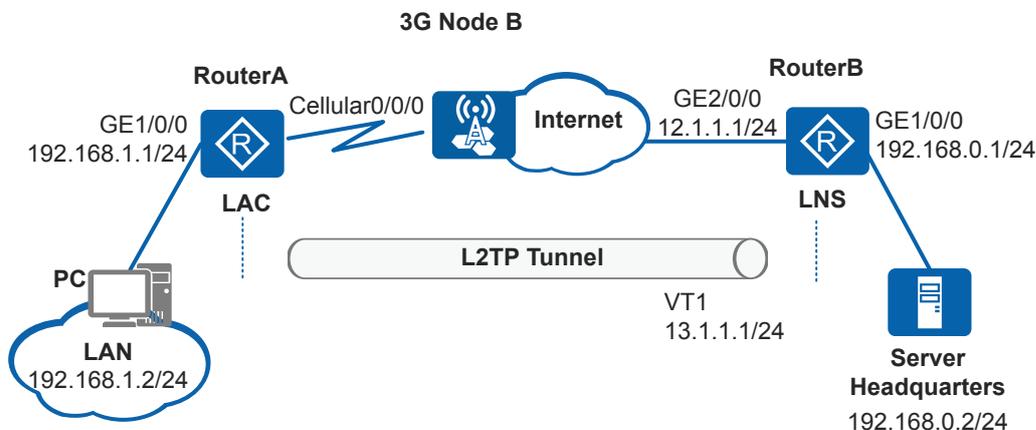
1.8.6 Example for Configuring L2TP Client-Initiated L2TP Connections Using the 3G Interface

Networking Requirements

As shown in [Figure 1-24](#), an enterprise has some branches located in other cities, and its branches use the Ethernet network and have gateways deployed, so that branch hosts can access the Internet.

The headquarters provides VPDN services for the branch staff to allow any staff to access the network of the headquarters. The LNS only authenticates the L2TP Client. The L2TP Client dials up to establish L2TP connections between the L2TP Client and LNS.

Figure 1-24 Networking diagram for L2TP Client-Initiated L2TP connections using the 3G interface



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a dial string for dialup on a 3G interface and a route to the public network address.
2. Enable L2TP on the L2TP Client. The virtual PPP user sends a connection request to the server in the headquarters over an L2TP tunnel. After the PPP user is authenticated, a tunnel is set up.
3. Configure a route to the public network address with the 3G interface as the outbound interface, and enable the dial function on the L2TP Client.
4. On the LNS, configure L2TP, a virtual PPP user, and a route to the public network segment.

Procedure

Step 1 Configure RouterA (the L2TP Client side).

In this example, the IP address of Cellular0/0/0 on RouterA is allocated by the ISP, and the IP address of GE2/0/0 on RouterB is 12.1.1.1.

Configure dialup on Cellular0/0/0.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] dialer-rule
[RouterA-dialer-rule] dialer-rule 1 ip permit
[RouterA-dialer-rule] quit
[RouterA] interface cellular 0/0/0
[RouterA-Cellular0/0/0] link-protocol ppp
[RouterA-Cellular0/0/0] ip address ppp-negotiate
[RouterA-Cellular0/0/0] dialer enable-circular
[RouterA-Cellular0/0/0] dialer-group 1
[RouterA-Cellular0/0/0] dialer timer autodial 60
```

```
[RouterA-Cellular0/0/0] dialer number *99# autodial
[RouterA-Cellular0/0/0] mode wcdma wcdma-precedence
[RouterA-Cellular0/0/0] quit
[RouterA] apn profile 3gprofile
[RouterA-apn-profile-3gprofile] apn 3GNET
[RouterA-apn-profile-3gprofile] quit
[RouterA] interface cellular 0/0/0
[RouterA-Cellular0/0/0] apn-profile 3gprofile
[RouterA-Cellular0/0/0] shutdown
[RouterA-Cellular0/0/0] undo shutdown
[RouterA-Cellular0/0/0] quit
```

Configure an IP address for the public-network-side interface.

```
[RouterA] interface gigabitEthernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 192.168.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
```

Configure an L2TP group and its attributes.

```
[RouterA] l2tp enable
[RouterA] l2tp-group 1
[RouterA-l2tp1] tunnel name L2TP Client
[RouterA-l2tp1] start l2tp ip 12.1.1.1 fullusername huawei
```

Enable tunnel authentication and set the tunnel password.

```
[RouterA-l2tp1] tunnel authentication
[RouterA-l2tp1] tunnel password cipher 123
[RouterA-l2tp1] quit
```

Configure the user name and password, authentication mode, and IP address for the virtual PPP user.

```
[RouterA] interface virtual-template 1
[RouterA-Virtual-Template1] ppp chap user huawei
[RouterA-Virtual-Template1] ppp chap password ciphe Huawei@1234
[RouterA-Virtual-Template1] ip address 13.1.1.2 255.255.255.0
[RouterA-Virtual-Template1] quit
```

Configure a public route so that the packets sent to the headquarters are forwarded through the 3G interface.

```
[RouterA] ip route-static 0.0.0.0 0 Cellular0/0/0
```

Enable the L2TP Client to establish an L2TP tunnel.

```
[RouterA] interface virtual-template 1
[RouterA-virtual-template1] l2tp-auto-client enable
[RouterA-virtual-template1] quit
```

Configure private routes so that branches can communicate with the headquarters through the private network.

```
[RouterA] ip route-static 192.168.0.0 255.255.255.0 virtual-template 1
```

Step 2 Configure RouterB (the LNS side).

Assign an IP address to GigabitEthernet2/0/0 on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitEthernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 12.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a private IP address.

```
[RouterB] interface GigabitEthernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 192.168.0.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
```

Create and configure a virtual template.

```
[RouterB] interface virtual-template 1
[RouterB-Virtual-Template1] ppp authentication-mode chap
[RouterB-Virtual-Template1] ip address 13.1.1.1 255.255.255.0
[RouterB-Virtual-Template1] quit
```

Enable L2TP and configure an L2TP group.

```
[RouterB] l2tp enable
[RouterB] l2tp-group 1
```

Set the local and remote tunnel names for the LNS.

```
[RouterB-l2tp1] tunnel name LNS
[RouterB-l2tp1] allow l2tp virtual-template 1 remote L2TP Client
```

Enable tunnel authentication and set the tunnel password.

```
[RouterB-l2tp1] tunnel authentication
[RouterB-l2tp1] tunnel password cipher 123
[RouterB-l2tp1] quit
```

Set the user name and password to **huawei** and **Huawei@1234**, which must be the same as those on the L2TP Client side.

```
[RouterB] aaa
[RouterB-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[RouterB-aaa] local-user huawei service-type ppp
[RouterB-aaa] quit
```

Configure an IP address and a route to the Internet. For example, set the next hop address to the Internet to 12.1.1.2.

```
[RouterB] ip route-static 0.0.0.0 0 12.1.1.2
```

Configure private routes so that the headquarters can communicate with branches through the private network.

```
[RouterB] ip route-static 192.168.1.0 255.255.255.0 virtual-template 1
```

Step 3 Verify the configuration.

Run the **display l2tp tunnel** command on the L2TP Client and LNS. You can see that a tunnel has been established. The command output on the L2TP Client is used as an example.

```
[RouterA] display l2tp tunnel

Total tunnel : 1
LocalTID RemoteTID RemoteAddress      Port   Sessions RemoteName
1         1         12.1.1.1          1701   1         LNS
```

Run the **display l2tp session** command to check the session status. The command output on the LNS is used as an example.

```
[RouterB] display l2tp session
```

```
Total session : 1
LocalSID RemoteSID LocalTID
1 1 1
```

Check that PCs in the branch can access servers in the headquarters.

---End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#

l2tp
enable
#

interface Virtual-
Template1
 ppp chap user huawei
 ppp chap password cipher %^%#'&=6Q(|7-#|.]EB`mK$(h7[CY`2m)-YT)Q=Oh2~2%^%#

 ip address 13.1.1.2
255.255.255.0
 l2tp-auto-client
enable
#

interface
Cellular0/0/0
 link-protocol
ppp
 ip address ppp-
negotiate
 dialer enable-
circular
 dialer-group 1
 apn-profile 3GNET
 dialer timer autodial
60
 dialer number *99#
autodial
#

interface
GigabitEthernet1/0/0
 ip address 192.168.1.1
255.255.255.0
#

l2tp-group
1
 tunnel password cipher %@@@o6Xpp(i/i:WRC)`'0#3nJ*%@@@
 tunnel name L2TP
Client
 start l2tp ip 12.1.1.1 fullusername
huawei
#
 dialer-rule
 dialer-rule 1 ip permit
#
 apn profile 3GNET
#
 ip route-static 0.0.0.0 0.0.0.0 Cellular0/0/0
 ip route-static 192.168.0.0 255.255.255.0 Virtual-Template1
#
```

```
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
l2tp
enable
#
aaa
local-user huawei password cipher %^%#_<`.CO&(:LeS/$#F
\H0Qv8B]KAZja3}3q'RNx;VI%^%#
local-user huawei privilege level 0
local-user huawei service-type
ppp
#
interface Virtual-
Template1
 ppp authentication-mode
 chap
 ip address 13.1.1.1
255.255.255.0
#
interface
GigabitEthernet1/0/0
 ip address 192.168.0.1
255.255.255.0
#
interface
GigabitEthernet2/0/0
 ip address 12.1.1.1
255.255.255.0
#
l2tp-group
1
 allow l2tp virtual-template 1 remote L2TP
Client
 tunnel password cipher %@@@5j*=S&AGXK'J}kG])REK]_-o%@@@
 tunnel name
LNS
#
ip route-static 0.0.0.0 0.0.0.0 12.1.1.2
ip route-static 192.168.1.0 255.255.255.0 Virtual-Template1
#
return
```

1.9 Troubleshooting L2TP

This section describes common faults caused by incorrect L2TP configurations.

1.9.1 User Failed to Dial Up to the LNS

Fault Description

A remote user uses the built-in dial-up software of Windows XP to initiate L2TP connections to the LNS in the enterprise headquarters. However, the dialup fails.

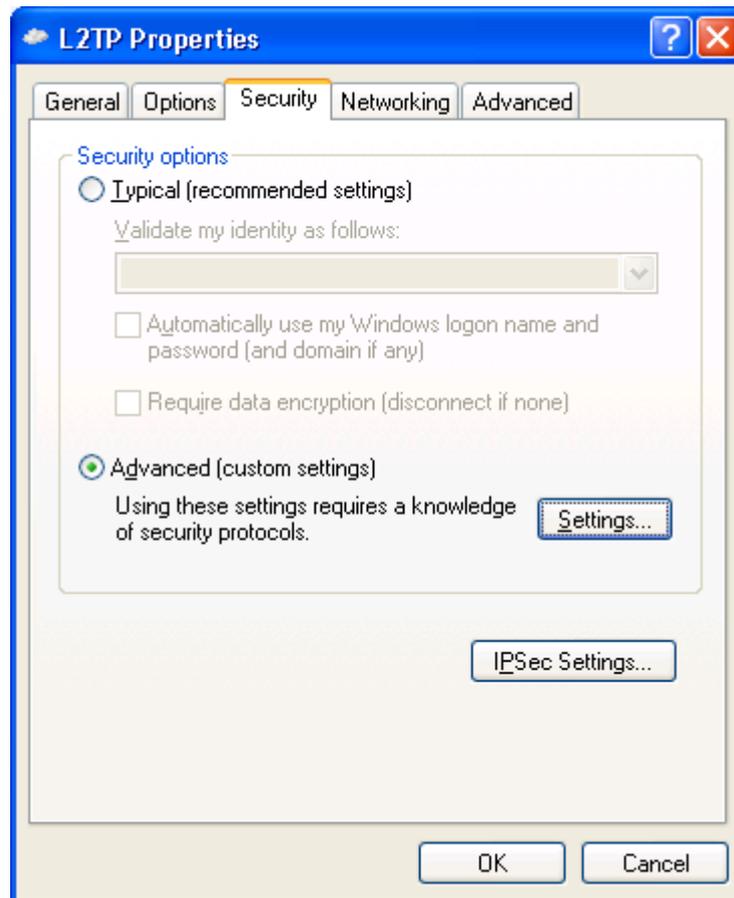
Procedure

Before handling this fault, ensure that a reachable route exists between the remote user and the LNS.

The possible causes are:

- The user information differs from that configured on the LNS.
In the AAA view, you can check user information configured on the LNS.
- The IP address pool on the LNS is incorrectly configured.
No gateway address is specified in the IP address pool or the specified gateway address is incorrect. Run the **gateway-list ip-address &<1-8>** command to specify the IP address of the virtual template interface as the gateway address.
- The authentication modes on the PC and the LNS are different.
Run the **display l2tp-group [group-number]** command on the LNS to check the **TunnelAuth** field to see whether tunnel authentication is enabled. Windows XP does not support tunnel authentication. If the remote user supports tunnel authentication, check whether the authentication password of the user is the same as that configured in the L2TP group view.
- PPP negotiation parameters are not consistent on the two ends.
In the VT interface view, check whether the PPP authentication mode configured on the LNS is **pap** or **chap**. Run the **display l2tp-group [group-number]** command to check the **ForceChap** field to see whether mandatory CHAP authentication is enabled. If the mandatory CHAP authentication is enabled, the authentication mode must be **chap** in the VT interface view. Check the L2TP connection attributes on the PC to ensure that **PAP** and **CHAP** are selected.

Figure 1-25 L2TP connection attributes





1.9.2 Data Transmission Fails After L2TP Connections Are Established

Symptoms

An L2TP connection is established, but data transmission fails. The remote user cannot ping IP addresses on the private network segment in the enterprise headquarters.

Troubleshooting Procedure

The possible causes and corresponding troubleshooting methods are as follows:

- No route to the private network segment in the enterprise headquarters is configured on the LNS.
Run the **display ip routing-table** command on the LNS to check the routing table.
- Network congestion occurs.
L2TP is UDP-based, but UDP does not perform error control on packets. When L2TP is enabled on an unstable network, the ping to the remote end may fail.
- The LNS cannot identify the format of received PPP packets after the PPP negotiation between the LAC and remote user.

Generally, the LNS can identify the format of received PPP packets after the PPP negotiation between the LAC and remote user. If PPP packets after the PPP negotiation

between the LAC of some vendors and remote user are compressed, the LNS may not be able to identify these packets. In this case, the remote user cannot ping IP addresses on the private network segment in the enterprise headquarters. If this problem occurs, run the **mandatory-lcp** command on the LNS to enable LCP renegotiation. The remote user and LNS can then start PPP negotiation again, and the LNS can identify PPP packets sent by the remote user.

- The NAT ALG function is disabled for DNS.

Run the **display nat alg** command on the device to check whether NAT ALG is enabled for DNS.

```
[Huawei]display nat alg
NAT Application Level Gateway
Information:
-----
Application
Status
-----
dns
Disabled
ftp
Disabled
rtsp
Enabled
sip
Disabled
pptp
Disabled
```

If NAT ALG is disabled, run the **nat alg** command to enable it.

1.10 FAQ About L2TP

This section describes the FAQ about L2TP.

1.10.1 Starting from Which Version Does the device Support NAT Traversal in L2TP?

Starting from V200R002C00SPC200, the device supports this function.

1.10.2 L2TP Dialup Is Successful After Dozens of Attempts and Error 691 Is Displayed. Why?

After L2TP is configured on the device, users can dial up successfully after several attempts and error 691 is displayed during the attempts. The cause is that the device supports only 16-byte challenge messages. When challenge messages are not of 16 bytes, CHAP authentication fails and error 691 indicating user name or password error is displayed. Configure L2TP renegotiation so that the LNS and client can negotiate the 16-byte challenge messages.

1.10.3 How Can I Quickly Locate Why the LAC Cannot Set Up an L2TP Tunnel with the LNS?

When configuring the L2TP function, the LAC cannot set up a tunnel with the LNS. How can I quickly locate the fault?

1. Run the **start l2tp** command on the LAC to check whether there is a reachable route to the LNS. If no, configure a reachable route to the LNS.
2. Check the L2TP configuration on the LNS and delete the parameter *remote* specified in the **allow l2tp** command. If an L2TP tunnel can be established successfully, the LAC cannot set up a tunnel with the LNS because the tunnel name on the LAC is incorrect or the tunnel name specified by the LNS is incorrect. Use either of the following methods to solve this problem:
 - Run the **tunnel name** command on the LAC to set the local tunnel name to the value of the parameter *remote* specified by the **allow l2tp** command on the LNS.
 - Run the **allow l2tp** command on the LNS to change the value of the parameter *remote* to the tunnel name configured on the LAC. If no local tunnel name is configured using the **tunnel name** command on the LAC, the value of the parameter *remote* is the device name of the LAC.

1.10.4 How Do I Configure the LNS That Trusts the LAC Not to Perform Second Authentication on Remote Users?

The LAC authenticates access users. After the users are authenticated, the LAC sends authentication information to the LNS which determines whether the users are valid users.

If the LAC is trusted by the LNS, you can run the **authentication-mode none** command in the authentication scheme view of the LNS to set the authentication mode to non-authentication. If the command is configured, the LNS does not perform second authentication on remote users.

1.10.5 What Can I Do If a PC Running the Windows 7 or XP Operating System Fails to Establish an L2TP over IPsec Tunnel with the Device?

The possible cause for an L2TP over IPsec tunnel establishment failure between a PC running the Windows 7 or XP operating system and the device is that the system registry is not modified.

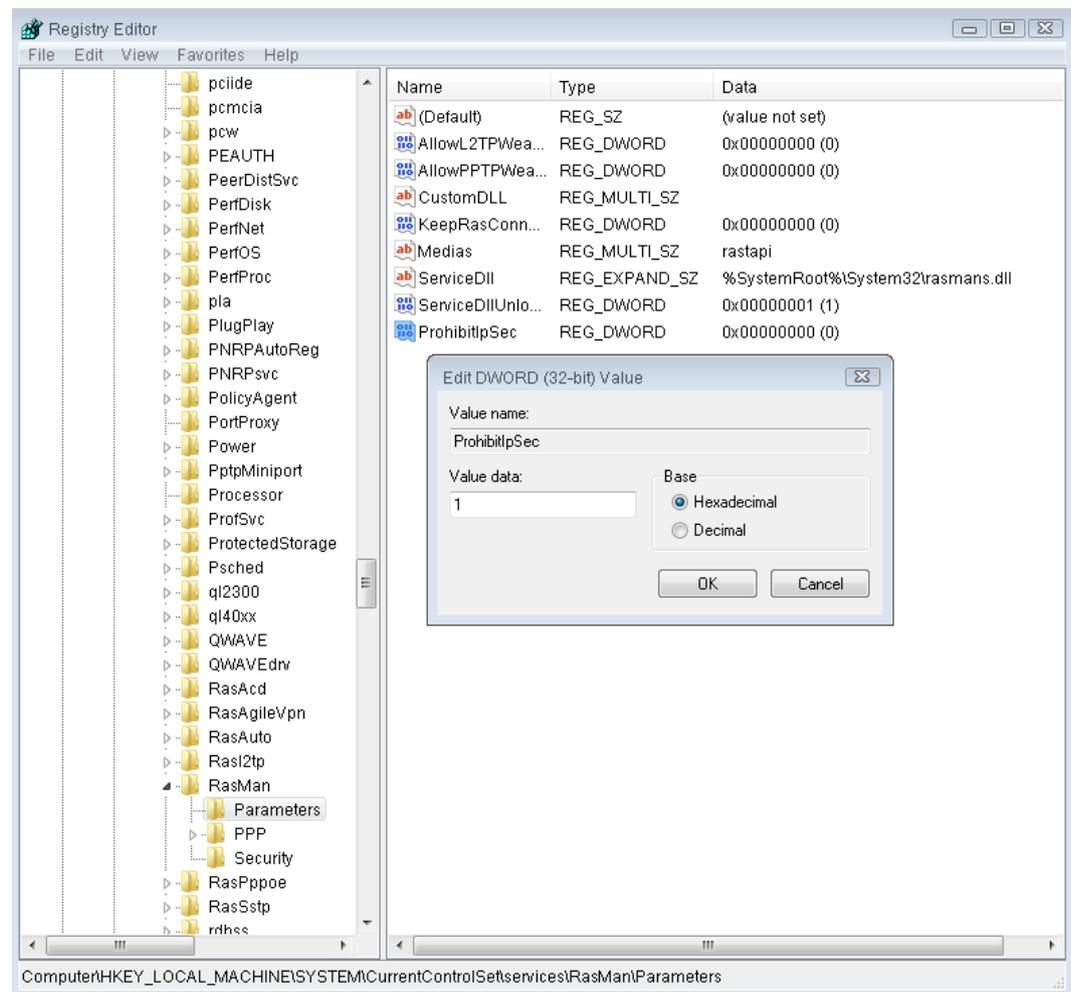
The following describes how to modify the registry of the Windows 7 operating system so as not to use digital certificate authentication.

NOTE

The operations in the 64-bit Windows 7 operating system are the same as those in the 32-bit operating system.

1. Choose **Start > cmd** and enter **regedit** to open the registry.
2. Choose **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\RasMan\Parameters** in the left, right-click a blank area in the right, and choose **New > DWORD (32-bit) Value** to generate the **New Value #1** file.
3. Right-click **New Value #1** and choose **Rename** to rename the file as **ProhibitIpSec**.
4. Right-click the **ProhibitIpSec** file and choose **Modify**.
5. In the displayed **Edit DWORD (32-bit) Value** dialog box, set **Value data** to **1** and select **Hexadecimal** under **Base**, as shown in [Figure 1-26](#).
6. Restart the PC to make the configuration take effect.

Figure 1-26 Registry Editor



1.11 References for L2TP

This section lists references for L2TP.

The following table lists the references for L2TP.

Document	Description
RFC 2661	Layer Two Tunneling Protocol (L2TP)

2 L2TPv3 Configuration

About This Chapter

This section describes how to configure L2TPv3 tunnels on IP network to implement transparent transmission of Layer 2 data over Layer 3 networks.

- [2.1 Overview of L2TPv3](#)
- [2.2 Understanding L2TPv3](#)
- [2.3 Application Scenarios for L2TPv3](#)
- [2.4 Licensing Requirements and Limitations for L2TPv3](#)
- [2.5 Configuring L2TPv3](#)
- [2.6 Monitoring the L2TPv3 Tunnel Running Status](#)
- [2.7 Configuration Examples for L2TPv3](#)
- [2.8 References for L2TPv3](#)

2.1 Overview of L2TPv3

Definition

Layer 2 Tunneling Protocol - Version 3 (L2TPv3) sets up Layer 2 access links to transparently transmit Layer 2 packets, such as Point-to-Point Protocol (PPP), Ethernet, High-Level Data Link Control (HDLC), and asynchronous transfer mode (ATM) packets, over a packet switched network (PSN).

NOTE

Currently, L2TPv3 feature of the router can transparently transmit only Ethernet packets, and user-side interfaces must be VLANIF interfaces, wide area network (WAN) interfaces, or sub-interfaces.

Purpose

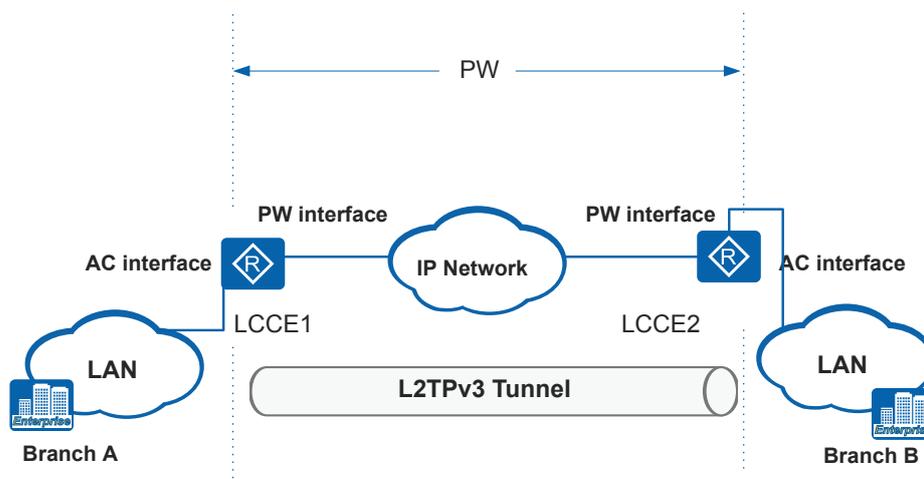
Virtual leased lines (VLL) can be deployed to set up Layer 2 connections between the headquarters and branch or between branches. However, the VLL deployment cost is high.

L2TPv3 can set up Layer 2 connections between enterprise branches and the headquarters or between branches, without a need to reconstruct the existing IP networks. Compared with the VLL solution, L2TPv3 helps carriers lower network construction costs, so that enterprises can obtain network services at lower expenses.

2.2 Understanding L2TPv3

In [Figure 2-1](#), enterprise branch LANs need to exchange Layer 2 data over the IP network; therefore, L2TPv3 is configured on the egress gateways.

Figure 2-1 L2TPv3 tunnel



Concepts

- **LCCE**
An L2TP Control Connection Endpoint (LCCE) is a node at either end of an L2TP control connection tunnel (L2TPv3 tunnel). An LCCE can be an L2TP access concentrator (LAC) or an L2TP network server (LNS). The LCCE is an LAC if frames to be forwarded over the tunnel are processed at the data link layer and is an LNS if the frames are processed at the network layer.
- **PW**
A pseudo wire (PW) is a directly connected virtual data channel between two AC interfaces, used to transparently transmit Layer 2 data. Each L2TPv3 session corresponds to a PW.
- **AC interface**
An AC interface is connected to a user-side device, to receive and forward user-side traffic. In this document, only Layer 3 Ethernet interfaces, including WAN interfaces, sub-interfaces (not in selective QinQ mode), and VLANIF interfaces can be used as AC interfaces.
- **PW interface**
A PW interface is connected to the remote LCCE, to receive and forward L2TPv3 packets from the network side.

- Static tunnel

A static tunnel is established by manually configuring local and remote parameters. Data is directly forwarded over a static tunnel without the packet negotiation process.

Only one tunnel can be established on an interface, and one tunnel supports only one session. Multiple tunnels can be created at different interfaces of a device.

Working Process

In **Figure 2-1**, the gateways of enterprise branch A and branch B are LCCE1 and LCCE2 respectively. To establish an L2TPv3 tunnel between LCCE1 and LCCE2, perform the following operations:

1. Enable L2TPv3 globally on the LCCEs.
2. Create a tunnel interface on each LCCE. Set the tunneling protocol to L2TPv3 and the working mode to static. Configure the tunnel source address, tunnel destination address, session ID and other parameters.

 **NOTE**

An L2TPv3 tunnel can be established only when the same parameters are configured on the two ends.

3. Bind an AC interface to the tunnel interface on each LCCE.
4. The AC interface forwards traffic to the remote device through the L2TPv3 tunnel.

L2TPv3 Packet Format

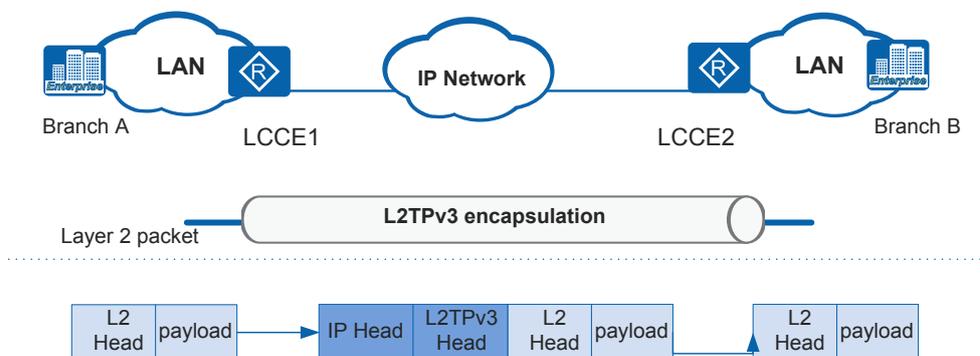
Figure 2-2 shows the L2TPv3 encapsulation format.

Figure 2-2 L2TPv3 packet format



L2TPv3 Packet Encapsulation

Figure 2-3 L2TPv3 packet encapsulation



In [Figure 2-3](#), packets sent from branch A to branch B are forwarded as follows:

1. Branch A sends a data packet to branch B.
2. LCCE1 receives the packet and adds VLAN encapsulation to it on the AC interface. The AC interface then adds an L2TPv3 header to the data packet based on the tunnel encapsulation table, and forwards the packet through the tunnel interface based on the routing table.
3. LCCE2 receives the packet and checks whether it is an L2TPv3 packet. If the packet is an L2TPv3 packet, LCCE2 searches the tunnel decapsulation table and checks whether the local parameters of the LCCE1 are the same as the remote parameters of the LCCE2. If they are the same, LCCE2 removes the L2TPv3 header. The AC interface on LCCE2 then processes the packet based on the VLAN encapsulation rule and forwards it to branch B. If the packet is not an L2TPv3 packet or the local parameters of the sender are different from the remote parameters of the remote device, LCCE2 discards the packet.

Service Access Modes

When two branches in different VLANs need to communicate, the branch devices send packets to the L2TPv3 tunnel in either of the following ways:

- Directly forwarding packets: When the AC interface is a WAN interface, that is, it has been bound to the tunnel using the **link-bridge** command, the L2TPv3 tunnel can transparently transmit untagged, single-tagged, or double-tagged Layer 2 packets.
- Terminating one tag: The sub-interface of the AC interface removes the outer tag from single-tagged or double-tagged packets and adds an L2TPv3 header before sending them to the L2TPv3 tunnel. The sub-interface adds one tag to packets received from the L2TPv3 tunnel before forwarding them.
 - If the received packets contain one tag, the sub-interface removes the C-Tag.
 - If the received packets contain double tags, the sub-interface removes the S-Tag.
- Terminating double tags: The sub-interface of the AC interface removes both S-Tag and C-Tag from packets and adds an L2TPv3 header before sending them to the L2TPv3 tunnel. The sub-interface adds both S-Tag and C-Tag to packets received from the L2TPv3 tunnel before forwarding them.

For details about tag termination on sub-interfaces, see VLAN Termination Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers CLI-based Configuration - Configuration Guide - Ethernet Switching*.

2.3 Application Scenarios for L2TPv3

An enterprise has branches located in different cities. The branches deploy Ethernet networks and have gateways to allow access to the IP network. To establish Layer 2 connections between the branches as well as between the headquarters and branches, L2TPv3 can be enabled on the gateways. The gateways then become L2TP Control Connection Endpoints (LCCEs) and set up L2TPv3 tunnels to transparently transmit Ethernet packets, lowering carrier network investment.

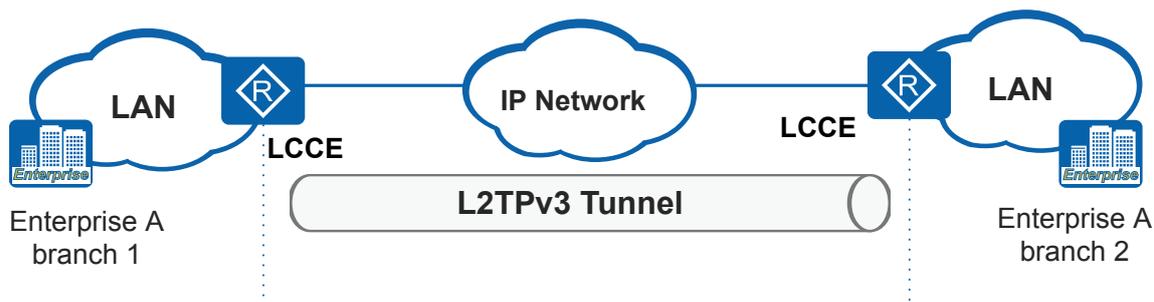
NOTE

L2TPv3 can be used to set up point-to-point (LAC-LAC) connections, but not point-to-multipoint connections.

L2TPv3 can be used only when the devices at both end of the tunnel connect to VLANs or use AC interfaces.

Only one VLAN can be configured for an L2TPv3 tunnel.

Figure 2-4 L2TPv3 tunnel setup



2.4 Licensing Requirements and Limitations for L2TPv3

Involved Network Elements

None

License Requirements

For L2TPv3-capable devices, their licensing requirements for the L2TPv3 function are as follows:

- AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S series: L2TPv3 is a basic feature of the device and is not under license control.
- AR2200-S&AR3200-S series: By default, this function is disabled on a new device. To use the L2TPv3 function, apply for and purchase the following license from the Huawei local office.
 - AR2200-S series: AR2200 value-added service package for data services
 - AR3200-S series: AR3200 value-added service package for data services

Feature Limitations

None

2.5 Configuring L2TPv3

2.5.1 Configuring a Static L2TPv3 Tunnel

Context

The Layer 2 Ethernet services are transparently transmitted over the L2TPv3 tunnel. This section describes how to create a static L2TPv3 tunnel between two LCCE devices to transmit Layer 2 data.

Procedure

Step 1 Enable the L2TPv3 function.

1. Run **system-view**
The system view is displayed.
2. Run **l2tpv3 enable**
The L2TPv3 function is enabled.
By default, the L2TPv3 function is disabled.

Step 2 Configure the L2TPv3 tunnel parameters.

1. Run **interface tunnel *interface-number***
A tunnel interface is created and the tunnel interface view is displayed.
2. Run **tunnel-protocol svpn**
A tunnel protocol on tunnel interface is set to **SVPN**.
By default, the tunnel protocol is none, indicating that packets are not encapsulated.
3. Run **encapsulation l2tpv3 [static]**
The tunnel type is set to static L2TPv3.
By default, the tunnel type is not specified.
4. Run **l2tpv3 local session-id *local-session-id***
The local session ID is specified.
By default, the local session ID is not specified.
5. Run **l2tpv3 remote session-id *remote-session-id***
The remote session ID is specified.
By default, the remote session ID is not specified.
6. (Optional) Run **l2tpv3 local cookie { key cipher *local-cookie* { length 4 plain lower-value *local-lower-value* | length 8 plain lower-value *local-lower-value* upper-value *local-high-value* } }**
The local cookie key is configured.
By default, the local cookie key is not configured.
7. (Optional) Run **l2tpv3 remote cookie { key cipher *local-cookie* { length 4 plain lower-value *local-lower-value* | length 8 plain lower-value *local-lower-value* upper-value *local-high-value* } }**
The remote cookie key is configured.

By default, the remote cookie key is not configured.

 **NOTE**

The local cookie key must be the same as the remote cookie key. If they are different, traffic cannot be forwarded.

8. Run **tunnel-source** { *source-ip-address* | *interface-type interface-number* }

A source address or source interface is specified for the tunnel interface.

By default, no source address is specified for the tunnel.

 **NOTE**

The source address of a tunnel is the IP address of the interface that sends packets. The source interface of a tunnel is the interface that sends packets.

9. Run **tunnel-destination** [**vpn-instance** *vpn-instance-name*] *dest-ip-address*

The destination address is specified for the tunnel.

By default, no destination address is specified for a tunnel.

 **NOTE**

The destination address of a tunnel is the IP address of the interface that receives packets.

10. Run **quit**

Return to the system view.

Step 3 Bind an AC interface to a tunnel interface.

1. Run **interface** *interface-type interface-number*

The AC interface view is displayed.

2. Run **link-bridge** *interface-type interface-number* { **tagged** | **raw** }

The AC interface is bound to a tunnel interface.

By default, an AC interface is not bound to a tunnel interface.

3. Run **quit**

Return to the system view.

----End

2.5.2 Verifying the L2TPv3 Configuration

Prerequisites

The configurations of an L2TPv3 tunnel are complete.

Procedure

- Run the **display interface tunnel** [*interface-number* | **main**] command to view the running status of the tunnel interface.
- Run the **display interface brief** [**main**] command to view the brief interface status and configuration information.

----End

2.6 Monitoring the L2TPv3 Tunnel Running Status

Context

To check whether an L2TPv3 tunnel is running properly, view the status of AC and tunnel interfaces.

After an AC interface is bound to a tunnel interface using the **link-bridge** command, the status of the AC and tunnel interfaces are associated. The association between the AC and tunnel interfaces are as follows:

1. If the physical status of the AC and tunnel interfaces is UP and the protocol status is UP (Spoofing), the status of the tunnel is UP.
2. If the physical status of the tunnel interface is Standby (^down), the physical status of the AC interface is down or *down.
Cause: No network cable is installed or the AC interface has been shut down.
3. If the physical status of the AC interface is Standby (^down), the protocol status of a tunnel is down.
Cause: Parameters of the tunnel are incomplete, the route is not reachable, no license is uploaded, or the tunnel has been shut down.

NOTE

If parameters are changed, a tunnel is deleted. If the new configuration is correct, a tunnel will be established again.

Procedure

- Run the **display interface tunnel** [*interface-number*] command to view the running status of the tunnel interface.
- Run the **display interface brief** [*main*] command to view the brief interface status and configuration information.

----End

2.7 Configuration Examples for L2TPv3

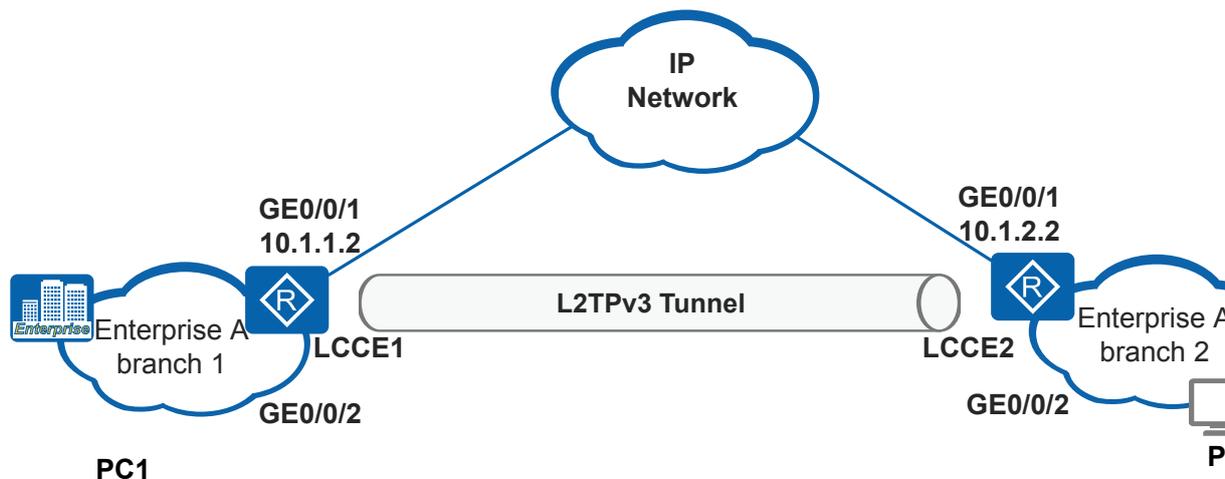
2.7.1 Example for Establishing a Static L2TPv3 Tunnel

Networking Requirements

In [Figure 2-5](#), enterprise A has two branches that connect to the IP network through LCCE1 and LCCE2 respectively. Branch 1 deploys a local area network (LAN) and uses LCCE1 as the gateway. Branch 2 deploys a LAN and uses LCCE2 as the gateway.

Branches 1 and 2 need to transparently transmit Layer 2 data through an IP network, implementing communication between LANs.

Figure 2-5 Transmitting packets over the L2TPv3 tunnel after removing one tag



Configuration Roadmap

To implement communication between branches 1 and 2 through a Layer 3 network, deploy an L2TPv3 tunnel between LCCE1 and LCCE2 to transparently transmit Layer 2 data through an IP network.

The configuration roadmap is as follows:

1. Configure a route to ensure communication between LCCE1 and LCCE2.
2. Enable the L2TPv3 function globally.
3. Establish tunnel interfaces and configure the L2TPv3 tunnel parameters.
4. Configure a Dot1q sub-interface on the AC interface and connect the Dot1q sub-interface to the L2TPv3 tunnel.
5. Configure the link bridge function to bind an AC interface to a tunnel interface.

NOTE

Upload a license to enable the L2TPv3 function.

Procedure

- Step 1** Configure IP addresses and a static route for the PW interfaces on LCCE1 and LCCE2 respectively.

Configure an IP address for the PW interface on LCCE1.

```
<Huawei> system-view
[Huawei] sysname LCCE1
[LCCE1] interface gigabitethernet 0/0/1
[LCCE1-GigabitEthernet0/0/1] ip address 10.1.1.2 24
[LCCE1-GigabitEthernet0/0/1] quit
```

Configure a static route to LCCE2 on LCCE1. This example assumes that the next hop address in the route is 10.1.1.3.

```
[LCCE1] ip route-static 10.1.2.0 255.255.255.0 10.1.1.3
```

Configure an IP address for the PW interface on LCCE2.

```
<Huawei> system-view
[Huawei] sysname LCCE2
[LCCE2] interface gigabitEthernet 0/0/1
[LCCE2-GigabitEthernet0/0/1] ip address 10.1.2.2 24
[LCCE2-GigabitEthernet0/0/1] quit
```

Configure a static route to LCCE1 on LCCE2. This example assumes that the next hop address in the route is 10.1.2.3.

```
[LCCE2] ip route-static 10.1.1.0 255.255.255.0 10.1.2.3
```

Step 2 Enable the L2TPv3 function globally.

Enable the L2TPv3 function on LCCE1.

```
[LCCE1] l2tpv3 enable
```

Enable the L2TPv3 function on LCCE2.

```
[LCCE2] l2tpv3 enable
```

Step 3 Configure L2TPv3 parameters for tunnel interfaces.

Create a tunnel on LCCE1 and configure parameters for the tunnel.

```
[LCCE1] interface tunnel 0/0/1
[LCCE1-Tunnel0/0/1] tunnel-protocol svpn
[LCCE1-Tunnel0/0/1] encapsulation l2tpv3 static
[LCCE1-Tunnel0/0/1] l2tpv3 local session-id 1
[LCCE1-Tunnel0/0/1] l2tpv3 remote session-id 4
[LCCE1-Tunnel0/0/1] l2tpv3 local cookie length 4 plain lower-value 11
[LCCE1-Tunnel0/0/1] l2tpv3 remote cookie length 4 plain lower-value 22
[LCCE1-Tunnel0/0/1] tunnel-source 10.1.1.2
[LCCE1-Tunnel0/0/1] tunnel-destination 10.1.2.2
[LCCE1-Tunnel0/0/1] quit
```

Create a tunnel on LCCE2 and configure parameters for the tunnel.

```
[LCCE2] interface tunnel 0/0/1
[LCCE2-Tunnel0/0/1] tunnel-protocol svpn
[LCCE2-Tunnel0/0/1] encapsulation l2tpv3 static
[LCCE2-Tunnel0/0/1] l2tpv3 local session-id 4
[LCCE2-Tunnel0/0/1] l2tpv3 remote session-id 1
[LCCE2-Tunnel0/0/1] l2tpv3 local cookie length 4 plain lower-value 22
[LCCE2-Tunnel0/0/1] l2tpv3 remote cookie length 4 plain lower-value 11
[LCCE2-Tunnel0/0/1] tunnel-source 10.1.2.2
[LCCE2-Tunnel0/0/1] tunnel-destination 10.1.1.2
[LCCE2-Tunnel0/0/1] quit
```

Step 4 Configure a Dot1q sub-interface on the AC interface and connect the Dot1q sub-interface to the L2TPv3 tunnel.

Create a sub-interface on LCCE1. Connect the sub-interface to the L2TPv3 tunnel as a Dot1q sub-interface.

```
[LCCE1] interface gigabitEthernet 0/0/2.1
[LCCE1-GigabitEthernet0/0/2.1] dot1q termination vid 9
```

Create a sub-interface on LCCE2. Connect the sub-interface to the L2TPv3 tunnel as a Dot1q sub-interface.

```
[LCCE2] interface gigabitEthernet 0/0/2.1
[LCCE2-GigabitEthernet0/0/2.1] dot1q termination vid 20
```

Step 5 Configure the link bridge function.

Configure the link bridge function on LCCE1 and bind an AC interface to a tunnel interface.

```
[LCCE1-GigabitEthernet0/0/2.1] link-bridge tunnel0/0/1 tagged
```

Configure the link bridge function on LCCE2 and bind an AC interface to a tunnel interface.

```
[LCCE2-GigabitEthernet0/0/2.1] link-bridge tunnel0/0/1 tagged
```

Step 6 Verify the configuration.

After the configurations are complete, run the **display interface brief** command on LCCE1 and LCCE2 to view the brief interface and IP information, including the IP addresses, subnet mask, physical and protocol status (Up or Down), and the number of interfaces in different status. The command output on LCCE1 is used as an example.

```
[LCCE1] display interface brief
PHY: Physical
*down: administratively down
(l): loopback
(s): spoofing
(b): BFD down
^down: standby
(e): ETHOAM down
InUti/OutUti: input utility/output utility
Interface          PHY   Protocol  InUti  OutUti   inErrors  outErrors
Atm8/0/0           down  down      0%     0%       0         0
Atm8/0/1           down  down      0%     0%       0         0
Atm8/0/2           down  down      0%     0%       0         0
Atm8/0/3           down  down      0%     0%       0         0
Cellular0/0/0      down  down      0%     0%       0         0
Cellular0/0/1      down  down      0%     0%       0         0
Ethernet1/0/0      up    up        0%     0%       0         0
Ethernet1/0/1      up    down     0.01%  0%       0         0
Ethernet2/0/0      down  down      0%     0%       0         0
GigabitEthernet0/0/0  up    up      0.01%  0.01%    0         0
GigabitEthernet0/0/1  up    up      0.01%  0%       0         0
GigabitEthernet0/0/2  up    up      0.01%  0%       0         0
GigabitEthernet0/0/2.1 up    up      0%     0%       0         0
GigabitEthernet0/0/3  up    down     0.01%  0%       0         0
GigabitEthernet3/0/0  down  down      0%     0%       0         0
MFR0/0/1           down  down      0%     0%       0         0
Mp-group0/0/1      down  down      0%     0%       0         0
NULL0              up    up(s)    0%     0%       0         0
Serial4/0/0        up    up      0.05%  0.05%    0         0
Serial6/0/0        down  down      0%     0%       0         0
Serial6/0/1        down  down      0%     0%       0         0
Serial6/0/2        down  down      0%     0%       0         0
Serial6/0/3        down  down      0%     0%       0         0
Serial6/0/4        down  down      0%     0%       0         0
Serial6/0/5        down  down      0%     0%       0         0
Serial6/0/6        down  down      0%     0%       0         0
Serial6/0/7        down  down      0%     0%       0         0
Tunnel0/0/1        up    up(s)    0%     0%       0         0
Virtual-Templatel  up    down      0%     0%       0         0
```

Run the **display interface tunnel 0/0/1** command on LCCE1 and LCCE2 to view the tunnel interface status. You can find that the status is Up (spoofing). The command output on LCCE1 is used as an example.

```
[LCCE1] display interface tunnel 0/0/1
Tunnel0/0/1 current state : UP
Line protocol current state : UP (spoofing)
Description:HUAWEI, AR Series, Tunnel0/0/1 Interface
Route Port,The Maximum Transmit Unit is 1500
Internet protocol processing : disabled
Encapsulation is TUNNEL, loopback not set
Tunnel protocol/transport SVPN/IP
Current system time: 2016-02-25 17:10:48
 300 seconds input rate 0 bits/sec, 0 packets/sec
 300 seconds output rate 0 bits/sec, 0 packets/sec
 99 seconds input rate 0 bits/sec, 0 packets/sec
 99 seconds output rate 0 bits/sec, 0 packets/sec
```

```
0 packets input, 0 bytes
0 input error
0 packets output, 0 bytes
0 output error
Input bandwidth utilization : 0%
Output bandwidth utilization : 0%
```

---End

Configuration Files

- LCCE1 configuration file

```
#
sysname LCCE1
#
l2tpv3 enable
#
interface GigabitEthernet0/0/1
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet0/0/2.1
dot1q termination vid 9
link-bridge Tunnel0/0/1 tagged
#
interface Tunnel0/0/1
tunnel-protocol svpn
encapsulation l2tpv3
l2tpv3 local session-id 1
l2tpv3 remote session-id 4
l2tpv3 local cookie length 4 plain lower-value 11
l2tpv3 remote cookie length 4 plain lower-value 22
tunnel-source 10.1.1.2
tunnel-destination 10.1.2.2
#
ip route-static 10.1.2.0 255.255.255.0 10.1.1.3
#
return
```

- LCCE2 configuration file

```
#
sysname LCCE2
#
l2tpv3 enable
#
interface GigabitEthernet0/0/1
ip address 10.1.2.2 255.255.255.0
#
interface GigabitEthernet0/0/2.1
dot1q termination vid 20
link-bridge Tunnel0/0/1 tagged
#
interface Tunnel0/0/1
tunnel-protocol svpn
encapsulation l2tpv3
l2tpv3 local session-id 4
l2tpv3 remote session-id 1
l2tpv3 local cookie length 4 plain lower-value 22
l2tpv3 remote cookie length 4 plain lower-value 11
tunnel-source 10.1.2.2
tunnel-destination 10.1.1.2
#
ip route-static 10.1.1.0 255.255.255.0 10.1.2.3
#
return
```

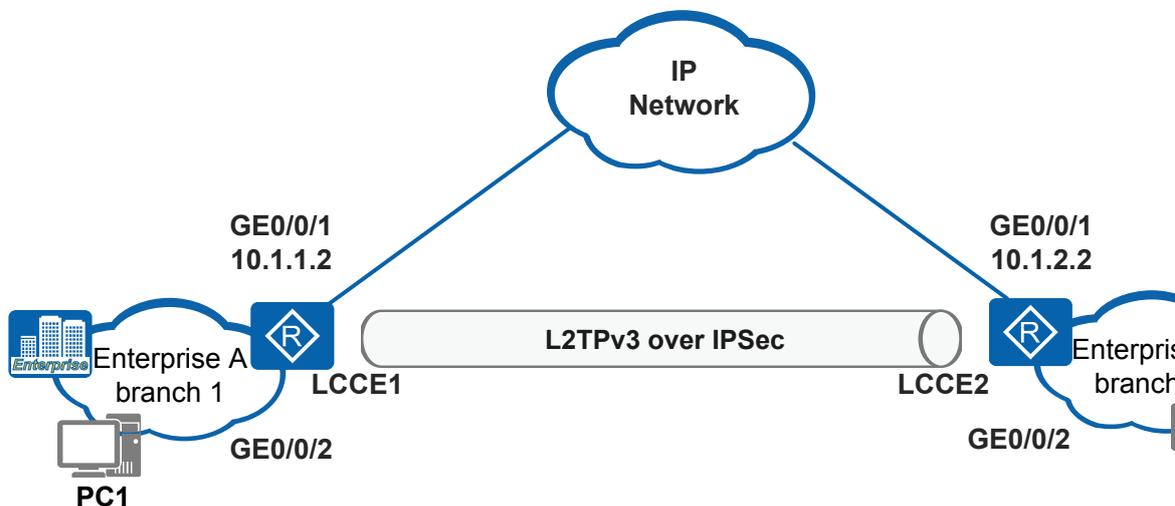
2.7.2 Example for Configuring L2TPv3 over IPsec to Implement Secure Communication Between Branches

Networking Requirements

In [Figure 2-6](#), enterprise A has two branches that connect to the IP network through LCCE1 and LCCE2 respectively. Branch 1 deploys a local area network (LAN) and uses LCCE1 as the gateway. Branch 2 deploys a LAN and uses LCCE2 as the gateway.

The enterprise wants to protect the services transmitted through the L2TPv3 tunnel against interception and tampering. To encrypt and protect the services, L2TPv3 over IPsec can be used.

Figure 2-6 Configuring L2TPv3 over IPsec for secure communication between branches



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a route to ensure communication between LCCE1 and LCCE2.
2. Enable the L2TPv3 function globally.
3. Establish a tunnel and configure the L2TPv3 tunnel parameters.
4. Configure the link bridge function to bind an AC interface to a tunnel interface.
5. Configure an ACL to define the data flows to be protected by IPsec.
6. Configure an IPsec proposal and define the traffic protection method.
7. Configure an IKE peer and define the attributes used for IKE negotiation.
8. Configure an IPsec policy, and apply the ACL, IPsec proposal, and IKE peer to the IPsec policy to define the data flows to be protected and protection method.
9. Apply the IPsec policy group to an interface so that the interface can protect traffic.

Procedure

- Step 1** Configure IP addresses and a static route for the PW interfaces on LCCE1 and LCCE2 respectively.

Configure an IP address for the PW interface on LCCE1.

```
<Huawei> system-view
[Huawei] sysname LCCE1
[LCCE1] interface gigabitethernet 0/0/1
[LCCE1-GigabitEthernet0/0/1] ip address 10.1.1.2 24
[LCCE1-GigabitEthernet0/0/1] quit
```

Configure a static route to LCCE2 on LCCE1. This example assumes that the next hop address in the route is 10.1.1.3.

```
[LCCE1] ip route-static 10.1.2.0 255.255.255.0 10.1.1.3
```

Configure an IP address for the PW interface on LCCE2.

```
<Huawei> system-view
[Huawei] sysname LCCE2
[LCCE2] interface gigabitethernet 0/0/1
[LCCE2-GigabitEthernet0/0/1] ip address 10.1.2.2 24
[LCCE2-GigabitEthernet0/0/1] quit
```

Configure a static route to LCCE1 on LCCE2. This example assumes that the next hop address in the route is 10.1.2.3.

```
[LCCE2] ip route-static 10.1.1.0 255.255.255.0 10.1.2.3
```

- Step 2** Enable the L2TPv3 function globally.

Enable the L2TPv3 function on LCCE1.

```
[LCCE1] l2tpv3 enable
```

Enable the L2TPv3 function on LCCE2.

```
[LCCE2] l2tpv3 enable
```

- Step 3** Configure L2TPv3 parameters for tunnel interfaces.

Create a tunnel on LCCE1 and configure parameters for the tunnel.

```
[LCCE1] interface tunnel 0/0/1
[LCCE1-Tunnel0/0/1] tunnel-protocol svpn
[LCCE1-Tunnel0/0/1] encapsulation l2tpv3 static
[LCCE1-Tunnel0/0/1] l2tpv3 local session-id 1
[LCCE1-Tunnel0/0/1] l2tpv3 remote session-id 4
[LCCE1-Tunnel0/0/1] l2tpv3 local cookie length 4 plain lower-value 11
[LCCE1-Tunnel0/0/1] l2tpv3 remote cookie length 4 plain lower-value 22
[LCCE1-Tunnel0/0/1] tunnel-source 10.1.1.2
[LCCE1-Tunnel0/0/1] tunnel-destination 10.1.2.2
[LCCE1-Tunnel0/0/1] quit
```

Create a tunnel on LCCE2 and configure parameters for the tunnel.

```
[LCCE2] interface tunnel 0/0/1
[LCCE2-Tunnel0/0/1] tunnel-protocol svpn
[LCCE2-Tunnel0/0/1] encapsulation l2tpv3 static
[LCCE2-Tunnel0/0/1] l2tpv3 local session-id 4
[LCCE2-Tunnel0/0/1] l2tpv3 remote session-id 1
[LCCE2-Tunnel0/0/1] l2tpv3 local cookie length 4 plain lower-value 22
[LCCE2-Tunnel0/0/1] l2tpv3 remote cookie length 4 plain lower-value 11
[LCCE2-Tunnel0/0/1] tunnel-source 10.1.2.2
[LCCE2-Tunnel0/0/1] tunnel-destination 10.1.1.2
[LCCE2-Tunnel0/0/1] quit
```

Step 4 Configure the link bridge function.

Configure the link bridge function on LCCE1 and bind an AC interface to a tunnel interface.

```
[LCCE1] interface GigabitEthernet 0/0/2  
[LCCE1-GigabitEthernet0/0/2] link-bridge tunnel10/0/1 tagged
```

Configure the link bridge function on LCCE2 and bind an AC interface to a tunnel interface.

```
[LCCE2] interface GigabitEthernet 0/0/2  
[LCCE2-GigabitEthernet0/0/2] link-bridge tunnel10/0/1 tagged
```

Step 5 Configure an ACL to define the data flows to be protected.

NOTE

The tunnel encapsulation protocol is IP (the protocol number is 115). UDP is not supported.

Configure ACL on LCCE1.

```
[LCCE1] acl number 3000  
[LCCE1-acl-adv-3000] rule permit 115 source 10.1.1.2 0 destination 10.1.2.2 0  
[LCCE1-acl-adv-3000] quit
```

Configure ACL on LCCE2.

```
[LCCE2] acl number 3000  
[LCCE2-acl-adv-3000] rule permit 115 source 10.1.2.2 0 destination 10.1.1.2 0  
[LCCE2-acl-adv-3000] quit
```

Step 6 Create an IPSec proposal.

Create an IPSec proposal on LCCE1.

```
[LCCE1] ipsec proposal rtb  
[LCCE1-ipsec-proposal-rtb] esp authentication-algorithm sha2-256  
[LCCE1-ipsec-proposal-rtb] esp encryption-algorithm aes-192  
[LCCE1-ipsec-proposal-rtb] quit
```

Create an IPSec proposal on LCCE2.

```
[LCCE2] ipsec proposal rta  
[LCCE2-ipsec-proposal-rta] esp authentication-algorithm sha2-256  
[LCCE2-ipsec-proposal-rta] esp encryption-algorithm aes-192  
[LCCE2-ipsec-proposal-rta] quit
```

Step 7 Configure an IKE peer.

Configure an IKE proposal on LCCE1.

```
[LCCE1] ike proposal 1  
[LCCE1-ike-proposal-1] encryption-algorithm aes-256  
[LCCE1-ike-proposal-1] authentication-algorithm sha2-256  
[LCCE1-ike-proposal-1] quit
```

Configure an IKE peer on LCCE1 and configure the pre-shared key and the remote ID of the IKE peer.

```
[LCCE1] ike peer rtb  
[LCCE1-ike-peer-rtb] ike-proposal 1  
[LCCE1-ike-peer-rtb] pre-shared-key cipher huawei@123  
[LCCE1-ike-peer-rtb] remote-address 10.1.2.2  
[LCCE1-ike-peer-rtb] quit
```

Configure an IKE proposal on LCCE2.

```
[LCCE2] ike proposal 1  
[LCCE2-ike-proposal-1] encryption-algorithm aes-256  
[LCCE2-ike-proposal-1] authentication-algorithm sha2-256  
[LCCE2-ike-proposal-1] quit
```

Configure an IKE peer on LCCE2 and configure the pre-shared key and the remote ID of the IKE peer.

```
[LCCE2] ike peer rta
[LCCE2-ike-peer-rta] ike-proposal 1
[LCCE2-ike-peer-rta] pre-shared-key cipher huawei@123
[LCCE2-ike-peer-rta] remote-address 10.1.1.2
[LCCE2-ike-peer-rta] quit
```

Step 8 Create an IPsec policy.

Configure an IPsec policy in IKE negotiation mode on LCCE1.

```
[LCCE1] ipsec policy rtb 1 isakmp
[LCCE1-ipsec-policy-isakmp-rtb-1] ike-peer rtb
[LCCE1-ipsec-policy-isakmp-rtb-1] proposal rtb
[LCCE1-ipsec-policy-isakmp-rtb-1] security acl 3000
[LCCE1-ipsec-policy-isakmp-rtb-1] quit
```

Configure an IPsec policy in IKE negotiation mode on LCCE2.

```
[LCCE2] ipsec policy rta 1 isakmp
[LCCE2-ipsec-policy-isakmp-rta-1] ike-peer rta
[LCCE2-ipsec-policy-isakmp-rta-1] proposal rta
[LCCE2-ipsec-policy-isakmp-rta-1] security acl 3000
[LCCE2-ipsec-policy-isakmp-rta-1] quit
```

Step 9 Apply the IPsec policy group to an interface so that the interface can protect traffic.

Apply the IPsec policy group to the PW interface of LCCE1.

```
[LCCE1] interface gigabitethernet0/0/1
[LCCE1-GigabitEthernet0/0/1] ipsec policy rtb
[LCCE1-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy group to the PW interface of LCCE2.

```
[LCCE2] interface gigabitethernet0/0/1
[LCCE2-GigabitEthernet0/0/1] ipsec policy rta
[LCCE2-GigabitEthernet0/0/1] quit
```

Step 10 Verify the configuration.

After the configurations are complete, PC1 can ping PC2 successfully. The data transmitted between PC1 and PC2 is encrypted.

Run the **display ipsec sa** command on LCCE1 and LCCE2 to view the IPsec configuration. The command output on LCCE1 is used as an example.

```
[LCCE1] display ipsec sa
```

```
ipsec sa
information:

=====

Interface:
GigabitEthernet0/0/1

=====
```

```
-----  
IPSec policy name:  
"rtb"  
  
Sequence number :  
1  
  
Acl group      :  
3000  
  
Acl rule      :  
5  
  
Mode          :  
ISAKMP  
  
-----  
  
Connection ID  :  
9  
  
Encapsulation mode:  
Tunnel  
  
Tunnel local   :  
10.1.1.2  
  
Tunnel remote  :  
10.1.2.2  
  
Flow source    : 10.1.1.2/255.255.255.255  
0/0  
Flow destination : 10.1.2.2/255.255.255.255  
0/0  
  
[Outbound ESP  
SAs]  
  
SPI: 1380002640  
(0x52412b50)  
  
Proposal: ESP-ENCRYPT-AES-192 ESP-AUTH-  
SHA2-256-128  
  
SA remaining key duration (kilobytes/sec):  
1532270/3514  
  
Outpacket count      :  
2686500  
  
Outpacket encap count :  
2686495  
  
Outpacket drop count :  
0  
  
Max sent sequence-number:  
2686293  
  
UDP encapsulation used for NAT traversal:  
N  
  
[Inbound ESP  
SAs]
```

```

    SPI: 2595661893
    (0x9ab6a845)

    Proposal: ESP-ENCRYPT-AES-192 ESP-AUTH-
    SHA2-256-128

    SA remaining key duration (kilobytes/sec):
    1490295/3514

    Inpacket count      :
    3068764

    Inpacket decap count :
    3068761

    Inpacket drop count :
    0

    Max received sequence-number:
    3068590

    UDP encapsulation used for NAT traversal:
    N

    Anti-replay :
    Enable

    Anti-replay window size: 1024
  
```

Run the **display interface brief** command on LCCE1 and LCCE2 to view the brief interface and IP information, including the IP addresses, subnet mask, physical and protocol status (Up or Down), and the number of interfaces in different status. The command output on LCCE1 is used as an example.

```

[LCCE1] display interface brief
PHY: Physical
*down: administratively down
(l): loopback
(s): spoofing
(b): BFD down
^down: standby
(e): ETHOAM down
InUti/OutUti: input utility/output utility
Interface          PHY   Protocol  InUti  OutUti  inErrors  outErrors
Atm8/0/0           down  down      0%    0%      0         0
Atm8/0/1           down  down      0%    0%      0         0
Atm8/0/2           down  down      0%    0%      0         0
Atm8/0/3           down  down      0%    0%      0         0
Cellular0/0/0     down  down      0%    0%      0         0
Cellular0/0/1     down  down      0%    0%      0         0
Ethernet1/0/0     up    up        0%    0%      0         0
Ethernet1/0/1     up    down     0.01% 0%      0         0
Ethernet2/0/0     down  down      0%    0%      0         0
GigabitEthernet0/0/0  up    up      0.01% 0.01%  0         0
GigabitEthernet0/0/1  up    up      0.01% 0%     0         0
GigabitEthernet0/0/2  up    up      0.01% 0%     0         0
GigabitEthernet0/0/3  up    down     0.01% 0%     0         0
GigabitEthernet3/0/0  down  down      0%    0%      0         0
MFR0/0/1          down  down      0%    0%      0         0
Mp-group0/0/1     down  down      0%    0%      0         0
NULL0             up    up(s)    0%    0%      0         0
Serial4/0/0       up    up      0.05% 0.05%  0         0
Serial6/0/0       down  down      0%    0%      0         0
Serial6/0/1       down  down      0%    0%      0         0
Serial6/0/2       down  down      0%    0%      0         0
Serial6/0/3       down  down      0%    0%      0         0
Serial6/0/4       down  down      0%    0%      0         0
Serial6/0/5       down  down      0%    0%      0         0
  
```

Serial6/0/6	down	down	0%	0%	0	0
Serial6/0/7	down	down	0%	0%	0	0
Tunnel0/0/1	up	up(s)	0%	0%	0	0
Virtual-Template1	up	down	0%	0%	0	0

Run the **display interface tunnel 0/0/1** command on LCCE1 and LCCE2 to view the tunnel interface status. You can find that the status is Up (spoofing). The command output on LCCE1 is used as an example.

```
[LCCE1] display interface tunnel 0/0/1
Tunnel0/0/1 current state : UP
Line protocol current state : UP (spoofing)
Description:HUAWEI, AR Series, Tunnel0/0/1 Interface
Route Port,The Maximum Transmit Unit is 1500
Internet protocol processing : disabled
Encapsulation is TUNNEL, loopback not set
Tunnel protocol/transport SVPN/IP
Current system time: 2016-02-25 17:10:48
 300 seconds input rate 0 bits/sec, 0 packets/sec
 300 seconds output rate 0 bits/sec, 0 packets/sec
 99 seconds input rate 0 bits/sec, 0 packets/sec
 99 seconds output rate 0 bits/sec, 0 packets/sec
 0 packets input, 0 bytes
 0 input error
 0 packets output, 0 bytes
 0 output error
Input bandwidth utilization : 0%
Output bandwidth utilization : 0%
```

----End

Configuration Files

- LCCE1 configuration file

```
#
sysname LCCE1
#
l2tpv3 enable
#
acl number 3000
 rule 5 permit 115 source 10.1.1.2 0 destination 10.1.2.2 0
#
ipsec proposal rtb
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-192
#
ike proposal 1
 encryption-algorithm aes-256
 authentication-algorithm sha2-256
#
ike peer rtb
 pre-shared-key cipher %^%#`KJ()J4dRTcJ2eLBf[3SEp3hQbWrGA;#K()Bw*h1%^%#
 ike-proposal 1
 remote-address 10.1.2.2
#
ipsec policy rtb 1 isakmp
 security acl 3000
 ike-peer rtb
 proposal rtb
#
interface GigabitEthernet0/0/1
 ip address 10.1.1.2 255.255.255.0
 ipsec policy rtb
#
interface GigabitEthernet0/0/2
 link-bridge Tunnel0/0/1 tagged
#
interface Tunnel0/0/1
```

```
tunnel-protocol svpn
encapsulation l2tpv3
l2tpv3 local session-id 1
l2tpv3 remote session-id 4
l2tpv3 local cookie length 4 plain lower-value 11
l2tpv3 remote cookie length 4 plain lower-value 22
tunnel-source 10.1.1.2
tunnel-destination 10.1.2.2
#
ip route-static 10.1.2.0 255.255.255.0 10.1.1.3
#
return
```

- LCCE2 configuration file

```
#
sysname LCCE2
#
l2tpv3 enable
#
acl number 3000
rule 5 permit 115 source 10.1.2.2 0 destination 10.1.1.2 0
#
ipsec proposal rta
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-192
#
ike proposal 1
encryption-algorithm aes-256
authentication-algorithm sha2-256
#
ike peer rta
pre-shared-key cipher %^%#`KJ{)J4dRTcJ2eLBf[3SEp3hQbWrGA;#K()Bw*h1%^%#
ike-proposal 1
remote-address 10.1.1.2
#
ipsec policy rta 1 isakmp
security acl 3000
ike-peer rta
proposal rta
#
interface GigabitEthernet0/0/1
ip address 10.1.2.2 255.255.255.0
ipsec policy rta
#
interface GigabitEthernet0/0/2
link-bridge Tunnel0/0/1 tagged
#
interface Tunnel0/0/1
tunnel-protocol svpn
encapsulation l2tpv3
l2tpv3 local session-id 4
l2tpv3 remote session-id 1
l2tpv3 local cookie length 4 plain lower-value 22
l2tpv3 remote cookie length 4 plain lower-value 11
tunnel-source 10.1.2.2
tunnel-destination 10.1.1.2
#
ip route-static 10.1.1.0 255.255.255.0 10.1.2.3
#
return
```

2.8 References for L2TPv3

The following table lists the references for L2TPv3.

Document	Description
RFC3931	Layer Two Tunneling Protocol - Version 3 (L2TPv3)
RFC4719	Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)

3 GRE Configuration

About This Chapter

Generic Routing Encapsulation (GRE) encapsulates packets of certain network protocols, such as IP and Asynchronous Transfer Mode (ATM), so that the encapsulated packets can be transmitted over the IPv4 network.

[3.1 Overview of GRE](#)

[3.2 Understanding GRE](#)

This section outlines the basic principles used when implementing GRE technology.

[3.3 Application Scenarios for GRE](#)

This section describes the application scenarios for GRE.

[3.4 Licensing Requirements and Limitations for GRE](#)

This section describes GRE configuration notes.

[3.5 Default Settings for GRE](#)

This section provides the default settings for GRE.

[3.6 Configuring a GRE Tunnel](#)

This section describes how to configure a GRE tunnel on an IPv4 network.

[3.7 Maintaining the GRE Tunnel](#)

This section describes how to collect and view statistics on tunnel interfaces, monitor the GRE running status, and reset the Keepalive packet statistics on tunnel interfaces.

[3.8 Configuration Examples for GRE](#)

This section provides GRE configuration examples, including networking requirements, configuration roadmap, and configuration procedure.

[3.9 Troubleshooting GRE](#)

This section describes common faults caused by incorrect configurations and provides the troubleshooting procedure.

[3.10 FAQ About GRE](#)

This section describes the FAQ about GRE.

[3.11 References for GRE](#)

This section lists references for GRE.

3.1 Overview of GRE

Definition

Generic Routing Encapsulation (GRE) is a protocol that encapsulates packets of some network layer protocols, such as Internetwork Packet Exchange (IPX), Asynchronous Transfer Mode (ATM), IPv6, and AppleTalk. Then the encapsulated packets can be transmitted over a different network layer protocol, such as IPv4.

As Layer 3 tunneling technology, GRE encapsulates packets of a protocol into packets of another protocol to transparently transmit packets over GRE tunnels. This technology enables packet transmission on heterogeneous networks.

Benefits

- GRE is easy to implement and increases only a few loads on devices at both ends of a tunnel.
- GRE sets up tunnels over an IPv4 network to connect networks running different protocols, using the original network structure and reducing costs.
- GRE enlarges the operation scope of network protocols that support limited hop counts, allowing for flexible topologies on enterprise networks.
- GRE can encapsulate multicast data and work with IPSec to ensure security of multicast services such as voice and video services.
- GRE can work with Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP). MPLS LDP packets are transmitted over GRE tunnels to set up LDP label switched paths (LSPs), so that MPLS backbone networks can be connected through a heterogeneous network.
- GRE connects discontinuous subnets and sets up virtual private networks (VPNs) to ensure secure connections between the enterprise headquarters and branches.

3.2 Understanding GRE

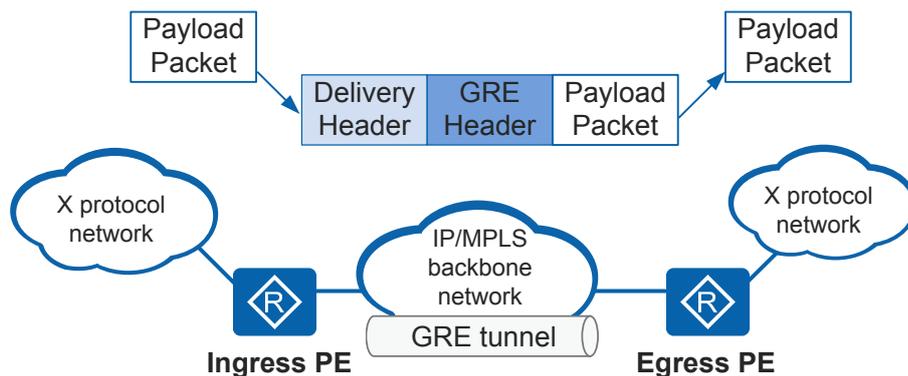
This section outlines the basic principles used when implementing GRE technology.

3.2.1 Basic Concepts

Implementation

Packets transmitted over a GRE tunnel undergo encapsulation and decapsulation processes. As shown in [Figure 3-1](#), if the ingress PE transmits X protocol packets to the egress PE, the ingress PE encapsulates the packets and the egress PE decapsulates the packets. A GRE tunnel is a path along which encapsulated packets are transmitted.

Figure 3-1 Transmitting X protocol packets over a GRE tunnel



- Encapsulation
 - a. After receiving an X protocol packet from the interface connected to the X network, the ingress PE sends the packet to the X protocol.
 - b. The X protocol checks the destination address in the packet header and searches the routing table or the forwarding table for the outbound interface. If the outbound interface is a GRE tunnel interface, the ingress PE adds a GRE header to the packet.
 - c. The ingress PE adds an IP header to the packet because the backbone network runs the IP protocol. The source address in the IP header is the tunnel source address and the destination address in the IP header is the tunnel destination address.
 - d. The ingress PE searches the IP routing table for the outbound interface based on the destination address in the IP header (tunnel destination address) and transmits the packet over the IP backbone network.

For details on the format of the encapsulated packet, see [Packet Format](#).

- Decapsulation

The decapsulation process is opposite to the encapsulation process.

 - a. After receiving the packet from the GRE tunnel interface, the egress PE analyzes the IP header in the packet and finds that itself is the destination of the packet. Then the egress PE removes the IP header and delivers the packet to the GRE protocol for processing.
 - b. The GRE protocol removes the GRE header and delivers the packet to the X protocol.

Packet Format

Figure 3-2 shows the format of a GRE-encapsulated packet.

- Passenger protocol: indicates the protocol of the original packet. The original packet is encapsulated in a GRE packet as the payload.
- Encapsulation protocol: indicates the protocol used to encapsulate passenger protocol packets by adding a GRE header. It is also called the carrier protocol.
- Transport protocol (or delivery protocol): indicates the protocol used to transmit the encapsulated packets.

Figure 3-2 GRE packet format

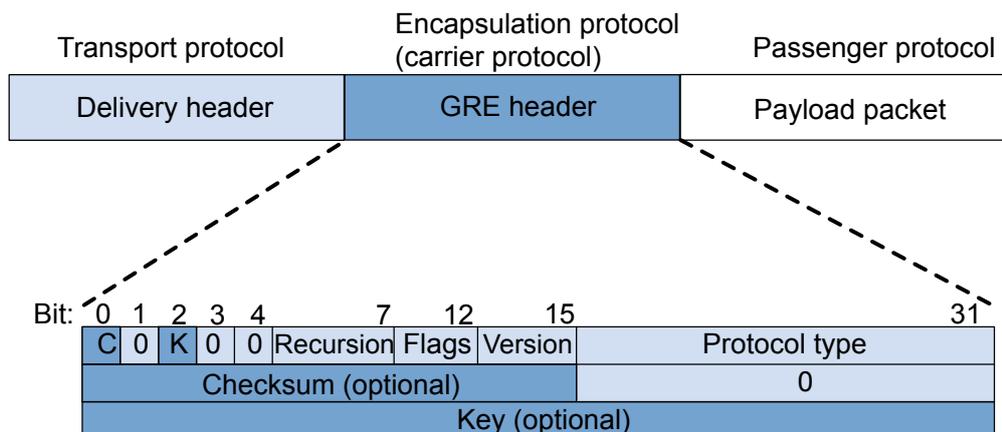


Table 3-1 describes the fields in a GRE header.

Table 3-1 Description of fields in a GRE header

Field	Description
C	Checksum bit. <ul style="list-style-type: none"> ● 1: The GRE header contains the Checksum field. ● 0: The GRE header does not contain the Checksum field.
K	Key bit. <ul style="list-style-type: none"> ● 1: The GRE header contains the Key field. ● 0: The GRE header does not contain the Key field.

Field	Description
Recursion	<p>Number of times a packet is encapsulated by GRE. The value of this field increases by 1 every time the packet is encapsulated. If the packet is encapsulated more than three times, the device discards the packet. This field prevents a packet from being encapsulated infinitely.</p> <p>NOTE</p> <ul style="list-style-type: none"> ● According to RFC1701, the default value of this field is 0. ● According to RFC2784, no error will occur if the field value on the transmit end is different from that on the receive end, and the receive end must ignore the field. ● This field is only used to indicate the number of times a packet is encapsulated. When GRE decapsulates a packet, this field does not affect packet processing.
Flags	Reserved field. The value must be set to 0.
Version	Version number. The value must be set to 0.
Protocol Type	<p>Type of the passenger protocol. A common passenger protocol is the IPv4 protocol, with the value of 0800.</p> <p>The protocol code of Ethernet over GRE is 0x6558.</p>
Checksum	Checksum of the GRE header and the payload.
Key	Key used to authenticate the packet at the receive end.

 **NOTE**

Currently, the GRE header does not contain the Source Route field; therefore, bit 1, bit 3, and bit 4 are all set to 0.

3.2.2 GRE Security Mechanisms

GRE provides two types of security mechanisms:

- [Checksum Verification](#)
- [Key Authentication](#)

Checksum Verification

Checksum verification is an end-to-end check on encapsulated packets.

If the C bit in the GRE header is set to 1, the checksum is valid. The sender calculates the checksum based on information in the GRE header and the payload and then sends the packet

containing the checksum to the receiver. The receiver calculates the checksum based on information in the received packet and compares the calculated value with the checksum in the packet. If they are the same, the receiver forwards the packet. If they are different, the receiver discards the packet.

You can enable or disable checksum verification on both ends of a tunnel in actual applications. If checksum verification is enabled on the local end and disabled on the remote end, the local end does not check the checksum values of received packets, but checks the checksum values of packets to be sent. If checksum verification is disabled on the local end and enabled on the remote end, the local end checks the checksum values of received packets, but does not check the checksum values of packets to be sent.

Key Authentication

Key authentication is used to verify the validity of a tunnel interface. This security mechanism ensures that a device accepts only packets sent from a valid tunnel interface and discards invalid packets.

According to RFC 1701, if the K bit in the GRE header is set to 1, a four-byte Key field is inserted into the GRE header. Both the receiver and the sender need to authenticate the key.

The Key field is used to identify the traffic in a tunnel. Packets transmitted over the same tunnel use the same key. During decapsulation, GRE identifies data packets based on the key. Packets pass key authentication only when the keys on both ends of the tunnel are consistent. Otherwise, packets failing the key authentication are discarded. When both ends of a tunnel have no key or the same key, the key configurations on the two ends are consistent.

3.2.3 Keepalive Detection

GRE does not provide the link status detection function. If the remote interface is unreachable, the tunnel cannot be immediately torn down. As a result, the source continuously forwards packets to the remote end which cannot receive the packets, generating a data black hole.

The Keepalive detection function monitors the tunnel status to check whether the remote end is reachable. If the remote end is unreachable, the source end tears down the tunnel immediately. This prevents data loss and data black holes and ensures reliable data transmission.

The Keepalive detection function is implemented as follows:

1. After the Keepalive detection function is enabled on the source end of a GRE tunnel, the source end starts a timer to periodically send and count Keepalive probes. The number increases by 1 every time a Keepalive probe is sent.
2. The remote end sends a reply packet to the source end after receiving a probe.
3. If the source end receives a reply packet before the counter value reaches the preset value, the source end considers the remote end reachable. If the source end does not receive any reply packet when the counter reaches the preset value (retry times), the source end considers the remote end unreachable and tears down the tunnel. In this case, the source interface still sends Keepalive probes to the remote interface. When the remote interface becomes Up, the source interface becomes Up too and sets up a tunnel with the remote interface.

 **NOTE**

The Keepalive detection function takes effect on one end of a tunnel as long as it is configured at that end, regardless of whether it is configured on the other end. If the remote end receives a Keepalive probe, it sends a replay packet to the source end, regardless of whether it is configured with the Keepalive detection function.

3.2.4 Ethernet over GRE

Generic Routing Encapsulation (GRE) provides a mechanism to encapsulate packets of a protocol into packets of another protocol. This allows packets to be transmitted over heterogeneous networks. A channel for transmitting heterogeneous packets is called a tunnel.

A GRE tunnel can be established using the following tunnel interfaces:

- GRE tunnel interface

A GRE tunnel interface is a point-to-point virtual interface used to encapsulate packets, and has the source address, destination address, and tunnel interface IP address.

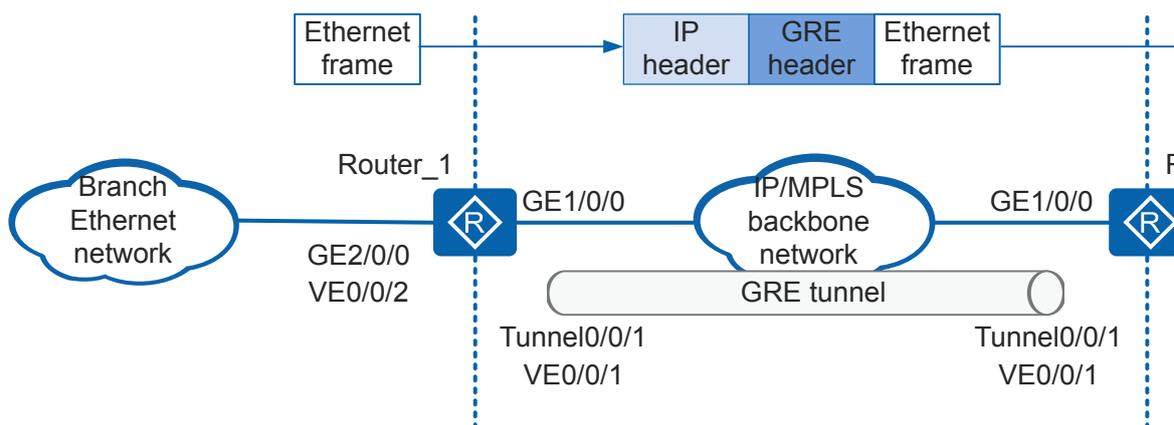
- mGRE tunnel interface

An mGRE tunnel interface is a point-to-multipoint virtual interface used in Dynamic Smart VPN (DSVPN) applications, and has the source address, destination address, and tunnel interface IP address.

The destination IP address of a GRE tunnel interface is manually configured, whereas the destination IP address of an mGRE tunnel is resolved by the next hop resolution protocol (NHRP). An mGRE tunnel interface has multiple remote ends because there are multiple GRE tunnels on the interface.

In [Figure 3-3](#), the enterprise headquarters and branch use Ethernet networks and are connected by an IP backbone network. Ethernet over GRE can be deployed to transparently transmit Ethernet packets over a GRE tunnel, enabling communication between the enterprise headquarters and branch.

Figure 3-3 Ethernet over GRE networking



Ethernet over GRE encapsulates Ethernet packets using GRE and transmits the encapsulated packets over a network running another network layer protocol, such as IPv4. The detailed working process is as follows:

1. Layer 2 virtual Ethernet (VE) interfaces VE0/0/2 and VE0/0/1 are bound to the LAN-side physical Ethernet interface GE2/0/0 and WAN-side tunnel interface Tunnel0/0/1 of the routers, respectively.
2. LAN-side GE2/0/0 on Router_1 receives an Ethernet packet containing a VLAN tag from the branch network.
3. GE2/0/0 forwards the packet to VE0/0/2. VE0/0/2 processes the VLAN tag, forwards the packet at Layer 2 based on the MAC address and VLAN tag, and finds the outbound interface VE0/0/1.
4. VE0/0/1 processes the VLAN tag in the Ethernet packet and forwards it to the bound Tunnel0/0/1. Tunnel0/0/1 encapsulates the Ethernet packet using GRE (with the protocol code 0x6558) and forwards the encapsulated packet over a GRE tunnel.
5. Tunnel0/0/1 on Router_2 decapsulates the received packet using GRE, finds that the protocol code is 0x6558, and forwards the decapsulated Ethernet packet to the inbound interface VE0/0/1.
6. VE0/0/1 processes the VLAN tag in the packet and forwards it to the outbound interface VE0/0/2. VE0/0/2 processes the VLAN tag in the packet and sends it to the outbound interface GE2/0/0.
7. GE2/0/0 sends the Ethernet packet containing a new VLAN tag to the headquarters network.

3.2.5 Ethernet over mGRE

Generic Routing Encapsulation (GRE) provides a mechanism to encapsulate packets of a protocol into packets of another protocol. This allows packets to be transmitted over heterogeneous networks. A channel for transmitting heterogeneous packets is called a tunnel.

A GRE tunnel can be established using the following tunnel interfaces:

- GRE tunnel interface

A GRE tunnel interface is a point-to-point virtual interface used to encapsulate packets, and has the source address, destination address, and tunnel interface IP address.

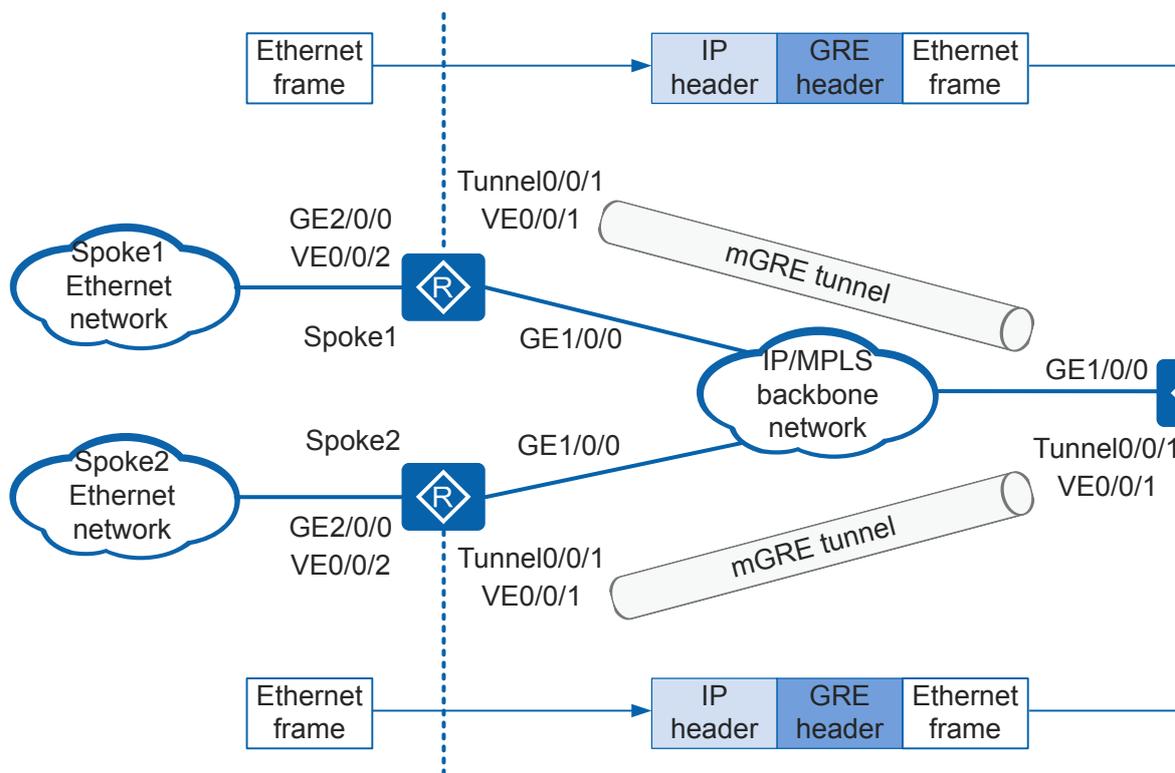
- mGRE tunnel interface

An mGRE tunnel interface is a point-to-multipoint virtual interface used in Dynamic Smart VPN (DSVPN) applications, and has the source address, destination address, and tunnel interface IP address.

The destination IP address of a GRE tunnel interface is manually configured, whereas the destination IP address of an mGRE tunnel is resolved by the next hop resolution protocol (NHRP). An mGRE tunnel interface has multiple remote ends because there are multiple GRE tunnels on the interface.

In [Figure 3-4](#), the enterprise headquarters (Hub) and branches (Spokes) use Ethernet networks and are connected by an IP backbone network. Ethernet over mGRE can be deployed to transparently transmit Ethernet packets over mGRE tunnels, enabling communication between the Hub and Spokes.

Figure 3-4 Ethernet over mGRE networking



Ethernet over mGRE encapsulates Ethernet packets using GRE and transmits the encapsulated packets over a network running another network layer protocol, such as IPv4. The detailed working process is as follows:

1. Layer 2 virtual Ethernet (VE) interfaces VE0/0/2 and VE0/0/1 are bound to the LAN-side physical Ethernet interface GE2/0/0 and WAN-side tunnel interface Tunnel0/0/1 of the routers, respectively.
2. LAN-side GE2/0/0 on Spoke1 receives an Ethernet packet containing a VLAN tag from branch network 1.
3. GE2/0/0 forwards the packet to VE0/0/2. VE0/0/2 processes the VLAN tag, forwards the packet at Layer 2 based on the MAC address and VLAN tag, and finds the outbound interface VE0/0/1.
4. VE0/0/1 processes the VLAN tag in the Ethernet packet and forwards it to the bound Tunnel0/0/1. Tunnel0/0/1 encapsulates the Ethernet packet using GRE (with the protocol code 0x6558) and forwards the encapsulated packet over an mGRE tunnel.
5. Tunnel0/0/1 on Hub decapsulates the received packet using GRE, finds that the protocol code is 0x6558, and forwards the decapsulated Ethernet packet to the inbound interface VE0/0/1.
6. VE0/0/1 processes the VLAN tag in the packet and forwards it to the outbound interface VE0/0/2. VE0/0/2 processes the VLAN tag in the packet and sends it to the outbound interface GE2/0/0.

7. GE2/0/0 sends the Ethernet packet containing a new VLAN tag to the headquarters network.
8. The forwarding process of packets from Spoke2 to Hub is the same as that from Spoke1 to Hub.

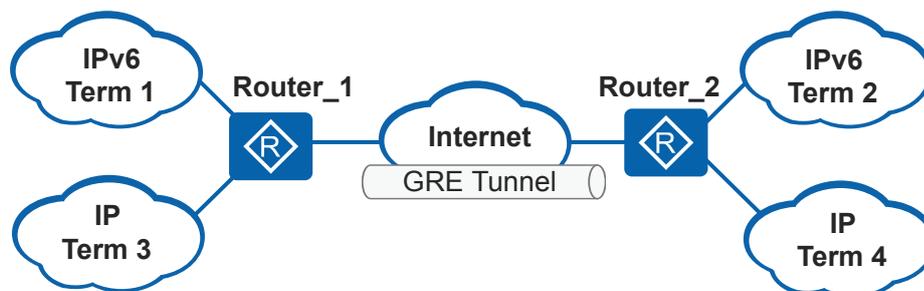
3.3 Application Scenarios for GRE

This section describes the application scenarios for GRE.

3.3.1 Transmitting Data of Multi-Protocol Local Networks Through a GRE Tunnel

As shown in [Figure 3-5](#), Term1 and Term2 are the local networks running IPv6. Term3 and Term4 are local networks running the IP protocol. These subnets, located in different areas, need to communicate through the public IP network.

Figure 3-5 Transmitting data of multi-protocol local networks through a GRE tunnel



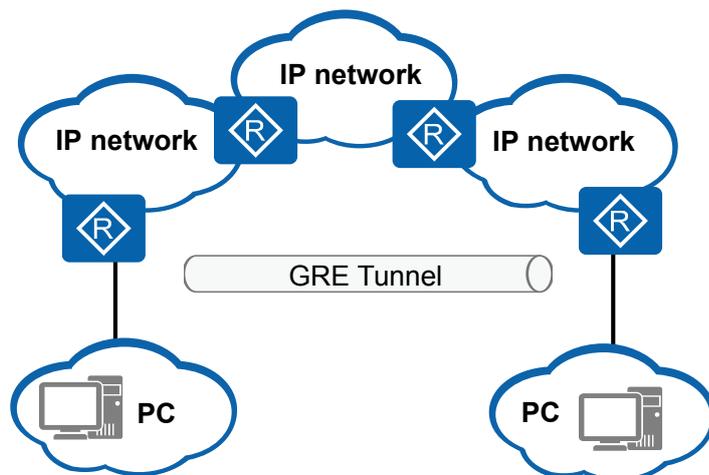
Router_1 and Router_2 set up a GRE tunnel through which Term1 can communicate with Term2 without affecting communication between Term3 and Term4.

3.3.2 Enlarging the Operation Scope of a Network with a Hop Limit

As shown in [Figure 3-6](#), the network runs the IP protocol. Assume that the IP protocol limits the hop count to 255. If the hop count between two PCs is more than 255, they cannot communicate with each other. You can set up a GRE tunnel between two devices on the network to hide the hops between them. This enlarges the network operation scope.

For example, the Routing Information Protocol (RIP) protocol defines that a route is unreachable when the hop count reaches 16. You can configure a GRE tunnel between two devices to reduce the hop count of the RIP route passing through the GRE tunnel to less than 16. Then the route is reachable.

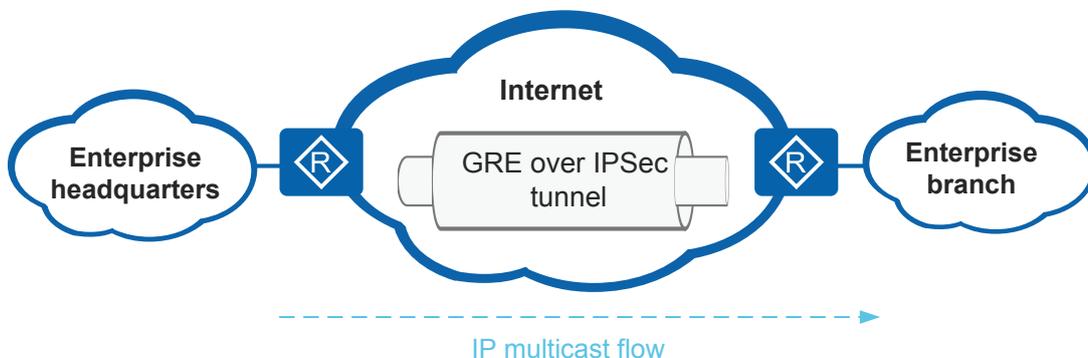
Figure 3-6 Enlarging the network operation scope



3.3.3 Combining GRE with IPsec to Protect Multicast Data

GRE can encapsulate multicast data and the data in a GRE tunnel. As show in [Figure 3-7](#), for multicast data that needs to be transmitted over an IPsec tunnel, you can set up a GRE tunnel, encapsulate multicast data with GRE, encrypt the encapsulated data with IPsec, and then transmit the data over the IPsec tunnel.

Figure 3-7 Application of the GRE over IPsec tunnel



3.3.4 Setting Up an L2VPN and an L3VPN Using a GRE Tunnel

NOTE

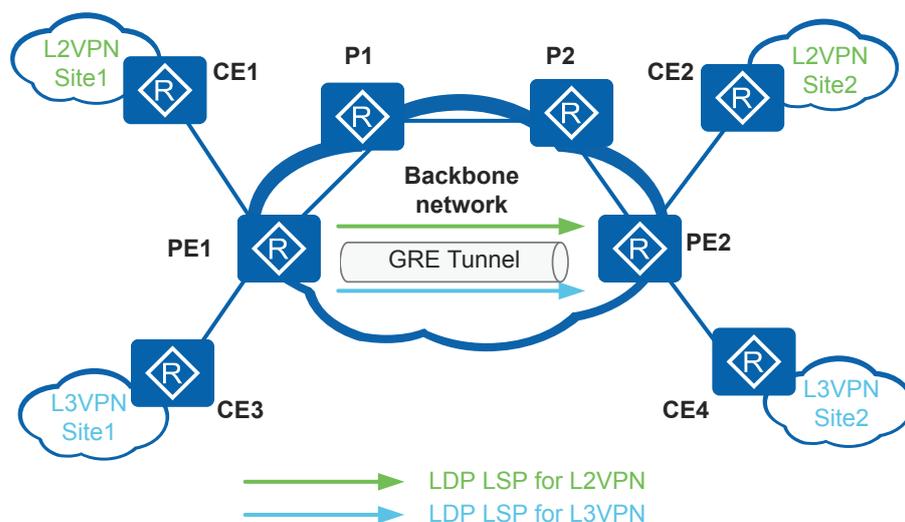
The AR120-S&AR150-S&AR160-S&AR200-S series routers cannot be used in this scenario.

Usually, a Multiprotocol Label Switching (MPLS) VPN backbone network uses label switched paths (LSPs) as public network tunnels. If the core devices (P devices) on the backbone network do not support MPLS but the edge devices (PE devices) support MPLS, LSPs cannot be used as public network tunnels. In this scenario, a GRE tunnel can be used instead to provide the Layer 3 virtual private network (L3VPN) or Layer 2 virtual private network (L2VPN) solution on the backbone network.

Label Distribution Protocol (LDP) over GRE enables MPLS LDP packets to be transmitted over a GRE tunnel. After MPLS LDP is enabled on GRE tunnel interfaces, LDP LSPs can be set up over the GRE tunnel.

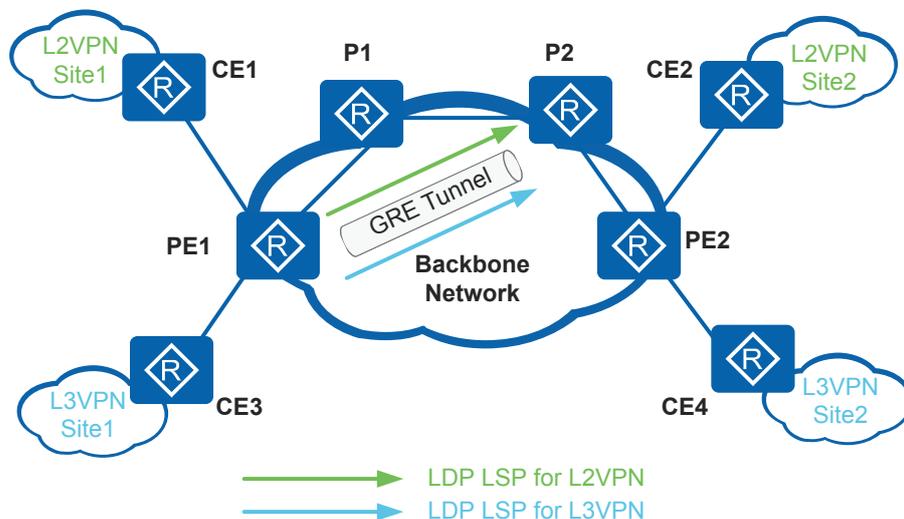
As shown in **Figure 3-8**, L2VPN or L3VPN services are deployed on PE1 and PE2 to allow communication between the L2VPN or L3VPN sites of the enterprise. When backbone network devices have MPLS disabled or do not support MPLS, an LDP LSP over a GRE tunnel needs to be set up between PE1 and PE2.

Figure 3-8 LDP over GRE application in an enterprise L3VPN or L2VPN solution (MPLS not supported on the P devices)



As shown in **Figure 3-9**, MPLS is enabled only on P2 but not P1 on the backbone network. A GRE tunnel can be set up between PE1 and P2, in order to set up an LDP LSP over the GRE tunnel.

Figure 3-9 LDP over GRE application in an enterprise L3VPN or L2VPN solution (MPLS supported only on P2)



3.3.5 Connecting CE Devices to an MPLS VPN Network

NOTE

The AR120-S&AR150-S&AR160-S&AR200-S cannot work on an MPLS backbone network.

The MPLS VPN solution provides better services than the traditional IP VPN solution. Therefore, MPLS VPN technology is now carrier's preferred VPN technology. However, the Internet is IP based and a large number of backbone networks still use IP technology.

In the MPLS VPN solution, a customer edge (CE) device must have a direct physical link to a provider edge (PE) device on the MPLS backbone network to connect to the VPN. That is, the CE and PE devices must be on the same network. In this networking, you must associate the VPN instance with the PE device's physical interface connected to the CE device.

In actual networking, the CE and PE devices may not be directly connected by physical links. For example, the CE devices of multiple organizations that are connected to the Internet or an IP-based backbone network may be far away from the PE devices on the MPLS backbone network; therefore, they cannot be connected directly. These organizations cannot directly connect to the internal sites of the MPLS VPN through the Internet or the IP backbone network.

Figure 3-10 Connecting CE devices to an MPLS VPN backbone network through an IP backbone network



To connect a CE device to an MPLS VPN backbone network, create a logical direct connection between the CE and PE devices. You can connect the CE and PE devices using a public or private network, and create a GRE tunnel between the CE and PE devices. Then, the CE and PE devices can communicate as if they were directly connected, and the GRE tunnel can be associated with the VPN as a physical interface.

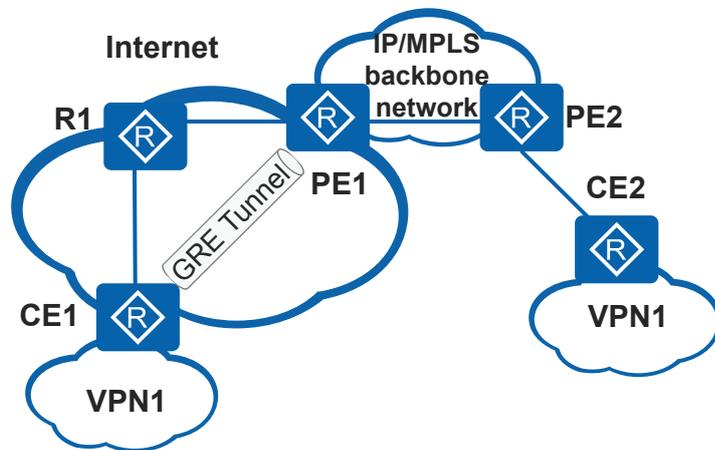
A GRE tunnel can be set up in the following ways to connect CE devices to an MPLS VPN network:

- GRE tunnel over a private network: The GRE tunnel is associated with a VPN instance, and the source interface (or the source address) and the destination address of the GRE tunnel belong to this VPN instance.
- GRE tunnel over a public network: The GRE tunnel is associated with a VPN instance. However, the source address and destination address of the GRE tunnel are public IP addresses and do not belong to the VPN instance.
- GRE over a VPN: The GRE tunnel is associated with a VPN instance (such as VPN1), while the source interface of the GRE tunnel is bound to another VPN instance (such as VPN2). The GRE tunnel traverses VPN2.

GRE Tunnel over a Public Network

In this networking, the CE and PE devices must have one interface using a public IP address. The CE and PE devices must have a route to each other in their public network routing tables.

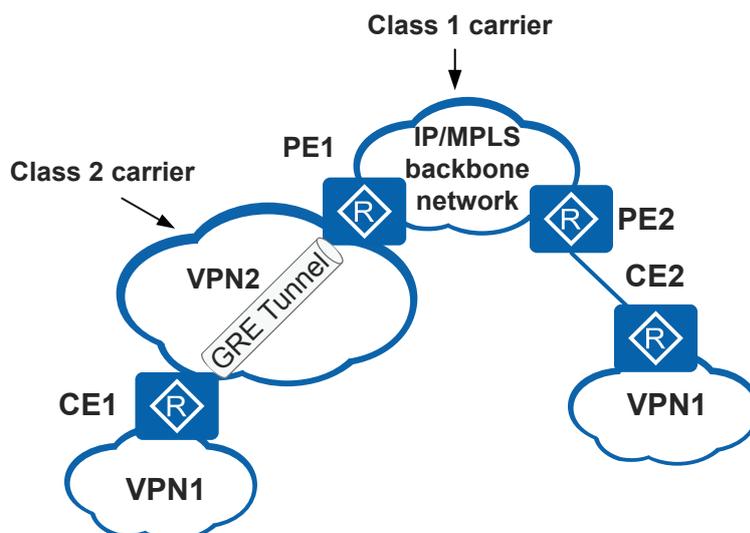
Figure 3-11 GRE tunnel over a public network



GRE Tunnel over a VPN

This networking differs from a GRE tunnel over a public network in that the CE device is connected to the PE device across a VPN (VPN2 in this example), but not a public network. Both the outbound interface of the private data from the CE and the outbound interface of the private data from the PE belong to VPN2.

Figure 3-12 GRE tunnel over a VPN



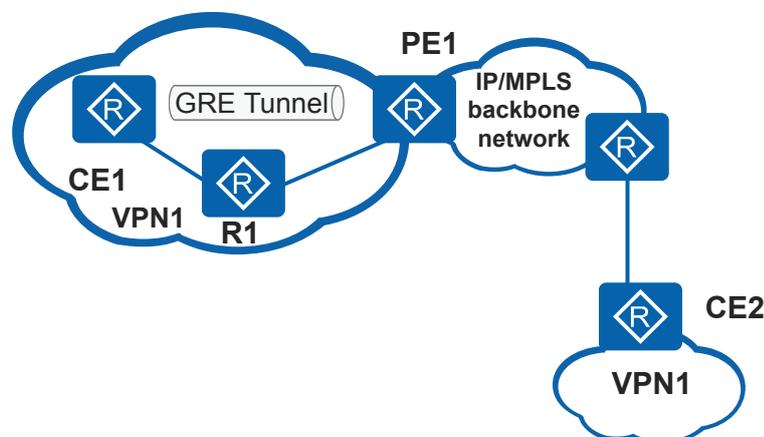
In Figure 3-12, PE1 and PE2 are the edge devices of the first carrier on the MPLS backbone network. VPN2 is a VPN of a second carrier network. CE1 and CE2 are customer devices.

To deploy a VPN (VPN1 in this example) based on the MPLS network, you can set up a GRE tunnel between PE1 and CE1 across VPN2. Then CE1 and PE1 are directly connected through the GRE tunnel.

GRE Tunnel over a Private Network

In this networking, the source address and the destination address of the GRE tunnel belong to the private network. In actual applications, creating a tunnel on a private network serves no purpose; therefore, this networking is not recommended. As shown in Figure 3-13, R1 can be used as a CE device so no GRE tunnel is required.

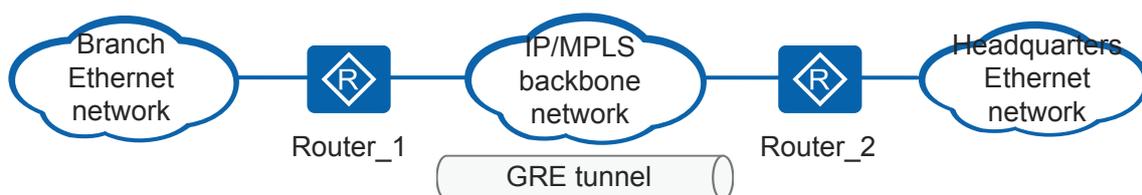
Figure 3-13 GRE tunnel over a private network



3.3.6 Ethernet over GRE Application

In **Figure 3-14**, LAN-side interfaces of Router_1 and Router_2 connect to Ethernet networks, and their WAN-side interfaces connect to an IP backbone network. Ethernet packets need to be transparently transmitted over a GRE tunnel.

Figure 3-14 Ethernet over GRE application

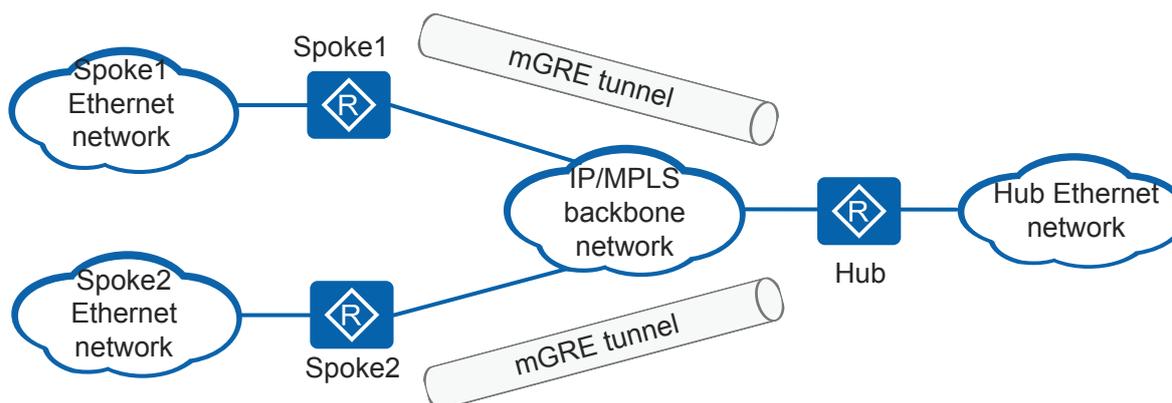


Ethernet over GRE can be configured on Router_1 and Router_2 and Layer 2 VE interfaces are bound to the physical Ethernet interface and tunnel interface to transparently transmit Ethernet packets over a GRE tunnel.

3.3.7 Ethernet over mGRE Application

In **Figure 3-15**, the enterprise headquarters (Hub) and branches (Spokes) use Ethernet networks and are connected by an IP backbone network. Ethernet packets between the Hub and Spokes need to be transparently transmitted over mGRE tunnels.

Figure 3-15 Ethernet over mGRE application



Ethernet over mGRE can be configured on Spoke1, Spoke2, and Hub and Layer 2 VE interfaces are bound to the physical Ethernet interfaces and tunnel interfaces to transparently transmit Ethernet packets over mGRE tunnels.

3.4 Licensing Requirements and Limitations for GRE

This section describes GRE configuration notes.

Involved Network Elements

None

License Requirements

GRE is a basic feature of the device and is not under license control.

Feature Limitations

None

3.5 Default Settings for GRE

This section provides the default settings for GRE.

Table 3-2 Default settings for GRE

Parameter	Default Setting
Keepalive detection	Disabled
GRE checksum	Disabled
GRE key	Disabled

3.6 Configuring a GRE Tunnel

This section describes how to configure a GRE tunnel on an IPv4 network.

Pre-configuration Tasks

Before configuring a GRE tunnel, complete the following task:

- Configuring a reachable route between the source and destination interfaces according to IP Unicast Routing Configuration Guide

Configuration Procedure

Perform the following operations in sequence to configure a GRE tunnel. You can determine whether to perform optional operations based on site requirements.

Protocol	Task
Network layer protocol, such as IPX, ATM, IPv6, and AppleTalk	3.6.1 Configuring a Tunnel Interface 3.6.2 Configuring a Route on a Tunnel Interface 3.6.4 (Optional) Configuring a Security Mechanism for GRE 3.6.5 (Optional) Enabling the Keepalive Detection Function for GRE
FR, HDLC, or PPP protocol	3.6.1 Configuring a Tunnel Interface 3.6.3 (Optional) Configuring the Link Bridge Function 3.6.4 (Optional) Configuring a Security Mechanism for GRE 3.6.5 (Optional) Enabling the Keepalive Detection Function for GRE
Ethernet protocol	3.6.1 Configuring a Tunnel Interface 3.6.2 Configuring a Route on a Tunnel Interface 3.6.3 (Optional) Configuring the Link Bridge Function or 3.6.6 (Optional) Configuring Ethernet over GRE 3.6.4 (Optional) Configuring a Security Mechanism for GRE 3.6.5 (Optional) Enabling the Keepalive Detection Function for GRE

3.6.1 Configuring a Tunnel Interface

Context

A GRE tunnel is established between two tunnel interfaces; therefore, you need to configure tunnel interfaces on devices at both ends of a tunnel. Set the protocol type to GRE, specify the tunnel source address (or interface) and tunnel destination address, and specify IP addresses for tunnel interfaces.

- Source IP address of the tunnel: Indicates the source IP address defined in the packet transmission protocol. When the configured value is an IP address, the value is directly used as the source IP address. When the configured value is a source interface, the IP address of the interface is used as the source IP address.
- Destination IP address of the tunnel: Indicates the destination IP address defined in the packet transmission protocol.
- IP address of the tunnel interface: Indicates an IP address assigned to the tunnel interface. A dynamic or static routing protocol uses this IP address to advertise the tunnel interface. The IP address of the tunnel interface may be a public network address or not. It can also be an IP address borrowed from another interface to save IP addresses. However, if the IP address of the tunnel interface is borrowed from another interface, tunnel interface communication cannot be implemented using this IP address. Therefore, to achieve tunnel reachability if the IP address is borrowed, you must configure a static route or a routing protocol to implement reachability of the IP address.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

Step 3 Run **tunnel-protocol gre**

The protocol type of the tunnel interface is set to GRE.

Step 4 (Optional) Run **gre key** { **plain** *key-number* | [**cipher**] *plain-cipher-text* }

The key number of a GRE tunnel.

NOTE

Normally, only one GRE tunnel can be established over a physical link that has only one source address and one destination address. To solve this problem, GRE keys are introduced to identify tunnel interfaces with the same source address and same destination address. This implementation allows multiple GRE tunnels to be established over a physical link that has only one source address and one destination address to carry different types of services.

If you want to configure the same source and destination addresses for multiple GRE tunnels, configure the **gre key** command first. Otherwise, the configuration fails.

Step 5 Run **source** { *source-ip-address* | *interface-type interface-number* }

A source address or source interface is specified for the tunnel.

NOTE

When configuring the source interface of a tunnel, note the following:

- Do not specify the tunnel interface of a GRE tunnel as its own source interface. You can specify the tunnel interface of another tunnel as the source interface.
- You can configure the virtual address of a VRRP group as the source address of a tunnel.
- Do not configure a bridge-if interface as the source interface of a tunnel.

Step 6 Run **destination** [**vpn-instance** *vpn-instance-name*] *dest-ip-address*

A destination address is specified for the tunnel.

If a customer edge (CE) is connected to a provider edge (PE) through the GRE tunnel, specify a virtual private network (VPN) instance to add the tunnel interface to a private network routing table when configuring the destination address for the tunnel.

Step 7 (Optional) Run **tunnel route-via** *interface-type interface-number* { **mandatory** | **preferred** }

The routing outbound interface for the GRE tunnel is specified.

By default, a GRE tunnel does not have a routing outbound interface.

GRE packets are forwarded based on routing tables. If there are multiple equal-cost routes to the destination address, GRE packets are shared among these routes. In some situations, the actual routing outbound interface of GRE packets transmitted over a GRE tunnel may be the routing outbound interface for another GRE tunnel. If Unicast Reverse Path Forwarding (URPF) is enabled on the next hop device, this device checks the source addresses of these GRE packets to identify their outbound interface and then determines whether the outbound interface for these packets are the same as the actual interface that receives these packets.

After the device finds that the outbound interface for these packets is different from the actual interface that receives these packets, the device drops these packets. To solve this problem, run the **tunnel route-via** command to specify the routing outbound interface for each GRE tunnel.

If you configure **mandatory** when running the **tunnel route-via** command, traffic is strictly forwarded through the specified routing outbound interface. Specifically, if the available routing outbound interfaces for GRE packets transmitted over a GRE tunnel do not include the routing outbound interface specified for the tunnel, packets cannot be forwarded. If you configure **preferred** when running the **tunnel route-via** command, traffic is preferentially forwarded through the specified routing outbound interface. Specifically, if the available routing outbound interfaces for GRE packets transmitted over a GRE tunnel do not include the routing outbound interface specified for the tunnel, packets can still be forwarded through available routing outbound interfaces.

Step 8 (Optional) Run **mtu mtu**

A maximum transmission unit (MTU) is configured for the tunnel interface.

By default, the MTU of a tunnel interface is 1500 bytes.

NOTE

To change the MTU of a tunnel interface, run the **shutdown** command and then the **undo shutdown** command on the interface to make the new MTU effective.

Step 9 (Optional) Run **description text**

An interface description is provided.

By default, the following description is provided for the tunnel interface: **HUAWEI, AR Series, Tunnel interface-number Interface**.

For example, the description of Tunnel0/0/1 is "**HUAWEI, AR Series, Tunnel0/0/1 Interface**."

Step 10 Run either of the following commands to specify an IP address for the tunnel interface.

- Specify an IP address.
 - Specify an IPv4 address for the tunnel interface when IPv4 networks communicate using the GRE tunnel.

Run **ip address ip-address { mask | mask-length } [sub]**

An IPv4 address is specified for the tunnel interface.

- Specify an IPv6 address for the tunnel interface when IPv6 networks communicate using the GRE tunnel.

Run **ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }**

An IPv6 address is specified for the tunnel interface.

NOTE

Before specifying an IPv6 address for an interface, run the **ipv6** command in the system view to enable IPv6 packet forwarding and run the **ipv6 enable** command on the interface to enable the IPv6 function.

- Borrow an IP address.

Run **ip address unnumbered interface interface-type interface-number**

The tunnel interface is configured to borrow an IP address.

 **NOTE**

A tunnel interface cannot borrow an IPv6 address.

----End

3.6.2 Configuring a Route on a Tunnel Interface

Context

GRE-encapsulated packets can be correctly forwarded only when a local and a remote device on the backbone network has a reachable route to each other and the route passes through the tunnel interfaces on the devices. The route can be a static route or a dynamic route.

 **NOTE**

As shown in [Figure 3-16](#), the X network runs the protocol frame relay (FR), High-Level Data Link Control (HDLC), or Point-to-Point Protocol (PPP). If packets from the X network need to be transparently transmitted over an IP network, the route configured on a tunnel interface does not take effect. In this case, you need to perform steps in [3.6.3 \(Optional\) Configuring the Link Bridge Function](#).

Router_1 in [Figure 3-16](#) is used as an example to illustrate the configuration notes.

- When configuring a static route, configure a route on both the local and the remote devices. Set the destination address of a static route to the original destination address of original packets (address of GE2/0/0 on Router_2), and set the outbound interface to the tunnel interface on the local device (Tunnel0/0/1 on Router_1).
- When configuring a dynamic routing protocol, configure the protocol on both the tunnel interface and the interface connected to the network running the X protocol.

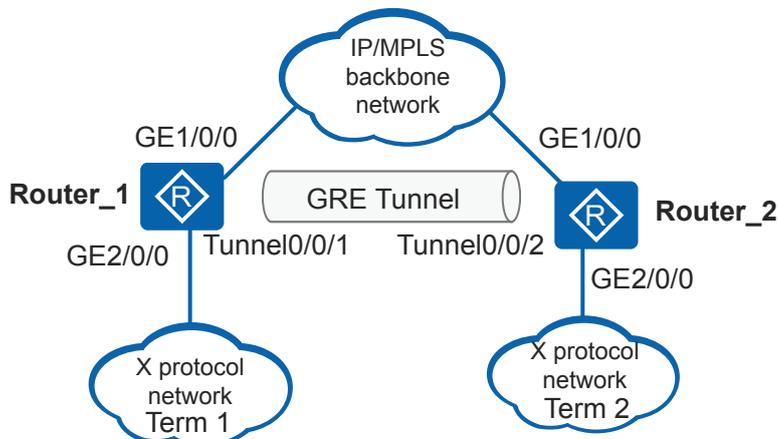
For example, as shown in [Figure 3-16](#), you must configure the dynamic routing protocol on the tunnel interface and GE2/0/0 connected to the network running the X protocol, and set the outbound interface to GE2/0/0 on Router_2 in the routing table to Tunnel0/0/1.

In practice, you must configure different types of routing protocols or different processes of the same type of routing protocol to advertise routes for the tunnel interface and the backbone network. This ensures that user packets are forwarded by a physical interface rather than the tunnel interface.

 **NOTE**

When a dynamic routing protocol is configured and the route import function is enabled on the tunnel interface, use a dynamic route or a 32-bit host route to implement interworking with the destination address. This ensures that the route to the destination address is not advertised to the tunnel interface, preventing tunnel flapping.

Figure 3-16 Networking for configuring a dynamic routing protocol for GRE



Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Choose either of the following methods to configure a route passing through a tunnel interface:

- Run **ip route-static** *ip-address* { *mask* | *mask-length* } { *next-hop-address* | **tunnel interface-number** [*next-hop-address*] } [**description text**]
A static route is configured.
- Configure a dynamic routing protocol. Dynamic routing can be implemented using Interior Gateway Protocol (IGP) or EGP, such as OSPF and RIP. For details on how to configure a dynamic routing protocol, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

When IPv6 networks communicate with each other over the GRE tunnel, you must configure an IPv6 routing protocol on the tunnel interface and the physical interface connected to the network running IPv6.

----End

3.6.3 (Optional) Configuring the Link Bridge Function

Context

Customers want to use GRE to transparently transmit FR, HDLC, PPP, or Ethernet packets over networks of a different network layer protocol, such as the IPv4 network. To transmit FR, HDLC, PPP, or Ethernet packets over a GRE tunnel, they need to configure the link bridge function to bind a serial, an Ethernet, a GE, a XGE, or a VLANIF interface with a tunnel interface, so that packets received from the serial, Ethernet, GE, XGE, or VLANIF interface can be directly sent out from the tunnel interface bound to it.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Run **link-bridge tag-id interface interface-type interface-number out-interface interface-type interface-number [untagged | tagged vlan-id]**

The link-bridge command binds an inbound interface to an outbound interface, so that packets received from the specified inbound interface can only be sent through the specified outbound interface.

By default, the link bridge function is not configured.

When configuring the link bridge function, pay attention to the following points:

- After you configure the link bridge function, the protocol status of the inbound interface turns Down and network-layer configurations on the inbound interface do not take effect. The inbound interface only functions as a bridge.
- After you configure this command, the protocol status of the inbound interface which is a serial interface turns Down and network-layer configurations on the inbound interface do not take effect. The inbound interface only functions as a bridge.
- After link bridge is bound to an interface, the interface does not support QoS. The inbound interface to which link bridge is bound supports traffic policy, traffic policing, traffic statistics collection, and mapping between 802.1p and DSCP priorities.
- Two link bridges must be configured with different tag IDs. The tag ID of a link bridge must be globally unique. If the tag ID of two link bridges is the same, an error message is displayed.
- Only one link bridge can be configured on a physical interface. If two link bridges are configured on the same physical interface, an error message is displayed.
- Only one link bridge can be configured on a tunnel interface. If two link bridges are configured on the same tunnel interface, an error message is displayed.
- If you specify the untagged mode, Ethernet packets transmitted over a GRE tunnel do not contain VLAN tags; otherwise, packets contain VLAN tags. You can determine whether to configure the untagged mode based on your actual networking to ensure normal traffic transmission over a tunnel. The following table describes the packet transmission rules.

Interface Type	Traffic Flow	Default Processing on Ethernet Packet	With Tag	Without Tag (Untagged)
Layer 2 Ethernet interface	From an Ethernet interface to a tunnel interface	The interface transparently transmits the packet.	The interface adds an outer tag to the packet before sending the packet.	If the packet contains a tag, the interface removes the tag before sending the packet.

Interface Type	Traffic Flow	Default Processing on Ethernet Packet	With Tag	Without Tag (Untagged)
	From a tunnel interface to an Ethernet interface	The interface transparently transmits the packet.	If the tag differs from the specified one, the interface discards the packet. Otherwise, the interface removes the outer tag before sending the packet.	The interface transparently transmits the packet.
Layer 3 Ethernet interface	From an Ethernet interface to a tunnel interface	The interface transparently transmits the packet.	The interface adds an outer tag to the packet before sending the packet.	If the packet contains a tag, the interface removes the tag before sending the packet.
	From a tunnel interface to an Ethernet interface	The interface transparently transmits the packet.	If the tag differs from the specified one, the interface discards the packet. Otherwise, the interface removes the outer tag before sending the packet.	The interface transparently transmits the packet.
VLANIF interface	From a VLANIF interface to a tunnel interface	If the VLAN ID in the packet is the same as the PVID, the interface removes the tag; otherwise, the interface transparently transmits the packet.	The interface adds an outer tag to the packet before sending the packet.	If the packet contains a tag, the interface removes the tag before sending the packet.

Interface Type	Traffic Flow	Default Processing on Ethernet Packet	With Tag	Without Tag (Untagged)
	From a tunnel interface to a VLANIF interface	The device checks whether the VLAN ID in the packet is the same as that of the VLANIF interface. If so, the interface sends the packet; otherwise, the interface discards the packet.	If the tag differs from the specified one, the interface discards the packet. Otherwise, the interface removes the outer tag before sending the packet.	The interface adds the VLAN ID of the VLANIF interface to the packet before sending the packet.
Ethernet sub-interface	From an Ethernet sub-interface to a tunnel interface	The interface transparently transmits the packet.	The interface adds an outer tag to the packet before sending the packet.	If the packet contains a tag, the interface removes the tag before sending the packet.
	From a tunnel interface to an Ethernet sub-interface	The device checks whether the VLAN ID in the packet is the same as that of the Dot1q or QinQ sub-interface. If so, the interface sends the packet; otherwise, the interface discards the packet.	If the tag differs from the specified one, the interface discards the packet. Otherwise, the interface removes the outer tag before sending the packet.	The interface adds the VLAN ID of the Dot1q or QinQ sub-interface to the packet before sending the packet.

----End

3.6.4 (Optional) Configuring a Security Mechanism for GRE

Context

You can configure the end-to-end check or a key for both ends of a GRE tunnel to improve the GRE tunnel security. This mechanism prevents devices from incorrectly identifying and receiving invalid packets.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 3 Run **gre checksum**

The checksum verification function is enabled for the GRE tunnel.

By default, the end-to-end checksum verification function is disabled on a tunnel.

Step 4 Run **gre key** { **plain** *key-number* | [**cipher**] *plain-cipher-text* }

A key is configured for the GRE tunnel.

Specify the same value for the parameter key number on tunnel interfaces on both ends of the GRE tunnel, or configure no key number for either end of the tunnel.

By default, no key is configured for the GRE tunnel.

 **NOTE**

If **plain** is selected, the key is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the key in cipher text.

The commands in Step 3 and Step 4 are independent of each other.

----End

3.6.5 (Optional) Enabling the Keepalive Detection Function for GRE

Context

After the Keepalive detection function is enabled, the local device periodically sends Keepalive probes to the peer to check tunnel connectivity. If the peer is reachable, the local device receives a reply packet from the peer. Otherwise, the local device cannot receive a reply packet, and then disconnects the tunnel connection.

The Keepalive detection function takes effect on one end of the tunnel as long as this end is configured with the Keepalive detection function. The other end of the tunnel is not required to have the Keepalive detection function configured. To ensure that both ends of the tunnel can detect whether the peer is reachable, you are advised to enable the Keepalive detection function on both ends.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 3 Run **keepalive** [**period** *period* [**retry-times** *retry-times*]]

The Keepalive detection function is enabled for GRE.

By default, the Keepalive detection function of GRE is disabled.

---End

3.6.6 (Optional) Configuring Ethernet over GRE

Context

Customers want to use GRE to transparently transmit Ethernet packets over networks of a different network layer protocol, such as the IPv4 network. Ethernet over GRE can be configured to transparently transmit Ethernet packets over a GRE tunnel. After you bind Layer 2 virtual Ethernet (VE) interfaces to the LAN-side physical Ethernet interface and WAN-side tunnel interface of a device, Ethernet packets received from the LAN-side interface are forwarded by the VE interfaces to the WAN-side tunnel interface. The WAN-side tunnel interface encapsulates the packets using GRE and transparently transmits the packets over a GRE tunnel.

Procedure

- Configuring a LAN-side physical Ethernet interface
 - a. Run **system-view**

The system view is displayed.
 - b. Run **interface virtual-ethernet** *ve-number*

A VE interface is created and the VE interface view is displayed.
 - c. Run **portswitch**

The VE interface is changed from Layer 3 mode to Layer 2 mode.

By default, a VE interface works in Layer 3 mode.
 - d. Perform the following configurations on a Layer 2 VE interface:
 - VLAN configuration: Configuring VLAN Assignment
 - QinQ configuration: Configuring Basic QinQ, Configuring Selective QinQ, and Configuring the TPID Value in an Outer VLAN Tag
 - VLAN mapping configuration: Configuring VLAN ID-based VLAN Mapping
 - e. Run **quit**

Return to the system view.
 - f. Run **interface** *interface-type interface-number*

The Ethernet interface view is displayed.

- g. Run **map interface virtual-ethernet** *ve-number*

The Layer 2 VE interface is bound to the physical Ethernet interface.

By default, no Layer 2 VE interface is bound to a physical Ethernet interface.

- Configuring a WAN-side tunnel interface

- a. Run **system-view**

The system view is displayed.

- b. (Optional) Run **gre map virtual-ethernet forward-broadcast disable**

A VE interface from forwarding broadcast, multicast, and unknown unicast packets to devices is disabled in the same VLAN .

By default, a VE interface can forward broadcast, multicast, and unknown unicast packets to devices in the same VLAN.

If branch CPE fails to obtain the MAC address of the HQ CPE, it will send broadcast, multicast, and unknown unicast packets to the VE interface bound to the tunnel interface of the HQ CPE. The VE interface then forwards the packets to other CPEs in the same VLAN. This consumes large network bandwidth, increases the workload of the HQ CPE, and may lead to drop of normal packets. To prevent this problem, you can configure the **gre map virtual-ethernet forward-broadcast disable** command on the HQ CPE to disable the VE interface from forwarding broadcast, multicast, and unknown unicast packets to other CPEs in the same VLAN.

- c. Run **interface virtual-ethernet** *ve-number*

A VE interface is created and the VE interface view is displayed.

- d. Run **portswitch**

The VE interface is changed from Layer 3 mode to Layer 2 mode.

By default, a VE interface works in Layer 3 mode.

- e. Perform the following configurations on a Layer 2 VE interface:

- VLAN configuration: Configuring VLAN Assignment
- QinQ configuration: Configuring Basic QinQ, Configuring Selective QinQ, and Configuring the TPID Value in an Outer VLAN Tag
- VLAN mapping configuration: Configuring VLAN ID-based VLAN Mapping

- f. Run **quit**

Return to the system view.

- g. Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

- h. Run **map interface virtual-ethernet** *ve-number*

The Layer 2 VE interface is bound to the tunnel interface.

By default, no Layer 2 VE interface is bound to a tunnel interface.

----End

3.6.7 (Optional) Configuring Ethernet over mGRE

Context

When the Hub and Spokes on a DSVPN network need to transparently transmit Ethernet packets, Ethernet over mGRE can be configured. After you bind Layer 2 virtual Ethernet (VE) interfaces to the LAN-side physical Ethernet interface and WAN-side tunnel interface of a device, Ethernet packets received from the LAN-side interface are forwarded by the VE interfaces to the WAN-side tunnel interface. The WAN-side tunnel interface encapsulates the packets using GRE and transparently transmits the packets over mGRE tunnels.

Perform the following configurations on the Hub and Spokes.

Procedure

- Configuring a LAN-side physical Ethernet interface
 - a. Run **system-view**

The system view is displayed.
 - b. Run **interface virtual-ethernet** *ve-number*

A VE interface is created and the VE interface view is displayed.
 - c. Run **portswitch**

The VE interface is changed from Layer 3 mode to Layer 2 mode.
By default, a VE interface works in Layer 3 mode.
 - d. Perform the following configurations on a Layer 2 VE interface:
 - VLAN configuration: Configuring VLAN Assignment
 - QinQ configuration: Configuring Basic QinQ, Configuring Selective QinQ, and Configuring the TPID Value in an Outer VLAN Tag
 - VLAN mapping configuration: Configuring VLAN ID-based VLAN Mapping
 - e. Run **quit**

Return to the system view.
 - f. Run **interface** *interface-type interface-number*

The Ethernet interface view is displayed.
 - g. Run **map interface virtual-ethernet** *ve-number*

The Layer 2 VE interface is bound to the physical Ethernet interface.
By default, no Layer 2 VE interface is bound to a physical Ethernet interface.
- Configuring a WAN-side tunnel interface
 - a. Run **system-view**

The system view is displayed.
 - b. (Optional) Run **gre map virtual-ethernet forward-broadcast disable**

A VE interface from forwarding broadcast, multicast, and unknown unicast packets to devices is disabled in the same VLAN .
By default, a VE interface can forward broadcast, multicast, and unknown unicast packets to devices in the same VLAN.

If branch CPE fails to obtain the MAC address of the HQ CPE, it will send broadcast, multicast, and unknown unicast packets to the VE interface bound to the tunnel interface of the HQ CPE. The VE interface then forwards the packets to other CPEs in the same VLAN. This consumes large network bandwidth, increases the workload of the HQ CPE, and may lead to drop of normal packets. To prevent this problem, you can configure the **gre map virtual-ethernet forward-broadcast disable** command on the HQ CPE to disable the VE interface from forwarding broadcast, multicast, and unknown unicast packets to other CPEs in the same VLAN.

- c. Run **interface virtual-ethernet** *ve-number*

A VE interface is created and the VE interface view is displayed.

- d. Run **portswitch**

The VE interface is changed from Layer 3 mode to Layer 2 mode.

By default, a VE interface works in Layer 3 mode.

- e. Perform the following configurations on a Layer 2 VE interface:

- VLAN configuration: Configuring VLAN Assignment
- QinQ configuration: Configuring Basic QinQ, Configuring Selective QinQ, and Configuring the TPID Value in an Outer VLAN Tag
- VLAN mapping configuration: Configuring VLAN ID-based VLAN Mapping

- f. Run **quit**

Return to the system view.

- g. Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

- h. Run **map interface virtual-ethernet** *ve-number*

The Layer 2 VE interface is bound to the tunnel interface.

By default, no Layer 2 VE interface is bound to a tunnel interface.

---End

3.6.8 (Optional) Configuring the DF Flag Bit for GRE Packets

Context

The length of GRE-encapsulated packets may exceed the maximum transmission unit (MTU) of the outbound interface on the local device. If the remote device on a GRE tunnel does not support fragmentation and reassembly, it cannot decapsulate packets and will discard or invalidly process packets, affecting packet transmission.

To prevent packet loss, you can set the DF flag bit for GRE packets to clear to allow fragmentation of GRE packets. After the DF flag bit is set to clear, the local device pre-calculates the packet length. If the packet length after GRE encapsulation exceeds the MTU, the local device fragments the GRE packets and encapsulates each fragment. After packets reach the GRE remote device, the remote device can decapsulate the fragments without having to reassemble them. The decapsulated packets will be forwarded normally.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **gre df-bit { clear | set | copy }**

The DF flag bit is set for GRE packets.

By default, the DF flag bit for GRE packets uses the clear mode.

---End

3.6.9 Verifying the GRE Tunnel Configuration

Prerequisites

The configurations of a GRE tunnel are complete.

Procedure

- Run the **display interface tunnel** [*interface-number*] command to view the status of the tunnel interface.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [*slots*] } command to view tunnel information.
- Run the **display ip routing-table** command to view the IPv4 routing table to check whether the outbound interface for a route to the specified destination address is a tunnel interface.
- Run the **display ipv6 routing-table** command to view the IPv6 routing table to check whether the outbound interface for a route to the specified destination address is a tunnel interface.
- Run the **ping -a source-ip-address host** command to check whether the local tunnel interface can ping the remote tunnel interface successfully.
- After Keepalive detection is enabled, run the **display keepalive packets count** command in the tunnel interface view to view the number of Keepalive probes and Keepalive reply packets on the tunnel interface.

---End

3.7 Maintaining the GRE Tunnel

This section describes how to collect and view statistics on tunnel interfaces, monitor the GRE running status, and reset the Keepalive packet statistics on tunnel interfaces.

3.7.1 Collecting and Viewing Statistics on Tunnel Interfaces

Context

To check the network status or locate network faults, you can enable traffic statistics collection on tunnel interfaces and collect traffic statistics on the tunnel interfaces.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

Step 3 Run **statistic enable** { **inbound** | **outbound** }

The traffic statistics collection is enabled on a Tunnel interface.

By default, traffic statistics collection is disabled on a Tunnel interface.

Step 4 Run **display interface tunnel**

The traffic statistics on tunnel interfaces is displayed.

---End

Follow-up Procedure

Run the **reset counters interface tunnel** [*interface-number*] command in the user view to reset statistics on a tunnel interface to avoid interference of the original statistics.



NOTICE

After you reset the statistics on a specified tunnel interface, statistics information about the number of sent and received packets and the packet transmission rate are cleared. Exercise caution when you run this command.

3.7.2 Monitoring the GRE Running Status

Context

In routine maintenance, you can run the following commands in any view to check the GRE running status, whether GRE tunnel interfaces are Up, whether error packets exist, and whether packets are properly forwarded to the destination address through the tunnel interface.

Procedure

- Run the **display interface tunnel** command to view the running status of the tunnel interface.
- Run the **display ip routing-table** command to view the IPv4 routing table to check whether the outbound interface for a route to the specified destination address is a tunnel interface.
- Run the **display ipv6 routing-table** command to view the IPv6 routing table to check whether the outbound interface for a route to the specified destination address is a tunnel interface.

- Run the **display ip routing-table vpn-instance** *vpn-instance* command to check VPN routing information on the tunnel interface.
When a tunnel interface is bound to a VPN instance, specify a VPN instance to check the VPN routing information.

----End

3.7.3 Resetting the Keepalive Packet Statistics on a Tunnel Interface

Context

When you need to calculate and analyze statistics on a tunnel interface, reset the statistics to avoid interference of the original statistics.

After you reset the Keepalive packet statistics on a specified tunnel interface, statistics on the number of Keepalive probes and Keepalive reply packets are cleared.



NOTICE

The cleared Keepalive packet statistics cannot be restored. Exercise caution when you run the command.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 3 Run **reset keepalive packets count**

The Keepalive packet statistics on the tunnel interface is reset.

----End

3.8 Configuration Examples for GRE

This section provides GRE configuration examples, including networking requirements, configuration roadmap, and configuration procedure.

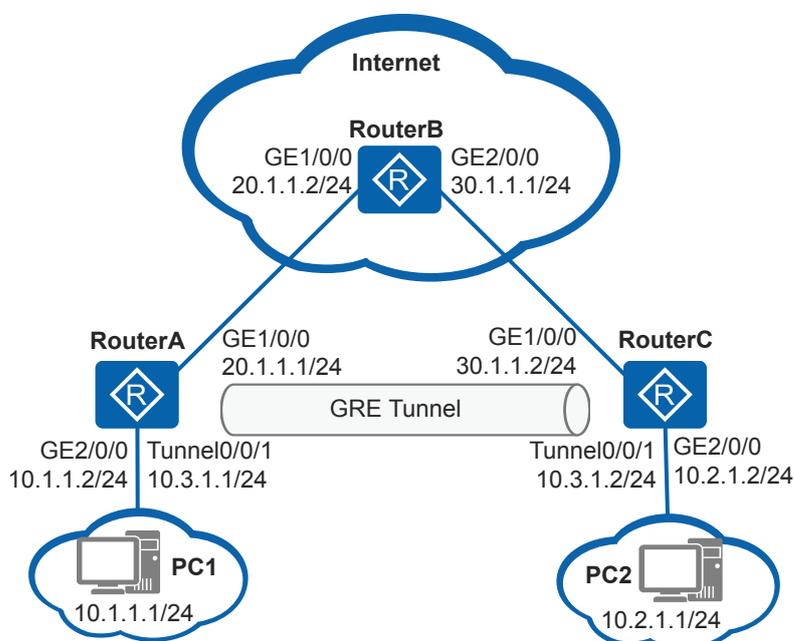
3.8.1 Example for Configuring a Static Route for GRE to Implement Interworking Between IPv4 Networks

Networking Requirements

As shown in [Figure 3-17](#), RouterA, RouterB, and RouterC run OSPF to implement interworking over the public network. PC1 and PC2 run the IPv4 proprietary protocol and communicate with each other over the public network.

PC1 and PC2 use RouterA and RouterC as their default gateways respectively.

Figure 3-17 Configuring a static route for GRE



Configuration Roadmap

To allow PC1 to communicate with PC2, you can configure a direct link between RouterA and RouterC to set up a GRE tunnel and configure a static route to forward packets through tunnel interfaces to the peer.

The configuration roadmap is as follows:

1. Run OSPF on the devices to implement interworking among them.
2. Create tunnel interfaces on RouterA and RouterC to set up a GRE tunnel, and configure a static route passing through tunnel interfaces on RouterA and RouterC, so that traffic between PC1 and PC2 can be transmitted over the GRE tunnel.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 20.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.2 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 20.1.1.2 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 30.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] ip address 30.1.1.2 255.255.255.0
[RouterC-GigabitEthernet1/0/0] quit
[RouterC] interface gigabitethernet 2/0/0
[RouterC-GigabitEthernet2/0/0] ip address 10.2.1.2 255.255.255.0
[RouterC-GigabitEthernet2/0/0] quit
```

Step 2 Configure OSPF on the devices.

Configure RouterA.

```
[RouterA] ospf 1
[RouterA-ospf-1] area 0
[RouterA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterA-ospf-1-area-0.0.0.0] quit
[RouterA-ospf-1] quit
```

Configure RouterB.

```
[RouterB] ospf 1
[RouterB-ospf-1] area 0
[RouterB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] quit
```

Configure RouterC.

```
[RouterC] ospf 1
[RouterC-ospf-1] area 0
[RouterC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.0] quit
[RouterC-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. The command output shows that they have learned the OSPF route destined for the network segment of the peer.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table protocol ospf
<keyword conref="../commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
-----
Public routing table :
OSPF
    Destinations : 1          Routes :
1
OSPF routing table status :
<Active>
    Destinations : 1          Routes :
1
Destination/Mask    Proto  Pre  Cost    Flags NextHop
Interface
          30.1.1.0/24  OSPF   10   2        D   20.1.1.2
GigabitEthernet1/0/0
OSPF routing table status :
<Inactive>
    Destinations : 0          Routes :
0
```

Step 3 Configure a tunnel interface.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] tunnel-protocol gre
[RouterA-Tunnel0/0/1] ip address 10.3.1.1 255.255.255.0
[RouterA-Tunnel0/0/1] source 20.1.1.1
[RouterA-Tunnel0/0/1] destination 30.1.1.2
[RouterA-Tunnel0/0/1] quit
```

Configure RouterC.

```
[RouterC] interface tunnel 0/0/1
[RouterC-Tunnel0/0/1] tunnel-protocol gre
[RouterC-Tunnel0/0/1] ip address 10.3.1.2 255.255.255.0
[RouterC-Tunnel0/0/1] source 30.1.1.2
[RouterC-Tunnel0/0/1] destination 20.1.1.1
[RouterC-Tunnel0/0/1] quit
```

After the configuration is complete, the tunnel interfaces turn Up and can ping each other. This indicates that a direct tunnel has been set up.

The command output on RouterA is used as an example.

```
[RouterA] ping -a 10.3.1.1 10.3.1.2
PING 10.3.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.3.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
```

```
round-trip min/avg/max = 1/1/1 ms
```

Step 4 Configure a static route.

Configure RouterA.

```
[RouterA] ip route-static 10.2.1.0 255.255.255.0 tunnel 0/0/1
```

Configure RouterC.

```
[RouterC] ip route-static 10.1.1.0 255.255.255.0 tunnel 0/0/1
```

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. The command output shows the static route from the tunnel interface to the user-side network segment.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table 10.2.1.0
<keyword conref=" ../commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto  Pre  Cost           Flags NextHop         Interface
-----
10.2.1.0/24        Static  60   0              D    10.3.1.2            Tunnel0/0/1
```

PC1 and PC2 can ping each other.

---End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
ip address 20.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.1.2 255.255.255.0
#
interface Tunnel0/0/1
ip address 10.3.1.1 255.255.255.0
tunnel-protocol gre
source 20.1.1.1
destination 30.1.1.2
#
ospf 1
area 0.0.0.0
network 20.1.1.0 0.0.0.255
#
ip route-static 10.2.1.0 255.255.255.0 Tunnel0/0/1
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
ip address 20.1.1.2 255.255.255.0
#
```

```
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
#
ospf 1
area 0.0.0.0
network 20.1.1.0 0.0.0.255
network 30.1.1.0 0.0.0.255
#
return
```

- Configuration file of RouterC

```
#
sysname RouterC
#
interface GigabitEthernet1/0/0
ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.2.1.2 255.255.255.0
#
interface Tunnel0/0/1
ip address 10.3.1.2 255.255.255.0
tunnel-protocol gre
source 30.1.1.2
destination 20.1.1.1
#
ospf 1
area 0.0.0.0
network 30.1.1.0 0.0.0.255
#
ip route-static 10.1.1.0 255.255.255.0 Tunnel0/0/1
#
return
```

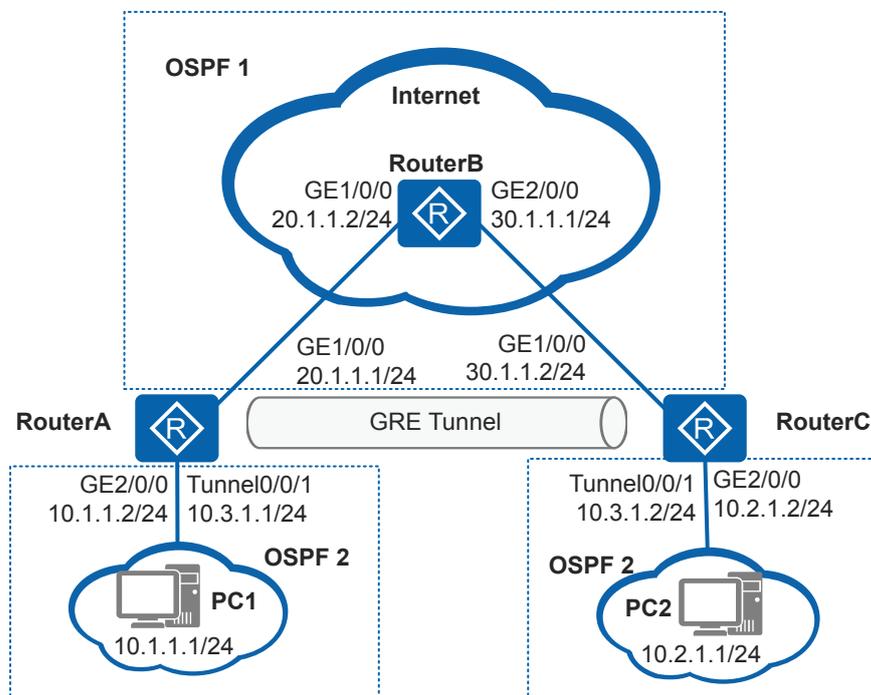
3.8.2 Example for Configuring OSPF for GRE to Implement Interworking Between IPv4 Networks

Networking Requirements

As shown in [Figure 3-18](#), RouterA, RouterB, and RouterC run OSPF to implement interworking over the public network. PC1 and PC2 run the IPv4 proprietary protocol and communicate with each other over the public network. Transmission of private data must be reliable.

PC1 and PC2 use RouterA and RouterC as their default gateways respectively.

Figure 3-18 Configuring a dynamic routing protocol for GRE



Configuration Roadmap

You can set up a directly connected GRE tunnel between RouterA and RouterC and configure OSPF on tunnel interfaces and interfaces connected to the private networks to allow PC1 to communicate with PC2. To monitor the tunnel link status, enable Keepalive detection on tunnel interfaces on both ends of the GRE tunnel.

The configuration roadmap is as follows:

1. Configure an IGP (OSPF process 1 in this example) to implement interworking among the devices.
2. Set up a GRE tunnel between devices connected to the PCs, enable Keepalive detection, and run an IGP (OSPF process 2 in this example) on the network segments connected to the PCs to transmit traffic between PC1 and PC2 over the GRE tunnel.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 20.1.1.1 255.255.255.0
```

```
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.2 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 20.1.1.2 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 30.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] ip address 30.1.1.2 255.255.255.0
[RouterC-GigabitEthernet1/0/0] quit
[RouterC] interface gigabitethernet 2/0/0
[RouterC-GigabitEthernet2/0/0] ip address 10.2.1.2 255.255.255.0
[RouterC-GigabitEthernet2/0/0] quit
```

Step 2 Configure OSPF on the devices.

Configure RouterA.

```
[RouterA] ospf 1
[RouterA-ospf-1] area 0
[RouterA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterA-ospf-1-area-0.0.0.0] quit
[RouterA-ospf-1] quit
```

Configure RouterB.

```
[RouterB] ospf 1
[RouterB-ospf-1] area 0
[RouterB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] quit
```

Configure RouterC.

```
[RouterC] ospf 1
[RouterC-ospf-1] area 0
[RouterC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.0] quit
[RouterC-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. The command output shows that they have learned the OSPF route destined for the network segment of the peer.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table protocol ospf
<keyword conref="./commonterms/commonterms.xml#commonterms/route-flags"></keyword>
-----
Public routing table : OSPF
    Destinations : 1          Routes : 1

OSPF routing table status : <Active>
    Destinations : 1          Routes : 1
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
30.1.1.0/24	OSPF	10	2	D	20.1.1.2	GigabitEthernet1/0/0

OSPF routing table status : <Inactive>
Destinations : 0 Routes : 0

Step 3 Configure a tunnel interface.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] tunnel-protocol gre
[RouterA-Tunnel0/0/1] ip address 10.3.1.1 255.255.255.0
[RouterA-Tunnel0/0/1] source 20.1.1.1
[RouterA-Tunnel0/0/1] destination 30.1.1.2
[RouterA-Tunnel0/0/1] keepalive
[RouterA-Tunnel0/0/1] quit
```

Configure RouterC.

```
[RouterC] interface tunnel 0/0/1
[RouterC-Tunnel0/0/1] tunnel-protocol gre
[RouterC-Tunnel0/0/1] ip address 10.3.1.2 255.255.255.0
[RouterC-Tunnel0/0/1] source 30.1.1.2
[RouterC-Tunnel0/0/1] destination 20.1.1.1
[RouterC-Tunnel0/0/1] keepalive
[RouterC-Tunnel0/0/1] quit
```

After the configuration is complete, the tunnel interfaces turn Up and can ping each other.

The command output on RouterA is used as an example.

```
[RouterA] ping -a 10.3.1.1 10.3.1.2
PING 10.3.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.3.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

Run the **display keepalive packets count** command to check the statistics on Keepalive packets.

The command output on RouterA is used as an example.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] display keepalive packets count
Send 10 keepalive packets to peers, Receive 10 keepalive response packets from peers
Receive 8 keepalive packets from peers, Send 8 keepalive response packets to peers.
```

Step 4 Configure OSPF on tunnel interfaces.

Configure RouterA.

```
[RouterA] ospf 2
[RouterA-ospf-2] area 0
[RouterA-ospf-2-area-0.0.0.0] network 10.3.1.0 0.0.0.255
[RouterA-ospf-2-area-0.0.0.0] network 10.1.1.0 0.0.0.255
```

```
[RouterA-ospf-2-area-0.0.0.0] quit
[RouterA-ospf-2] quit
```

Configure RouterC.

```
[RouterC] ospf 2
[RouterC-ospf-2] area 0
[RouterC-ospf-2-area-0.0.0.0] network 10.3.1.0 0.0.0.255
[RouterC-ospf-2-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[RouterC-ospf-2-area-0.0.0.0] quit
[RouterC-ospf-2] quit
```

Step 5 Verify the configuration.

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. The routing table of each router contains the OSPF route from the tunnel interface to the user-side network segment of the peer. In addition, the next hop of the route to the destination physical interface (30.1.1.0/24) of the tunnel is not a tunnel interface.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table protocol ospf
<keyword conref="./commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
-----
Public routing table : OSPF
      Destinations : 2          Routes : 2

OSPF routing table status : <Active>
      Destinations : 2          Routes : 2

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
-----
      10.2.1.0/24    OSPF    10   1563      D    10.3.1.2         Tunnel0/0/1
      30.1.1.0/24    OSPF    10    2         D    20.1.1.2
GigabitEthernet1/0/0

OSPF routing table status : <Inactive>
      Destinations : 0          Routes : 0
```

PC1 and PC2 can ping each other.

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
ip address 20.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.1.2 255.255.255.0
#
interface Tunnel0/0/1
ip address 10.3.1.1 255.255.255.0
tunnel-protocol gre
keepalive
source 20.1.1.1
destination 30.1.1.2
#
ospf 1
area 0.0.0.0
network 20.1.1.0 0.0.0.255
#
```

```
ospf 2
 area 0.0.0.0
  network 10.3.1.0 0.0.0.255
  network 10.1.1.0 0.0.0.255
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 20.1.1.0 0.0.0.255
  network 30.1.1.0 0.0.0.255
#
return
```

- Configuration file of RouterC

```
#
sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.2.1.2 255.255.255.0
#
interface Tunnel0/0/1
 ip address 10.3.1.2 255.255.255.0
 tunnel-protocol gre
 keepalive
 source 30.1.1.2
 destination 20.1.1.1
#
ospf 1
 area 0.0.0.0
  network 30.1.1.0 0.0.0.255
#
ospf 2
 area 0.0.0.0
  network 10.3.1.0 0.0.0.255
  network 10.2.1.0 0.0.0.255
#
return
```

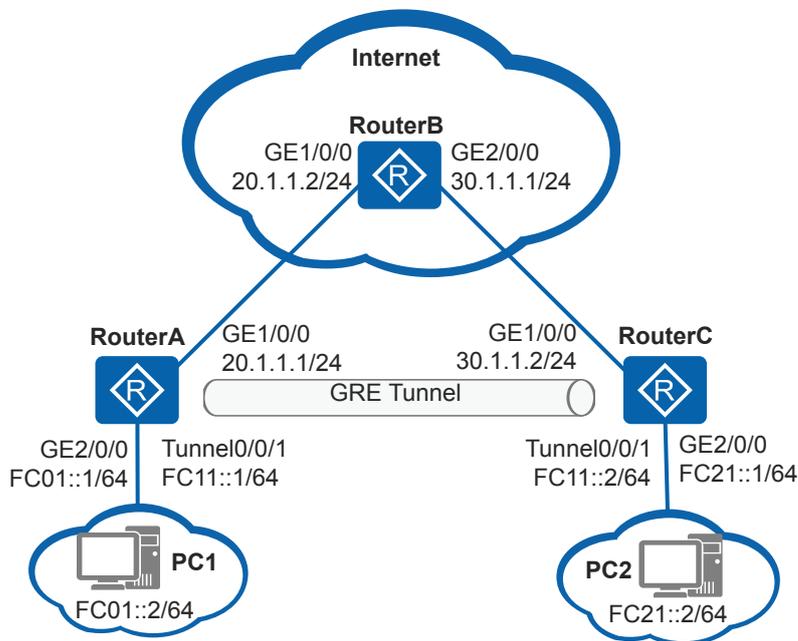
3.8.3 Example for Configuring a GRE Tunnel to Implement Interworking Between IPv6 Networks

Networking Requirements

As shown in [Figure 3-19](#), RouterA and RouterC on IPv6 networks connect to RouterB on an IPv4 network. PC1 and PC2 on the two IPv6 networks need to communicate with each other.

PC1 and PC2 use RouterA and RouterC as their default gateways respectively.

Figure 3-19 Configuring a GRE tunnel to implement interworking between IPv6 networks



Configuration Roadmap

To allow PC1 and PC2 on the IPv6 networks to communicate with each other, you can configure a direct link between RouterA and RouterC to set up a GRE tunnel and configure a static route to forward packets through tunnel interfaces to the peer.

The configuration roadmap is as follows:

1. Configure IP addresses for physical interfaces and configure an IPv4 static route to implement interworking over the IPv4 network.
2. Create tunnel interfaces on RouterA and RouterC to set up a GRE tunnel, and configure an IPv6 static route passing through tunnel interfaces on RouterA and RouterC, so that traffic between PC1 and PC2 can be transmitted over the GRE tunnel.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 20.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] ipv6
```

```
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ipv6 enable
[RouterA-GigabitEthernet2/0/0] ipv6 address FC01::1 64
[RouterA-GigabitEthernet2/0/0] quit
```

Configure RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 20.1.1.2 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 30.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] ip address 30.1.1.2 255.255.255.0
[RouterC-GigabitEthernet1/0/0] quit
[RouterC] ipv6
[RouterC] interface gigabitethernet 2/0/0
[RouterC-GigabitEthernet2/0/0] ipv6 enable
[RouterC-GigabitEthernet2/0/0] ipv6 address FC21::1 64
[RouterC-GigabitEthernet2/0/0] quit
```

Step 2 Configure an IPv4 static route.

Configure RouterA.

```
[RouterA] ip route-static 30.1.1.2 255.255.255.0 20.1.1.2
```

Configure RouterC.

```
[RouterC] ip route-static 20.1.1.1 255.255.255.0 30.1.1.1
```

Step 3 Configure a tunnel interface.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] tunnel-protocol gre
[RouterA-Tunnel0/0/1] ipv6 enable
[RouterA-Tunnel0/0/1] ipv6 address FC11::1 64
[RouterA-Tunnel0/0/1] source 20.1.1.1
[RouterA-Tunnel0/0/1] destination 30.1.1.2
[RouterA-Tunnel0/0/1] quit
```

Configure RouterC.

```
[RouterC] interface tunnel 0/0/1
[RouterC-Tunnel0/0/1] tunnel-protocol gre
[RouterC-Tunnel0/0/1] ipv6 enable
[RouterC-Tunnel0/0/1] ipv6 address FC11::2 64
[RouterC-Tunnel0/0/1] source 30.1.1.2
[RouterC-Tunnel0/0/1] destination 20.1.1.1
[RouterC-Tunnel0/0/1] quit
```

Step 4 Configure a static tunnel route.

Configure RouterA.

```
[RouterA] ipv6 route-static FC21::1 64 tunnel 0/0/1
```

Configure RouterC.

```
[RouterC] ipv6 route-static FC01::1 64 tunnel 0/0/1
```

Step 5 Verify the configuration.

Ping the IPv4 address of RouterA from RouterC. RouterC can receive a Reply packet from RouterA.

```
[RouterC] ping 20.1.1.1
PING 20.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 20.1.1.1: bytes=56 Sequence=1 ttl=255 time=84 ms
  Reply from 20.1.1.1: bytes=56 Sequence=2 ttl=255 time=27 ms
  Reply from 20.1.1.1: bytes=56 Sequence=3 ttl=255 time=25 ms
  Reply from 20.1.1.1: bytes=56 Sequence=4 ttl=255 time=3 ms
  Reply from 20.1.1.1: bytes=56 Sequence=5 ttl=255 time=24 ms

--- 20.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/32/84 ms
```

Ping the IPv6 address of RouterA from RouterC. RouterC can receive a Reply packet from RouterA.

```
[RouterC] ping ipv6 FC01::1
PING FC01::1 : 56 data bytes, press CTRL_C to break
  Reply from FC01::1
  bytes=56 Sequence=1 hop limit=64 time = 28 ms
  Reply from FC01::1
  bytes=56 Sequence=2 hop limit=64 time = 27 ms
  Reply from FC01::1
  bytes=56 Sequence=3 hop limit=64 time = 26 ms
  Reply from FC01::1
  bytes=56 Sequence=4 hop limit=64 time = 27 ms
  Reply from FC01::1
  bytes=56 Sequence=5 hop limit=64 time = 26 ms

--- FC01::1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 26/26/28 ms
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
ipv6
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ipv6 enable
 ipv6 address FC01::1/64
#
interface Tunnel0/0/1
 ipv6 enable
 ipv6 address FC11::1/64
 tunnel-protocol gre
 source 20.1.1.1
 destination 30.1.1.2
#
ip route-static 30.1.1.0 255.255.255.0 20.1.1.2
#
```

```
ipv6 route-static FC21:: 64 Tunnel0/0/1
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
#
return
```

- Configuration file of RouterC

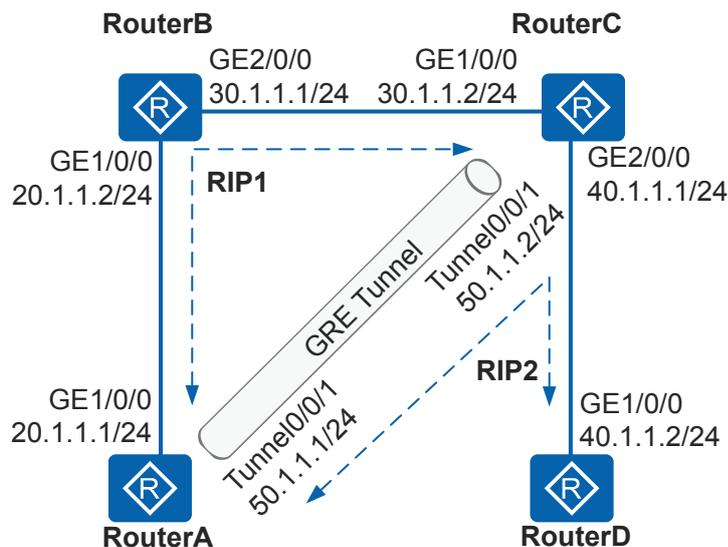
```
#
sysname RouterC
#
ipv6
#
interface GigabitEthernet1/0/0
ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ipv6 enable
ipv6 address FC21::1/64
#
interface Tunnel0/0/1
ipv6 enable
ipv6 address FC11::2/64
tunnel-protocol gre
source 30.1.1.2
destination 20.1.1.1
#
ip route-static 20.1.1.0 255.255.255.0 30.1.1.1
#
ipv6 route-static FC01:: 64 Tunnel0/0/1
#
return
```

3.8.4 Example for Enlarging the Operation Scope of a Network with a Hop Limit

Networking Requirements

As shown in [Figure 3-20](#), RouterA, RouterB, RouterC, and RouterD run RIP to implement interworking. Data sent from RouterA to RouterD must pass through only one hop. That is, the route cost is 1. RIP is deployed without changing the network topology. There are two hops between RouterA and RouterD. To reduce a hop, you need to set up a GRE tunnel between RouterA and RouterC. Although the logical hop count is 1, there are two devices on the path from RouterA to RouterD. Therefore, the hop count allowed on a RIP network is increased.

Figure 3-20 Enlarging the operation scope of a network with a hop limit



Configuration Roadmap

The configuration roadmap is as follows:

1. Run RIP process 1 on RouterA, RouterB, and RouterC to implement interworking among them.
2. Set up a GRE tunnel between RouterA and RouterC to hide RouterB.
3. Run RIP process 2 on RouterA, RouterC, and RouterD to forward packets over the GRE tunnel. The actual hop counts allowed on a RIP network is increased.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 20.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
```

Configure RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 20.1.1.2 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 30.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
```

```
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] ip address 30.1.1.2 255.255.255.0
[RouterC-GigabitEthernet1/0/0] quit
[RouterC] interface gigabitethernet 2/0/0
[RouterC-GigabitEthernet2/0/0] ip address 40.1.1.1 255.255.255.0
[RouterC-GigabitEthernet2/0/0] quit
```

Configure RouterD.

```
<Huawei> system-view
[Huawei] sysname RouterD
[RouterD] interface gigabitethernet 1/0/0
[RouterD-GigabitEthernet1/0/0] ip address 40.1.1.2 255.255.255.0
[RouterD-GigabitEthernet1/0/0] quit
```

Step 2 Run RIP process 1 on devices.

Configure RouterA.

```
[RouterA] rip 1
[RouterA-rip-1] version 2
[RouterA-rip-1] network 20.0.0.0
[RouterA-rip-1] quit
```

Configure RouterB.

```
[RouterB] rip 1
[RouterB-rip-1] version 2
[RouterB-rip-1] network 20.0.0.0
[RouterB-rip-1] network 30.0.0.0
[RouterB-rip-1] quit
```

Configure RouterC.

```
[RouterC] rip 1
[RouterC-rip-1] version 2
[RouterC-rip-1] network 30.0.0.0
[RouterC-rip-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. The command output shows that they have learned the RIP route destined for the network segment of the peer.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table
<keyword conref=" ../commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
-----
Routing Tables: Public
Destinations : 8          Routes : 8

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
20.1.1.0/24         Direct  0    0        D    20.1.1.1
GigabitEthernet1/0/0
20.1.1.1/32         Direct  0    0        D    127.0.0.1
GigabitEthernet1/0/0
20.1.1.255/32       Direct  0    0        D    127.0.0.1
GigabitEthernet1/0/0
30.1.1.0/24         RIP    100  1        D    20.1.1.2
GigabitEthernet1/0/0
127.0.0.0/8         Direct  0    0        D    127.0.0.1          InLoopBack0
127.0.0.1/32       Direct  0    0        D    127.0.0.1          InLoopBack0
127.255.255.255/32 Direct  0    0        D    127.0.0.1          InLoopBack0
255.255.255.255/32 Direct  0    0        D    127.0.0.1          InLoopBack0
```

Step 3 Configure a tunnel interface.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] tunnel-protocol gre
[RouterA-Tunnel0/0/1] ip address 50.1.1.1 255.255.255.0
[RouterA-Tunnel0/0/1] source 20.1.1.1
[RouterA-Tunnel0/0/1] destination 30.1.1.2
[RouterA-Tunnel0/0/1] quit
```

Configure RouterC.

```
[RouterC] interface tunnel 0/0/1
[RouterC-Tunnel0/0/1] tunnel-protocol gre
[RouterC-Tunnel0/0/1] ip address 50.1.1.2 255.255.255.0
[RouterC-Tunnel0/0/1] source 30.1.1.2
[RouterC-Tunnel0/0/1] destination 20.1.1.1
[RouterC-Tunnel0/0/1] quit
```

After the configuration is complete, the tunnel interfaces turn Up and can ping each other.

The command output on RouterA is used as an example.

```
[RouterA] ping -a 50.1.1.1 50.1.1.2
PING 50.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 50.1.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 50.1.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 50.1.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 50.1.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 50.1.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 50.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

Step 4 Run RIP process 2 on tunnel interfaces.

Configure RouterA.

```
[RouterA] rip 2
[RouterA-rip-2] version 2
[RouterA-rip-2] network 50.0.0.0
[RouterA-rip-2] quit
```

Configure RouterC.

```
[RouterC] rip 2
[RouterC-rip-2] version 2
[RouterC-rip-2] network 50.0.0.0
[RouterC-rip-2] network 40.0.0.0
[RouterC-rip-2] quit
```

Configure RouterD.

```
[RouterD] rip 2
[RouterD-rip-2] version 2
[RouterD-rip-2] network 40.0.0.0
[RouterD-rip-2] quit
```

Step 5 Verify the configuration.

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterD. The command output shows that the cost of the route to the destination address of the peer device is 1.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table
<keyword conref=" ../commonterms/commonterms.xml#commonterms/route-flags"></
```

```
keyword>
-----
Routing Tables: Public
      Destinations : 12          Routes : 12

Destination/Mask    Proto    Pre  Cost           Flags NextHop         Interface
-----
      20.1.1.0/24    Direct  0    0              D    20.1.1.1
GigabitEthernet1/0/0
      20.1.1.1/32    Direct  0    0              D    127.0.0.1
GigabitEthernet1/0/0
      20.1.1.255/32  Direct  0    0              D    127.0.0.1
GigabitEthernet1/0/0
      30.1.1.0/24    RIP     100  1              D    20.1.1.2
GigabitEthernet1/0/0
      40.1.1.0/24    RIP     100  1              D    50.1.1.2      Tunnel0/0/1
      50.1.1.0/24    Direct  0    0              D    50.1.1.1      Tunnel0/0/1
      50.1.1.1/32    Direct  0    0              D    127.0.0.1     Tunnel0/0/1
      50.1.1.255/32  Direct  0    0              D    127.0.0.1     Tunnel0/0/1
      127.0.0.0/8     Direct  0    0              D    127.0.0.1     InLoopBack0
      127.0.0.1/32   Direct  0    0              D    127.0.0.1     InLoopBack0
127.255.255.255/32  Direct  0    0              D    127.0.0.1     InLoopBack0
255.255.255.255/32  Direct  0    0              D    127.0.0.1     InLoopBack0
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
ip address 20.1.1.1 255.255.255.0
#
interface Tunnel0/0/1
ip address 50.1.1.1 255.255.255.0
tunnel-protocol gre
source 20.1.1.1
destination 30.1.1.2
#
rip 1
version 2
network 20.0.0.0
#
rip 2
version 2
network 50.0.0.0
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
#
rip 1
version 2
network 20.0.0.0
network 30.0.0.0
#
return
```

- Configuration file of RouterC

```
#
 sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 40.1.1.1 255.255.255.0
#
interface Tunnel0/0/1
 ip address 50.1.1.2 255.255.255.0
 tunnel-protocol gre
 source 30.1.1.2
 destination 20.1.1.1
#
rip 1
 version 2
 network 30.0.0.0
#
rip 2
 version 2
 network 40.0.0.0
 network 50.0.0.0
#
return
```

- Configuration file of RouterD

```
#
 sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 40.1.1.2 255.255.255.0
#
rip 2
 version 2
 network 40.0.0.0
#
return
```

3.8.5 Example for Configuring BGP/MPLS IP VPN to Use a GRE Tunnel

Networking Requirements

 **NOTE**

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot be used in this scenario.

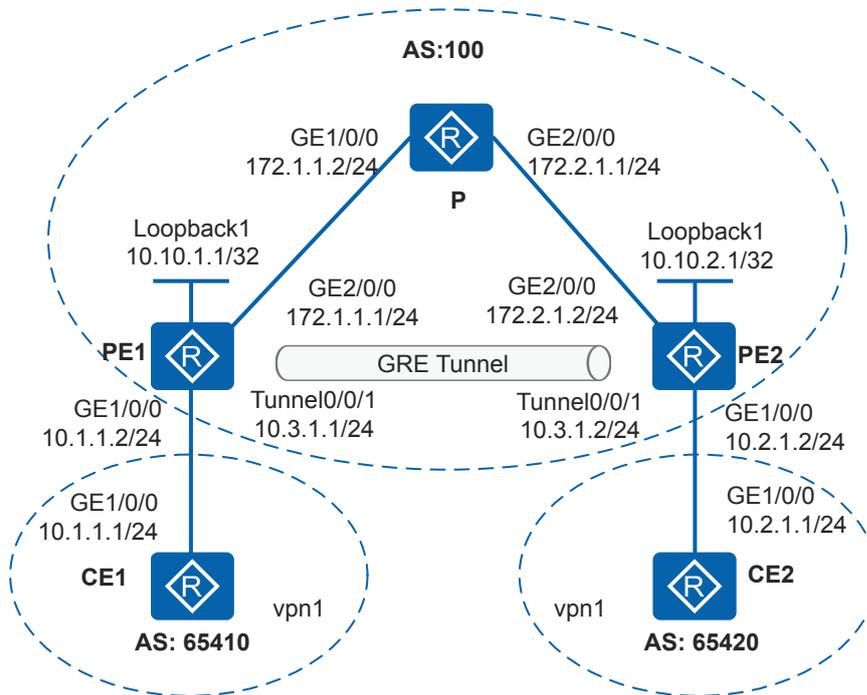
In [Figure 3-21](#):

- Branch 1 connects to the VPN backbone network through CE1 and PE1.
- Branch 2 connects to the VPN backbone network through CE2 and PE2.

On the backbone network, PEs provide MPLS functions, and the P does not provide MPLS functions.

The enterprise wants to establish a GRE tunnel between the PEs and use IP to forward VPN packets over the IP network.

Figure 3-21 Networking diagram for configuring BGP/MPLS IP VPN to use a GRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the PEs and P to implement IP connectivity on the backbone network.
2. Create a GRE tunnel between PEs so that VPN packets can be transmitted over the GRE tunnel.
3. Configure VPN instances on PEs and bind each PE interface connected to a CE to a VPN instance.
4. Because the P device does not support MPLS functions, an LSP cannot be used to transmit VPN packets. Configure a tunnel policy on the PEs to specify that VPN packets are transmitted over a GRE tunnel, and apply the tunnel policy.
5. Establish EBGP peer relationships between PEs and CEs to exchange routes so that a CE can learn routes from the peer CE and CE1 can communicate with CE2.

Procedure

Step 1 Configure an IP address for each interface.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
```

```
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24  
[CE1-GigabitEthernet1/0/0] quit
```

Configure IP addresses for interfaces on PE1 except the interface to be bound to a VPN instance. This is because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view  
[Huawei] sysname PE1  
[PE1] interface gigabitethernet 2/0/0  
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 24  
[PE1-GigabitEthernet2/0/0] quit  
[PE1] interface loopback 1  
[PE1-LoopBack1] ip address 10.10.1.1 32  
[PE1-LoopBack1] quit
```

Configure the P device.

```
<Huawei> system-view  
[Huawei] sysname P  
[P] interface gigabitethernet 1/0/0  
[P-GigabitEthernet1/0/0] ip address 172.1.1.2 24  
[P-GigabitEthernet1/0/0] quit  
[P] interface gigabitethernet 2/0/0  
[P-GigabitEthernet2/0/0] ip address 172.2.1.1 24  
[P-GigabitEthernet2/0/0] quit
```

Configure IP addresses for interfaces on PE2 except the interface to be bound to a VPN instance. This is because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view  
[Huawei] sysname PE2  
[PE2] interface gigabitethernet 2/0/0  
[PE2-GigabitEthernet2/0/0] ip address 172.2.1.2 24  
[PE2-GigabitEthernet2/0/0] quit  
[PE2] interface loopback 1  
[PE2-LoopBack1] ip address 10.10.2.1 32  
[PE2-LoopBack1] quit
```

Configure CE2.

```
<Huawei> system-view  
[Huawei] sysname CE2  
[CE2] interface gigabitethernet 1/0/0  
[CE2-GigabitEthernet1/0/0] ip address 10.2.1.1 24  
[CE2-GigabitEthernet1/0/0] quit
```

Step 2 Configure IGP on the MPLS backbone network to implement interworking between PEs.

Configure PE1.

```
[PE1] ospf 1  
[PE1-ospf-1] area 0  
[PE1-ospf-1-area-0.0.0.0] network 10.10.1.1 0.0.0.0  
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255  
[PE1-ospf-1-area-0.0.0.0] quit  
[PE1-ospf-1] quit
```

Configure the P device.

```
[P] ospf 1  
[P-ospf-1] area 0  
[P-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255  
[P-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255  
[P-ospf-1-area-0.0.0.0] quit  
[P-ospf-1] quit
```

Configure PE2.

```
[PE2] ospf 1
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 10.10.2.1 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

After the configurations are complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

Step 3 Configure a GRE tunnel.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] source loopback 1
[PE1-Tunnel0/0/1] destination 10.10.2.1
[PE1-Tunnel0/0/1] ip address 10.3.1.1 24
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] tunnel-protocol gre
[PE2-Tunnel0/0/1] source loopback 1
[PE2-Tunnel0/0/1] destination 10.10.1.1
[PE2-Tunnel0/0/1] ip address 10.3.1.2 24
[PE2-Tunnel0/0/1] quit
```

Step 4 Enable basic MPLS functions on the PEs.

Configure PE1.

```
[PE1] mpls lsr-id 10.10.1.1
[PE1] mpls
[PE1-mpls] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 10.10.2.1
[PE2] mpls
[PE2-mpls] quit
```

Step 5 Configure VPN instances on PEs and bind each interface that connects a PE to a CE to a VPN instance. Apply tunnel policies on the PEs to specify the GRE tunnel used to forward VPN packets.

Configure PE1.

```
[PE1] tunnel-policy gre1
[PE1-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE1-tunnel-policy-gre1] quit
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE1-vpn-instance-vpn1-af-ipv4] tnl-policy gre1
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] tunnel-policy gre1
[PE2-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE2-tunnel-policy-gre1] quit
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE2-vpn-instance-vpn1-af-ipv4] tnl-policy gre1
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet1/0/0] ip address 10.2.1.2 24
[PE2-GigabitEthernet1/0/0] quit
```

After the configurations are complete, run the **display ip vpn-instance verbose** command on PEs to view the configurations of VPN instances. Each PE can ping its local CE.

NOTE

If a PE has multiple interfaces bound to the same VPN instance, specify a source IP address by setting **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping a remote CE. If the source IP address is not specified, the ping operation fails.

Step 6 Set up EBGP peer relationships between the PEs and CEs and import VPN routes to EBGP.

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65420
[CE2-bgp] peer 10.2.1.2 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 as-number 65420
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

After the configurations are complete, run the **display bgp vpnv4 vpn-instance peer** command on PEs. You can see that BGP peer relationships have been established between PEs and CEs and are in **Established** state.

The command output on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 10.10.1.1
Local AS number : 100
```

```

VPN-Instance vpn1, Router ID 10.10.1.1:
Total number of peers : 1                Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.1.1.1     4          65410    6         3       0 00:01:14
Established    3
  
```

Step 7 Set up an MP-IBGP peer relationship between PEs.

Configure PE1.

```

[PE1] bgp 100
[PE1-bgp] peer 10.10.2.1 as-number 100
[PE1-bgp] peer 10.10.2.1 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 10.10.2.1 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
  
```

Configure PE2.

```

[PE2] bgp 100
[PE2-bgp] peer 10.10.1.1 as-number 100
[PE2-bgp] peer 10.10.1.1 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.10.1.1 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
  
```

After the configurations are complete, run the **display bgp vpnv4 all peer** command on a PE. The command output shows that the BGP peer relationships have been established between the PEs and are in the **Established** state.

```

[PE1] display bgp vpnv4 all peer

BGP local router ID : 10.10.1.1
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.10.2.1     4          100     4         7       0 00:02:54
Established    0

  Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 10.10.1.1:
  10.1.1.1     4          65410    122        119       0 01:57:43
Established    3
  
```

Step 8 Verify the configuration.

After the configuration is complete, CEs can learn routes to each other. CEs can successfully ping each other.

The command output on CE1 is used as an example.

```

[CE1] display ip routing-table 10.2.1.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto   Pre  Cost           Flags NextHop           Interface
-----
  10.2.1.0/24      EBGP    255  0              D    10.1.1.2
  
```

```
GigabitEthernet1/0/0

[CE1] ping 10.2.1.1
  PING 10.2.1.1: 56 data bytes, press CTRL_C to break
    Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=253 time=10 ms
    Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=253 time=1 ms

--- 10.2.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/2/10 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:1
tnl-policy gre1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
#
mpls lsr-id 10.10.1.1
mpls
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.1.1.1 255.255.255.0
#
interface LoopBack1
ip address 10.10.1.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.3.1.1 255.255.255.0
tunnel-protocol gre
source LoopBack1
destination 10.10.2.1
#
tunnel-policy gre1
```

```
tunnel select-seq gre load-balance-number 1
#
bgp 100
peer 10.10.2.1 as-number 100
peer 10.10.2.1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 10.10.2.1 enable
#
ipv4-family vpnv4
policy vpn-target
peer 10.10.2.1 enable
#
ipv4-family vpn-instance vpn1
peer 10.1.1.1 as-number 65410
import-route direct
#
ospf 1
area 0.0.0.0
network 10.10.1.1 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

- Configuration file of the P device

```
#
sysname P
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.2.1.1 255.255.255.0
#
ospf 1
area 0.0.0.0
network 172.1.1.0 0.0.0.255
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:2
tnl-policy gre1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
#
mpls lsr-id 10.10.2.1
mpls
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.2.1.2 255.255.255.0
#
interface LoopBack1
ip address 10.10.2.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.3.1.2 255.255.255.0
tunnel-protocol gre
source LoopBack1
```

```
destination 10.10.1.1
#
tunnel-policy gre1
 tunnel select-seq gre load-balance-number 1
#
bgp 100
 peer 10.10.1.1 as-number 100
 peer 10.10.1.1 connect-interface LoopBack1
#
ipv4-family unicast
 undo synchronization
 peer 10.10.1.1 enable
#
ipv4-family vpnv4
 policy vpn-target
 peer 10.10.1.1 enable
#
ipv4-family vpn-instance vpn1
 peer 10.2.1.1 as-number 65420
#
ospf 1
 area 0.0.0.0
 network 10.10.2.1 0.0.0.0
 network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
bgp 65420
 peer 10.2.1.2 as-number 100
#
ipv4-family unicast
 undo synchronization
 import-route direct
 peer 10.2.1.2 enable
#
return
```

3.8.6 Example for Configuring VLL to Use a GRE Tunnel

Networking Requirements

 **NOTE**

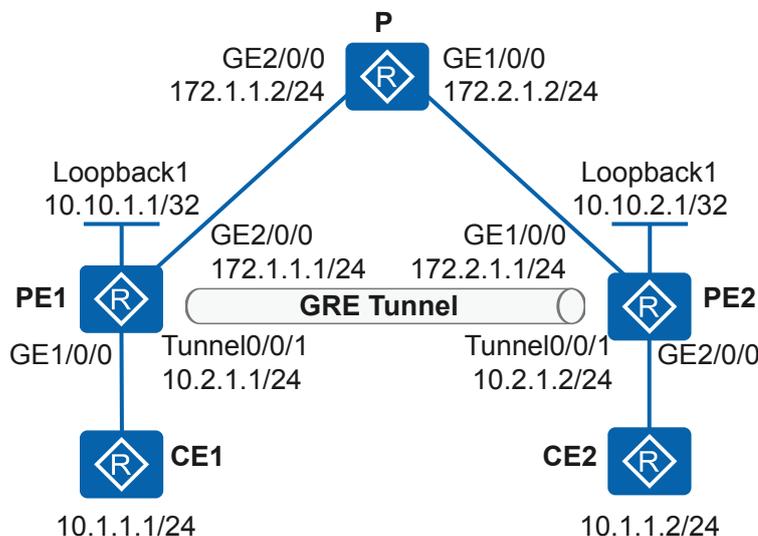
The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot be used in this scenario.

An ISP network provides the L2VPN service for users. Many users connect to the MPLS network through PE1 and PE2, and users on the PEs change frequently. A proper VPN solution is required to provide secure VPN services for users and to simplify configuration when new users connect to the network.

A Martini VLL connection can be set up between CE1 and CE2 to meet these requirements. By default, the system uses Label Switched Paths (LSPs) for Martini VLL, and does not perform load balancing. When the P does not provide MPLS functions, VLL cannot be implemented.

To solve the problem, apply a tunnel policy to Martini VLL to specify that VLL services are transmitted over a GRE tunnel.

Figure 3-22 Networking diagram for configuring VLL to use a GRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on the PE and P devices on the backbone network to ensure reachability between them.
2. Enable MPLS and MPLS LDP on PEs. Set up a remote LDP session between the PEs to exchange VC labels between the PEs.
3. Enable MPLS L2VPN on PEs. Enabling MPLS L2VPN is the prerequisite for VLL configuration.
4. Create GRE tunnel interfaces on PEs and establish a GRE tunnel between PEs.
5. Create VC connections on PEs. Because the P does not support MPLS functions, configure a tunnel policy and apply it when you create VC connections so that VLL services can be transmitted over a GRE tunnel.

Procedure

Step 1 Configure interface IP addresses and a routing protocol on the PEs and P.

Configure PE1. The configurations of PE2 and P are similar to the configuration of PE1, and are not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 10.10.1.1 255.255.255.255
[PE1-LoopBack1] quit
[PE1] ospf 1
```

```
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 10.10.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configurations are complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

Step 2 Configure basic MPLS functions and LDP on PEs and establish a remote LDP session between PEs.

Configure PE1.

```
[PE1] mpls lsr-id 10.10.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] mpls ldp remote-peer 10.10.2.1
[PE1-mpls-ldp-remote-10.10.2.1] remote-ip 10.10.2.1
[PE1-mpls-ldp-remote-10.10.2.1] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 10.10.2.1
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] mpls ldp remote-peer 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] remote-ip 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] quit
```

After the configurations are complete, run the **display mpls ldp session** command on PE1 to view the LDP session status. You can see that an LDP session is set up between PE1 and PE2.

The display on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
10.10.2.1:0      Operational DU   Passive  0000:00:01  1/1
-----
TOTAL: 1 session(s) Found.
```

Step 3 Enable MPLS L2VPN on PEs.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
```

Step 4 Create GRE tunnel interfaces on PEs and establish a GRE tunnel between PEs.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip address 10.2.1.1 255.255.255.0
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] source 10.10.1.1
[PE1-Tunnel0/0/1] destination 10.10.2.1
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] ip address 10.2.1.2 255.255.255.0
[PE2-Tunnel0/0/1] tunnel-protocol gre
[PE2-Tunnel0/0/1] source 10.10.2.1
[PE2-Tunnel0/0/1] destination 10.10.1.1
[PE2-Tunnel0/0/1] quit
```

After the configurations are complete, the tunnel interfaces go Up and can ping each other.

The display on PE1 is used as an example.

```
[PE1] ping -a 10.2.1.1 10.2.1.2
PING 10.2.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.2.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.2.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 10.2.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 10.2.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 10.2.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.2.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

Step 5 Configure a tunnel policy, create VC connections, and apply the policy to the VC connections so that VLL services can be transmitted over a GRE tunnel.

Configure PE1.

```
[PE1] tunnel-policy gre1
[PE1-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE1-tunnel-policy-gre1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 10.10.2.1 39 tunnel-policy gre1
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] tunnel-policy gre1
[PE2-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE2-tunnel-policy-gre1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc 10.10.1.1 39 tunnel-policy gre1
[PE2-GigabitEthernet2/0/0] quit
```

Step 6 Verify the configuration.

After the configurations are complete, check the L2VPN connection on PEs. You can see that an L2VC connection has been set up and is in Up state.

The display on PE1 is used as an example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state          : up
AC status              : up
VC state               : up
Label state            : 0
Token state            : 0
VC ID                  : 39
```

```

VC type           : Ethernet
destination       : 10.10.2.1
local group ID    : 0
local VC label    : 1025
remote group ID   : 0
remote VC label   : 1024
local AC OAM State : up
local PSN OAM State : up
local forwarding state : forwarding
local status code : 0x0
remote AC OAM state : up
remote PSN OAM state : up
remote forwarding state: forwarding
remote status code : 0x0
ignore standby state : no
BFD for PW       : unavailable
VCCV State       : up
manual fault     : not set
active state     : active
forwarding entry : exist
link state       : up
local VC MTU     : 1500
remote VC MTU    : 1500
local VCCV       : alert ttl lsp-ping bfd
remote VCCV      : alert ttl lsp-ping bfd
local control word : disable
remote control word : disable
tunnel policy name : gre1
PW template name  : --
primary or secondary : primary
load balance type : flow
Access-port      : false
Switchover Flag  : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type   : gre , TNL ID : 0x2
  Backup TNL type : lsp , TNL ID : 0x0
create time      : 0 days, 2 hours, 37 minutes, 1 seconds
up time          : 0 days, 0 hours, 2 minutes, 11 seconds
last change time : 0 days, 0 hours, 2 minutes, 11 seconds
VC last up time  : 2013/02/20 18:58:24
VC total up time : 0 days, 2 hours, 35 minutes, 58 seconds
CKey             : 2
NKey             : 1
PW redundancy mode : frr
AdminPw interface : --
AdminPw link state : --
Diffserv Mode    : uniform
Service Class    : --
Color            : --
DomainId         : --
Domain Name      : --
  
```

Run the **display tunnel-info tunnel-id** command on PEs according to the tunnel ID in the preceding command output. You can view details of the specified tunnel ID.

```

[PE1] display tunnel-info tunnel-id 2
Tunnel ID:          0x2
Tunnel Token:       2
Type:               gre
Destination:        10.10.2.1
Out Slot:           0
Instance ID:        0
Interface:          Tunnel0/0/1
  
```

CE1 and CE2 can ping each other successfully.

The display on CE1 is used as an example.

```

[CE1] ping 10.1.1.2
PING 10.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=31 ms
  Reply from 10.1.1.2: bytes=56 Sequence=2 ttl=255 time=10 ms
  Reply from 10.1.1.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.1.1.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  
```

```
Reply from 10.1.1.2: bytes=56 Sequence=5 ttl=255 time=28 ms
--- 10.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/15/31 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 10.10.1.1
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 10.10.2.1
remote-ip 10.10.2.1
#
interface GigabitEthernet1/0/0
mpls l2vc 10.10.2.1 39 tunnel-policy gre1
#
interface GigabitEthernet2/0/0
ip address 172.1.1.1 255.255.255.0
#
interface LoopBack1
ip address 10.10.1.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.2.1.1 255.255.255.0
tunnel-protocol gre
source 10.10.1.1
destination 10.10.2.1
#
ospf 1
area 0.0.0.0
network 10.10.1.1 0.0.0.0
network 172.1.1.0 0.0.0.255
#
tunnel-policy gre1
tunnel select-seq gre load-balance-number 1
#
return
```

- Configuration file of P

```
#
sysname P
#
interface GigabitEthernet2/0/0
ip address 172.1.1.2 255.255.255.0
#
interface GigabitEthernet1/0/0
ip address 172.2.1.2 255.255.255.0
```

```
#
ospf 1
 area 0.0.0.0
  network 172.1.1.0 0.0.0.255
  network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
 sysname PE2
#
 mpls lsr-id 10.10.2.1
 mpls
#
 mpls l2vpn
#
 mpls ldp
#
 mpls ldp remote-peer 10.10.1.1
 remote-ip 10.10.1.1
#
 interface GigabitEthernet1/0/0
 ip address 172.2.1.1 255.255.255.0
#
 interface GigabitEthernet2/0/0
 mpls l2vc 10.10.1.1 39 tunnel-policy gre1
#
 interface LoopBack1
 ip address 10.10.2.1 255.255.255.255
#
 interface Tunnel0/0/1
 ip address 10.2.1.2 255.255.255.0
 tunnel-protocol gre
 source 10.10.2.1
 destination 10.10.1.1
#
 ospf 1
 area 0.0.0.0
  network 10.10.2.1 0.0.0.0
  network 172.2.1.0 0.0.0.255
#
 tunnel-policy gre1
 tunnel select-seq gre load-balance-number 1
#
return
```

- Configuration file of CE2

```
#
 sysname CE2
#
 interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
#
return
```

3.8.7 Example for Connecting a CE to a VPN Through a GRE Tunnel over a Public Network

Networking Requirements

In [Figure 3-23](#):

- PE1 and PE2 reside on the MPLS backbone network.
- R1 connects CE1 and PE1 over the public network.

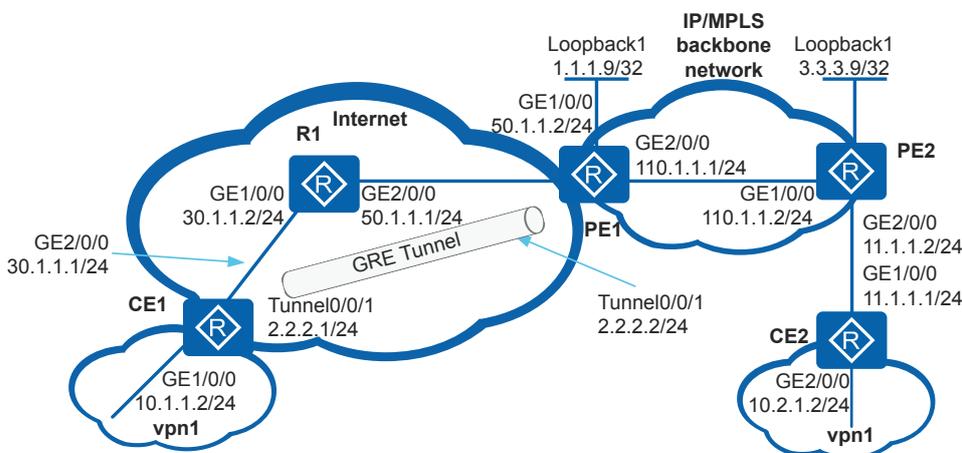
- CE2 is directly connected to PE2.
- CE1 and CE2 reside on the same VPN and are reachable to each other.

PE1 is indirectly connected to CE1. Therefore, no VPN instance can be bound to the physical interface of PE1. A GRE tunnel is set up between CE1 and PE1 and this tunnel traverses the public network. On PE1, bind the GRE tunnel to a VPN to connect CE1 to the VPN using the GRE tunnel.

NOTE

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot work on an MPLS backbone network.

Figure 3-23 Connecting a CE to a VPN through a GRE tunnel over a public network



Configuration Roadmap

The configuration roadmap is as follows:

1. Run OSPF process 10 on PE1 and PE2 to implement interworking between them, and enable MPLS.
2. Run OSPF process 20 on CE1, R1, and PE1 to implement interworking among them.
3. Set up a GRE tunnel between CE1 and PE1.
4. Create **vpn1** on PE1 and PE2. On PE1, bind **vpn1** to the GRE tunnel interface. On PE2, bind **vpn1** to the physical interface connected to CE2.
5. Configure IS-IS on CE1 and PE1 to calculate routes between CE2 and PE2 and their connected PEs.
6. Run BGP on the PEs to implement interworking between CE1 and CE2.

Procedure

Step 1 Configure an IP address for each interface.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] interface gigabitethernet 2/0/0
[CE1-GigabitEthernet2/0/0] ip address 30.1.1.1 24
[CE1-GigabitEthernet2/0/0] quit
```

Configure R1.

```
<Huawei> system-view
[Huawei] sysname R1
[R1] interface gigabitethernet 1/0/0
[R1-GigabitEthernet1/0/0] ip address 30.1.1.2 24
[R1-GigabitEthernet1/0/0] quit
[R1] interface gigabitethernet 2/0/0
[R1-GigabitEthernet2/0/0] ip address 50.1.1.1 24
[R1-GigabitEthernet2/0/0] quit
```

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 50.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 110.1.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
```

Configure IP addresses for interfaces on PE2 except the interface to be bound to a VPN instance, because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip address 110.1.1.2 24
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface loopback 1
[PE2-LoopBack1] ip address 3.3.3.9 32
[PE2-LoopBack1] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 11.1.1.1 24
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] ip address 10.2.1.2 24
[CE2-GigabitEthernet2/0/0] quit
```

Step 2 Configure routes between the PEs and enable MPLS.

On PE1, enable MPLS LDP, and run OSPF process 10 to configure reachable routes between the PEs. LSPs are set up automatically.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] lsp-trigger all
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
```

```
[PE1] ospf 10
[PE1-ospf-10] area 0
[PE1-ospf-10-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-10-area-0.0.0.0] network 110.1.1.0 0.0.0.255
[PE1-ospf-10-area-0.0.0.0] quit
[PE1-ospf-10] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

On PE2, enable MPLS LDP, and run OSPF process 10 to configure reachable routes between the PEs. LSPs are set up automatically.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] lsp-trigger all
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] ospf 10
[PE2-ospf-10] area 0
[PE2-ospf-10-area-0.0.0.0] network 3.3.3.9 0.0.0.0
[PE2-ospf-10-area-0.0.0.0] network 110.1.1.0 0.0.0.255
[PE2-ospf-10-area-0.0.0.0] quit
[PE2-ospf-10] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

Step 3 Create a VPN instance **vpn1** on PE1 and bind **vpn1** to the GRE tunnel.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 export-extcommunity
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 import-extcommunity
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip binding vpn-instance vpn1
[PE1-Tunnel0/0/1] ip address 2.2.2.2 255.255.255.0
[PE1-Tunnel0/0/1] quit
```

Step 4 Create a VPN instance **vpn1** on PE2 and bind **vpn1** to a user-side interface.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] route-distinguisher 200:1
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 export-extcommunity
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 import-extcommunity
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet2/0/0] ip address 11.1.1.2 255.255.255.0
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Configure tunnel interfaces of the GRE tunnel.

Configure CE1.

```
[CE1] interface tunnel 0/0/1
[CE1-Tunnel0/0/1] tunnel-protocol gre
[CE1-Tunnel0/0/1] source 30.1.1.1
[CE1-Tunnel0/0/1] destination 50.1.1.2
[CE1-Tunnel0/0/1] ip address 2.2.2.1 24
[CE1-Tunnel0/0/1] quit
```

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] tunnel-protocol gre
```

```
[PE1-Tunnel0/0/1] source 50.1.1.2
[PE1-Tunnel0/0/1] destination 30.1.1.1
[PE1-Tunnel0/0/1] quit
```

Step 6 Configure OSPF on CE1, R1, and PE1.

Configure CE1.

```
[CE1] ospf 20
[CE1-ospf-20] area 0
[CE1-ospf-20-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[CE1-ospf-20-area-0.0.0.0] quit
[CE1-ospf-20] quit
```

Configure R1.

```
[R1] ospf 20
[R1-ospf-20] area 0
[R1-ospf-20-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[R1-ospf-20-area-0.0.0.0] network 50.1.1.0 0.0.0.255
[R1-ospf-20-area-0.0.0.0] quit
[R1-ospf-20] quit
```

Configure PE1.

```
[PE1] ospf 20
[PE1-ospf-20] area 0
[PE1-ospf-20-area-0.0.0.0] network 50.1.1.0 0.0.0.255
[PE1-ospf-20-area-0.0.0.0] quit
[PE1-ospf-20] quit
```

Step 7 Configure IS-IS on CE1 and PE1 to calculate routes between them.

Configure CE1.

```
[CE1] isis 50
[CE1-isis-50] network-entity 50.0000.0000.0001.00
[CE1-isis-50] quit
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] isis enable 50
[CE1-GigabitEthernet1/0/0] quit
[CE1] interface tunnel 0/0/1
[CE1-Tunnel0/0/1] isis enable 50
[CE1-Tunnel0/0/1] quit
```

Configure PE1.

```
[PE1] isis 50 vpn-instance vpn1
[PE1-isis-50] network-entity 50.0000.0000.0002.00
[PE1-isis-50] quit
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] isis enable 50
[PE1-Tunnel0/0/1] quit
```

Step 8 Configure IS-IS on CE2 and PE2 to calculate routes between them.

Configure CE2.

```
[CE2] isis 50
[CE2-isis-50] network-entity 50.0000.0000.0004.00
[CE2-isis-50] quit
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] isis enable 50
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] isis enable 50
[CE2-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] isis 50 vpn-instance vpn1
[PE2-isis-50] network-entity 50.0000.0000.0003.00
[PE2-isis-50] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] isis enable 50
[PE2-GigabitEthernet2/0/0] quit
```

Step 9 Set up an MP-IBGP peer relationship between the PEs.

On PE1, configure an IBGP peer relationship with PE2 using a loopback interface to exchange VPN IPv4 route information.

```
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
```

Import IS-IS routes to vpn1.

```
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route isis 50
```

On PE2, configure an IBGP peer relationship with PE1 using a loopback interface to exchange VPN IPv4 route information.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
```

Import IS-IS routes to vpn1.

```
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] import-route isis 50
```

Step 10 Import BGP routes to the IS-IS routing table.

Configure PE1.

```
[PE1] isis 50
[PE1-isis-50] import-route bgp
```

Configure PE2.

```
[PE2] isis 50
[PE2-isis-50] import-route bgp
```

Step 11 Verify the configuration.

After the configuration is complete, CE1 and CE2 have reachable routes to each other. The command output on CE1 is used as an example.

```
<CE1> display ip routing-table 41.1.1.0
<keyword conref=" ../commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
```

```
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto   Pre  Cost           Flags NextHop           Interface
-----
41.1.1.0/24        ISIS-L2 15   74             D    2.2.2.2            Tunnel0/0/1
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
isis 50
network-entity 50.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
ip address 10.1.1.2 255.255.255.0
isis enable 50
#
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
#
interface Tunnel0/0/1
ip address 2.2.2.1 255.255.255.0
tunnel-protocol gre
source 30.1.1.1
destination 50.1.1.2
isis enable 50
#
ospf 20
area 0.0.0.0
network 30.1.1.0 0.0.0.255
#
return
```

- Configurations file of R1

```
#
sysname R1
#
interface GigabitEthernet1/0/0
ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 50.1.1.1 255.255.255.0
#
ospf 20
area 0.0.0.0
network 30.1.1.0 0.0.0.255
network 50.1.1.0 0.0.0.255
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
ip vpn-instance vpn1
route-distinguisher 100:1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
lsp-trigger all
#
mpls ldp
#
isis 50 vpn-instance vpn1
network-entity 50.0000.0000.0002.00
import-route bgp
#
interface GigabitEthernet1/0/0
ip address 50.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
```

```
ip address 110.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
interface Tunnel0/0/1
ip binding vpn-instance vpn1
ip address 2.2.2.2 255.255.255.0
tunnel-protocol gre
source 50.1.1.2
destination 30.1.1.1
isis enable 50
#
bgp 100
peer 3.3.3.9 as-number 100
peer 3.3.3.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 3.3.3.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 3.3.3.9 enable
#
ipv4-family vpn-instance vpn1
import-route isis 50
#
ospf 10
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 110.1.1.0 0.0.0.255
#
ospf 20
area 0.0.0.0
network 50.1.1.0 0.0.0.255
#
return
```

● Configuration file of PE2

```
#
sysname PE2
#
ip vpn-instance vpn1
route-distinguisher 200:1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
lsp-trigger all
#
mpls ldp
#
isis 50 vpn-instance vpn1
network-entity 50.0000.0000.0003.00
import-route bgp
#
interface GigabitEthernet1/0/0
ip address 110.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 11.1.1.2 255.255.255.0
isis enable 50
#
```

```
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.9 enable
#
 ipv4-family vpn-instance vpn1
  import-route isis 50
#
ospf 10
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 110.1.1.0 0.0.0.255
#
return
```

● Configuration file of CE2

```
#
 sysname CE2
#
 isis 50
  network-entity 50.0000.0000.0004.00
#
 interface GigabitEthernet1/0/0
  ip address 11.1.1.1 255.255.255.0
  isis enable 50
#
 interface GigabitEthernet2/0/0
  ip address 10.2.1.2 255.255.255.0
  isis enable 50
#
return
```

3.8.8 Example for Connecting a CE to a VPN Through a GRE Tunnel over a VPN

Networking Requirements

In [Figure 3-24](#):

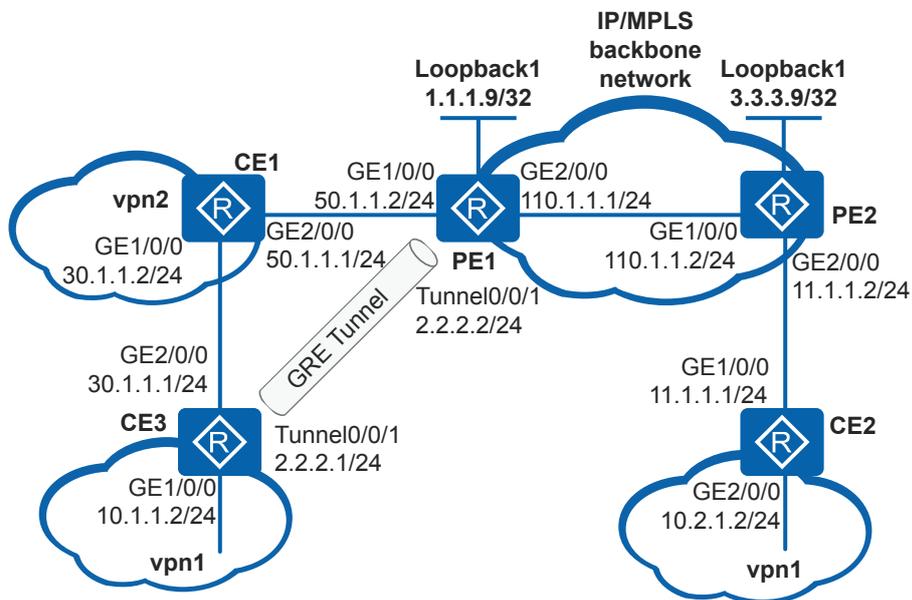
- PE1 and PE2 reside on a class 1 carrier's MPLS backbone network.
- The VPN instance **vpn2** belongs to a class 2 carrier's network, and CE1 is directly connected to PE1.
- CE2 and CE3 connect to user hosts. CE2 is directly connected to PE2, and CE3 is directly connected to CE1. CE2 and CE3 belong to **vpn1** and are reachable to each other.

PE1 is indirectly connected to CE3. Therefore, no VPN instance can be bound to the physical interface of PE1. A GRE tunnel is set up between CE3 and PE1 and this tunnel traverses **vpn2**. On PE1, bind the GRE tunnel to **vpn1** to connect CE3 to vpn1 using the GRE tunnel.

 **NOTE**

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot work on an MPLS backbone network.

Figure 3-24 Connecting a CE to a VPN through a GRE Tunnel over a VPN



Configuration Roadmap

The configuration roadmap is as follows:

1. Run OSPF process 10 on PE1 and PE2 to implement interworking between them, and enable MPLS.
2. Configure a VPN instance **vpn2** on PE1, and run OSPF process 20 on PE1, CE1, and CE3 to implement interworking among the three devices.
3. Set up a GRE tunnel between CE3 and PE1. CE3 is connected to PE1 over **vpn2**, and the interface on PE1 directly connected to CE1 is bound to **vpn2**. Therefore, the interfaces directly connecting CE3 to CE1 and directly connecting PE1 to CE1 belong to **vpn2**. When configuring a GRE tunnel between PE1 and CE3, you need to set a tunnel destination address that belongs to **vpn2**.
4. Create **vpn1** on PE1 and PE2. On PE1, bind **vpn1** to the GRE tunnel interface. On PE2, bind **vpn1** to the physical interface connected to CE2.
5. Run IS-IS on the devices to dynamically calculate routes between the CEs and PEs.
6. Run BGP on the PEs to implement interworking between CE2 and CE3.

Procedure

Step 1 Configure an IP address for each interface.

Configure CE3.

```
<Huawei> system-view
[Huawei] sysname CE3
[CE3] interface gigabitethernet 1/0/0
[CE3-GigabitEthernet1/0/0] ip address 10.1.1.2 24
```

```
[CE3-GigabitEthernet1/0/0] quit
[CE3] interface gigabitethernet 2/0/0
[CE3-GigabitEthernet2/0/0] ip address 30.1.1.1 24
[CE3-GigabitEthernet2/0/0] quit
```

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 30.1.1.2 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] interface gigabitethernet 2/0/0
[CE1-GigabitEthernet2/0/0] ip address 50.1.1.1 24
[CE1-GigabitEthernet2/0/0] quit
```

Configure IP addresses for interfaces on PE1 except the interface to be bound to a VPN instance, because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 110.1.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
```

Configure IP addresses for interfaces on PE2 except the interface to be bound to a VPN instance, because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip address 110.1.1.2 24
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface loopback 1
[PE2-LoopBack1] ip address 3.3.3.9 32
[PE2-LoopBack1] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 11.1.1.1 24
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] ip address 10.2.1.2 24
[CE2-GigabitEthernet2/0/0] quit
```

Step 2 Configure routes between the PEs and enable MPLS.

On PE1, enable MPLS LDP, and run OSPF process 10 to configure reachable routes between the PEs. LSPs are set up automatically.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] lsp-trigger all
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] ospf 10
[PE1-ospf-10] area 0
[PE1-ospf-10-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-10-area-0.0.0.0] network 110.1.1.0 0.0.0.255
```

```
[PE1-ospf-10-area-0.0.0.0] quit
[PE1-ospf-10] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

On PE2, enable MPLS LDP, and run OSPF process 10 to configure reachable routes between the PEs. LSPs are set up automatically.

```
[PE2] mpls lsr-id 3.3.3.9 32
[PE2] mpls
[PE2-mppls] lsp-trigger all
[PE2-mppls] quit
[PE2] mpls ldp
[PE2-mppls-ldp] quit
[PE2] ospf 10
[PE2-ospf-10] area 0
[PE2-ospf-10-area-0.0.0.0] network 3.3.3.9 0.0.0.0
[PE2-ospf-10-area-0.0.0.0] network 110.1.1.0 0.0.0.255
[PE2-ospf-10-area-0.0.0.0] quit
[PE2-ospf-10] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

Step 3 Create a VPN instance **vpn2** on PE1 and bind **vpn2** to an interface on a class 2 carrier's network.

```
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] route-distinguisher 100:2
[PE1-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 export-extcommunity
[PE1-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 import-extcommunity
[PE1-vpn-instance-vpn2-af-ipv4] quit
[PE1-vpn-instance-vpn2] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn2
[PE1-GigabitEthernet1/0/0] ip address 50.1.1.2 255.255.255.0
[PE1-GigabitEthernet1/0/0] quit
```

Step 4 Create a VPN instance **vpn1** on PE1 and bind **vpn1** to the GRE tunnel.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 export-extcommunity
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 import-extcommunity
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip binding vpn-instance vpn1
[PE1-Tunnel0/0/1] ip address 2.2.2.2 255.255.255.0
[PE1-Tunnel0/0/1] quit
```

Step 5 Create a VPN instance **vpn1** on PE2 and bind **vpn1** to a user-side interface.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] route-distinguisher 200:1
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 export-extcommunity
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 import-extcommunity
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet2/0/0] ip address 11.1.1.2 255.255.255.0
[PE2-GigabitEthernet2/0/0] quit
```

Step 6 Configure tunnel interfaces of the GRE tunnel.

Configure CE3.

```
[CE3] interface tunnel 0/0/1
[CE3-Tunnel0/0/1] tunnel-protocol gre
[CE3-Tunnel0/0/1] source 30.1.1.1
[CE3-Tunnel0/0/1] destination 50.1.1.2
[CE3-Tunnel0/0/1] ip address 2.2.2.1 24
[CE3-Tunnel0/0/1] quit
```

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] source 50.1.1.2
[PE1-Tunnel0/0/1] destination vpn-instance vpn2 30.1.1.1
[PE1-Tunnel0/0/1] quit
```

Step 7 Configure routing protocols on CE3, CE1, and PE1.

Configure CE3.

```
[CE3] ospf 20
[CE3-ospf-20] area 0
[CE3-ospf-20-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[CE3-ospf-20-area-0.0.0.0] quit
[CE3-ospf-20] quit
```

Configure CE1.

```
[CE1] ospf 20
[CE1-ospf-20] area 0
[CE1-ospf-20-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[CE1-ospf-20-area-0.0.0.0] network 50.1.1.0 0.0.0.255
[CE1-ospf-20-area-0.0.0.0] quit
[CE1-ospf-20] quit
```

Configure PE1.

```
[PE1] ospf 20 vpn-instance vpn2
[PE1-ospf-20] area 0
[PE1-ospf-20-area-0.0.0.0] network 50.1.1.0 0.0.0.255
[PE1-ospf-20-area-0.0.0.0] quit
[PE1-ospf-20] quit
```

Step 8 Configure IS-IS on CE3 and PE1 to calculate routes between them.

Configure CE3.

```
[CE3] isis 50
[CE3-isis-50] network-entity 50.0000.0000.0001.00
[CE3-isis-50] quit
[CE3] interface gigabitethernet 1/0/0
[CE3-GigabitEthernet1/0/0] isis enable 50
[CE3-GigabitEthernet1/0/0] quit
[CE3] interface tunnel 0/0/1
[CE3-Tunnel0/0/1] isis enable 50
[CE3-Tunnel0/0/1] quit
```

Configure PE1.

```
[PE1] isis 50 vpn-instance vpn1
[PE1-isis-50] network-entity 50.0000.0000.0002.00
[PE1-isis-50] quit
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] isis enable 50
[PE1-Tunnel0/0/1] quit
```

Step 9 Configure IS-IS on CE2 and PE2 to calculate routes between them.

Configure CE2.

```
[CE2] isis 50
[CE2-isis-50] network-entity 50.0000.0000.0004.00
```

```
[CE2-isis-50] quit
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] isis enable 50
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] isis enable 50
[CE2-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] isis 50 vpn-instance vpn1
[PE2-isis-50] network-entity 50.0000.0000.0003.00
[PE2-isis-50] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] isis enable 50
[PE2-GigabitEthernet2/0/0] quit
```

Step 10 Set up an MP-IBGP peer relationship between the PEs.

On PE1, configure an IBGP peer relationship with PE2 using a loopback interface to exchange VPN IPv4 route information.

```
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
```

Import IS-IS routes to vpn1.

```
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route isis 50
```

On PE2, configure an IBGP peer relationship with PE1 using a loopback interface to exchange VPN IPv4 route information.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
```

Import IS-IS routes to vpn1.

```
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] import-route isis 50
```

Step 11 Import BGP routes to the IS-IS routing table.

Configure PE1.

```
[PE1] isis 50
[PE1-isis-50] import-route bgp
```

Configure PE2.

```
[PE2] isis 50
[PE2-isis-50] import-route bgp
```

Step 12 Verify the configuration.

After the configuration is complete, CE1 and CE2 have reachable routes to each other. The command output on CE1 is used as an example.

```
<CE1> display ip routing-table 41.1.1.0
Route Flags: R - relay,
D - download to fib
```

```
-----  
Routing Table : Public  
Summary Count : 1  
Destination/Mask    Proto    Pre    Cost    Flags NextHop    Interface  
41.1.1.0/24        ISIS-L2  15     74      D     2.2.2.2    Tunnel0/0/1
```

----End

Configuration Files

- Configuration file of CE3

```
#  
sysname CE3  
#  
isis 50  
network-entity 50.0000.0000.0001.00  
#  
interface GigabitEthernet1/0/0  
ip address 10.1.1.2 255.255.255.0  
isis enable 50  
#  
interface GigabitEthernet2/0/0  
ip address 30.1.1.1 255.255.255.0  
#  
interface Tunnel0/0/1  
ip address 2.2.2.1 255.255.255.0  
tunnel-protocol gre  
source 30.1.1.1  
destination 50.1.1.2  
isis enable 50  
#  
ospf 20  
area 0.0.0.0  
network 30.1.1.0 0.0.0.255  
#  
return
```

- Configuration file of CE1

```
#  
sysname CE1  
#  
interface GigabitEthernet1/0/0  
ip address 30.1.1.2 255.255.255.0  
#  
interface GigabitEthernet2/0/0  
ip address 50.1.1.1 255.255.255.0  
#  
ospf 20  
area 0.0.0.0  
network 30.1.1.0 0.0.0.255  
network 50.1.1.0 0.0.0.255  
#  
return
```

- Configuration file of PE1

```
#  
sysname PE1  
#  
ip vpn-instance vpn1  
route-distinguisher 100:1  
vpn-target 111:1 export-extcommunity  
vpn-target 111:1 import-extcommunity  
#  
ip vpn-instance vpn2  
route-distinguisher 100:2  
vpn-target 222:2 export-extcommunity  
vpn-target 222:2 import-extcommunity
```

```
#
mpls lsr-id 1.1.1.9
mpls
  lsp-trigger all
#
mpls ldp
#
isis 50 vpn-instance vpn1
  network-entity 50.0000.0000.0002.00
  import-route bgp
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn2
  ip address 50.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip address 110.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
interface Tunnel0/0/1
  ip binding vpn-instance vpn1
  ip address 2.2.2.2 255.255.255.0
  tunnel-protocol gre
  source 50.1.1.2
  destination vpn-instance vpn2 30.1.1.1
  isis enable 50
#
bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpn1
    import-route isis 50
#
ospf 10
  area 0.0.0.0
  network 1.1.1.9 0.0.0.0
  network 110.1.1.0 0.0.0.255
#
ospf 20 vpn-instance vpn2
  area 0.0.0.0
  network 50.1.1.0 0.0.0.255
#
return
```

● Configuration file of PE2

```
#
sysname PE2
#
ip vpn-instance vpn1
  route-distinguisher 200:1
  vpn-target 111:1 export-extcommunity
  vpn-target 111:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
  lsp-trigger all
#
```

```
mpls ldp
#
isis 50 vpn-instance vpn1
network-entity 50.0000.0000.0003.00
import-route bgp
#
interface GigabitEthernet1/0/0
ip address 110.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 11.1.1.2 255.255.255.0
isis enable 50
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpn1
import-route isis 50
#
ospf 10
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 110.1.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

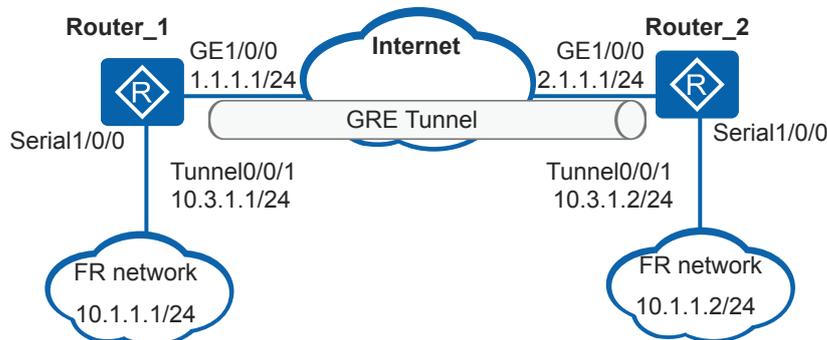
```
#
sysname CE2
#
isis 50
network-entity 50.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
ip address 11.1.1.1 255.255.255.0
isis enable 50
#
interface GigabitEthernet2/0/0
ip address 10.2.1.2 255.255.255.0
isis enable 50
#
return
```

3.8.9 Example for Configuring GRE to Implement Communication Between FR Networks

Networking Requirements

As shown in [Figure 3-25](#), Router_1 and Router_2 use the OSPF protocol to implement communication over the public network. The customer requires that FR networks can communicate over the public network.

Figure 3-25 Configuring GRE to implement communication between FR networks



Configuration Roadmap

To implement communication between FR networks, you need to directly connect Router_1 and Router_2 using a GRE tunnel and configure the link bridge function.

1. Configure an IGP (OSPF process 1 in this example) to implement interworking between the devices.
2. Set up a GRE tunnel between the routers and configure the link bridge function, so that traffic from FR networks can be transmitted over the GRE tunnel.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure Router_1.

```
<Huawei> system-view
[Huawei] sysname Router_1
[Router_1] interface gigabitethernet 1/0/0
[Router_1-GigabitEthernet1/0/0] ip address 1.1.1.1 255.255.255.0
[Router_1-GigabitEthernet1/0/0] quit
[Router_1] interface serial 1/0/0
[Router_1-Serial1/0/0] link-protocol fr
[Router_1-Serial1/0/0] fr dlci 200
[Router_1-fr-dlci-Serial1/0/0-200] quit
[Router_1-Serial1/0/0] quit
```

Configure Router_2.

```
<Huawei> system-view
[Huawei] sysname Router_2
[Router_2] interface gigabitethernet 1/0/0
[Router_2-GigabitEthernet1/0/0] ip address 2.1.1.1 255.255.255.0
[Router_2-GigabitEthernet1/0/0] quit
[Router_2] interface serial 1/0/0
[Router_2-Serial1/0/0] link-protocol fr
[Router_2-Serial1/0/0] fr dlci 200
[Router_2-fr-dlci-Serial1/0/0-200] quit
[Router_2-Serial1/0/0] quit
```

Step 2 Configure OSPF to ensure reachable routes between the routers over the public network.

Configure Router_1.

```
[Router_1] ospf 1
[Router_1-ospf-1] area 0
[Router_1-ospf-1-area-0.0.0.0] network 1.1.1.0 0.0.0.255
[Router_1-ospf-1-area-0.0.0.0] quit
[Router_1-ospf-1] quit
```

Configure Router_2.

```
[Router_2] ospf 1
[Router_2-ospf-1] area 0
[Router_2-ospf-1-area-0.0.0.0] network 2.1.1.0 0.0.0.255
[Router_2-ospf-1-area-0.0.0.0] quit
[Router_2-ospf-1] quit
```

Step 3 Configure tunnel interfaces.

Configure Router_1.

```
[Router_1] interface tunnel 0/0/1
[Router_1-Tunnel0/0/1] tunnel-protocol gre
[Router_1-Tunnel0/0/1] ip address 10.3.1.1 255.255.255.0
[Router_1-Tunnel0/0/1] source 1.1.1.1
[Router_1-Tunnel0/0/1] destination 2.1.1.1
[Router_1-Tunnel0/0/1] quit
```

Configure Router_2.

```
[Router_2] interface tunnel 0/0/1
[Router_2-Tunnel0/0/1] tunnel-protocol gre
[Router_2-Tunnel0/0/1] ip address 10.3.1.2 255.255.255.0
[Router_2-Tunnel0/0/1] source 2.1.1.1
[Router_2-Tunnel0/0/1] destination 1.1.1.1
[Router_2-Tunnel0/0/1] quit
```

After the configuration is complete, the tunnel interfaces turn Up and can ping each other.

Step 4 Configure the link bridge function.

Configure Router_1.

```
[Router_1] link-bridge 2 interface serial 1/0/0 out-interface tunnel 0/0/1
```

Configure Router_2.

```
[Router_2] link-bridge 2 interface serial 1/0/0 out-interface tunnel 0/0/1
```

Step 5 Verify the configuration.

FR networks can communicate over the public network.

----End

Configuration Files

- Configuration file of Router_1

```
#
sysname Router_1
#
link-bridge 2 interface Serial1/0/0 out-interface Tunnel0/0/1
#
interface Serial1/0/0
link-protocol fr
fr dlci 200
#
interface GigabitEthernet1/0/0
ip address 1.1.1.1 255.255.255.0
```

```
#
interface Tunnel0/0/1
 ip address 10.3.1.1 255.255.255.0
 tunnel-protocol gre
 source 1.1.1.1
 destination 2.1.1.1
#
ospf 1
 area 0.0.0.0
  network 1.1.1.0 0.0.0.255
#
return
```

- Configuration file of Router_2

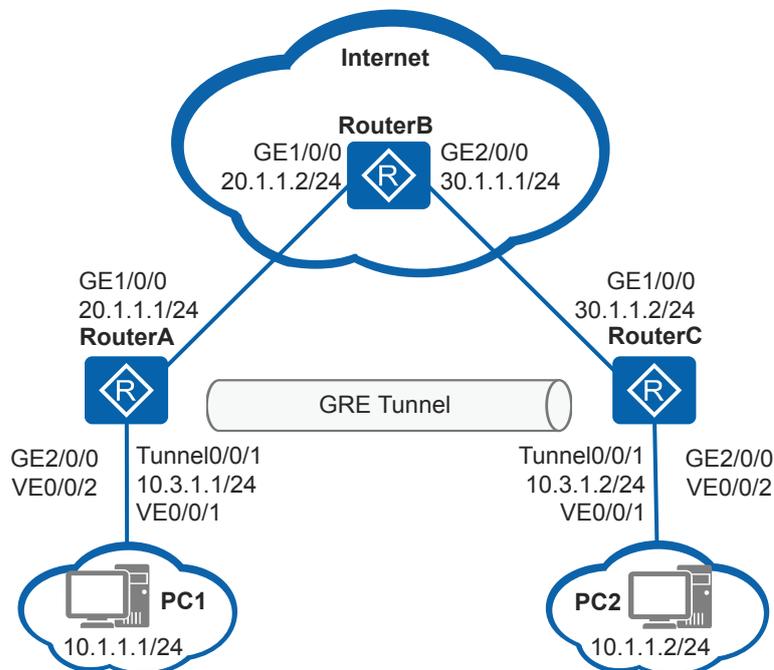
```
#
sysname Router_2
#
link-bridge 2 interface Serial1/0/0 out-interface Tunnel0/0/1
#
interface Serial1/0/0
 link-protocol fr
 fr dlci 200
#
interface GigabitEthernet1/0/0
 ip address 2.1.1.1 255.255.255.0
#
interface Tunnel0/0/1
 ip address 10.3.1.2 255.255.255.0
 tunnel-protocol gre
 source 2.1.1.1
 destination 1.1.1.1
#
ospf 1
 area 0.0.0.0
  network 2.1.1.0 0.0.0.255
#
return
```

3.8.10 Example for Configuring an Ethernet over GRE Tunnel

Networking Requirements

In [Figure 3-26](#), RouterA, RouterB, and RouterC use the Open Shortest Path First (OSPF) protocol to implement communication over the public network. PC1 and PC2 on the branch Ethernet networks belong to the same network segment, and need to communicate over the public network.

Figure 3-26 Ethernet over GRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Run OSPF on all the routers to implement reachable routes among them.
2. Create a GRE tunnel between RouterA and RouterC and configure Ethernet over GRE on them to transmit packets between PC1 and PC2 over the GRE tunnel.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 20.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
```

Configure RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 20.1.1.2 255.255.255.0
```

```
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 30.1.1.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] ip address 30.1.1.2 255.255.255.0
[RouterC-GigabitEthernet1/0/0] quit
```

Step 2 Configure OSPF on the routers.

Configure RouterA.

```
[RouterA] ospf 1
[RouterA-ospf-1] area 0
[RouterA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterA-ospf-1-area-0.0.0.0] quit
[RouterA-ospf-1] quit
```

Configure RouterB.

```
[RouterB] ospf 1
[RouterB-ospf-1] area 0
[RouterB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] quit
```

Configure RouterC.

```
[RouterC] ospf 1
[RouterC-ospf-1] area 0
[RouterC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.0] quit
[RouterC-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on RouterA and RouterC. You can find that they have learned the OSPF routes destined for the network segment of the peer.

The command output on RouterA is used as an example.

```
[RouterA] display ip routing-table protocol ospf
<keyword conref=" ../commonterms/commonterms.xml#commonterms/route-flags"></
keyword>
-----
Public routing table : OSPF
    Destinations : 1          Routes : 1

OSPF routing table status : <Active>
    Destinations : 1          Routes : 1

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
    30.1.1.0/24    OSPF   10   2       D    20.1.1.2
GigabitEthernet1/0/0

OSPF routing table status : <Inactive>
    Destinations : 0          Routes : 0
```

Step 3 Configure tunnel interfaces and create a GRE tunnel.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] tunnel-protocol gre
[RouterA-Tunnel0/0/1] ip address 10.3.1.1 255.255.255.0
[RouterA-Tunnel0/0/1] source 20.1.1.1
[RouterA-Tunnel0/0/1] destination 30.1.1.2
[RouterA-Tunnel0/0/1] quit
```

Configure RouterC.

```
[RouterC] interface tunnel 0/0/1
[RouterC-Tunnel0/0/1] tunnel-protocol gre
[RouterC-Tunnel0/0/1] ip address 10.3.1.2 255.255.255.0
[RouterC-Tunnel0/0/1] source 30.1.1.2
[RouterC-Tunnel0/0/1] destination 20.1.1.1
[RouterC-Tunnel0/0/1] quit
```

After the configuration is complete, the tunnel interfaces turn Up and can ping each other.

The command output on RouterA is used as an example.

```
[RouterA] ping -a 10.3.1.1 10.3.1.2
PING 10.3.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 10.3.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.3.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

Step 4 Configure Ethernet over GRE.

The configuration on RouterC is the same as that on RouterA. The configuration on RouterA is used as an example.

Configure a Layer 2 VE interface VE0/0/2 and bind it to the LAN-side physical Ethernet interface GE2/0/0.

```
[RouterA] vlan 100
[RouterA-vlan100] quit
[RouterA] interface virtual-ethernet 0/0/2
[RouterA-Virtual-Ethernet0/0/2] portswitch
[RouterA-Virtual-Ethernet0/0/2] port link-type access
[RouterA-Virtual-Ethernet0/0/2] port default vlan 100
[RouterA-Virtual-Ethernet0/0/2] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet0/0/2] map interface virtual-ethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] quit
```

Configure a Layer 2 VE interface VE0/0/1 and bind it to the WAN-side tunnel interface Tunnel0/0/1.

```
[RouterA] interface virtual-ethernet 0/0/1
[RouterA-Virtual-Ethernet0/0/1] portswitch
[RouterA-Virtual-Ethernet0/0/1] port link-type trunk
[RouterA-Virtual-Ethernet0/0/1] port trunk allow-pass vlan 100
[RouterA-Virtual-Ethernet0/0/1] quit
[RouterA] interface tunnel 0/0/1
[RouterA-Tunnel0/0/1] map interface virtual-ethernet 0/0/1
[RouterA-Tunnel0/0/1] quit
```

Step 5 Verify the configuration.

After the configurations are complete, PC1 and PC2 can ping each other successfully.

----End

Configuration Files

- RouterA configuration file

```
#
 sysname RouterA
#
vlan batch 100
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 map interface Virtual-Ethernet0/0/2
#
interface Virtual-Ethernet0/0/1
 portswitch
 port link-type trunk
 port trunk allow-pass vlan 100
#
interface Virtual-Ethernet0/0/2
 portswitch
 port link-type access
 port default vlan 100
#
interface Tunnel0/0/1
 ip address 10.3.1.1 255.255.255.0
 tunnel-protocol gre
 source 20.1.1.1
 destination 30.1.1.2
 map interface Virtual-Ethernet0/0/1
#
ospf 1
 area 0.0.0.0
 network 20.1.1.0 0.0.0.255
#
return
```

- RouterB configuration file

```
#
 sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
 network 20.1.1.0 0.0.0.255
 network 30.1.1.0 0.0.0.255
#
return
```

- RouterC configuration file

```
#
 sysname RouterC
#
vlan batch 100
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
```

```

map interface Virtual-Ethernet0/0/2
#
interface Virtual-Ethernet0/0/1
portswitch
port link-type trunk
port trunk allow-pass vlan 100
#
interface Virtual-Ethernet0/0/2
portswitch
port link-type access
port default vlan 100
#
interface Tunnel0/0/1
ip address 10.3.1.2 255.255.255.0
tunnel-protocol gre
source 30.1.1.2
destination 20.1.1.1
map interface Virtual-Ethernet0/0/1
#
ospf 1
area 0.0.0.0
network 30.1.1.0 0.0.0.255
#
return
    
```

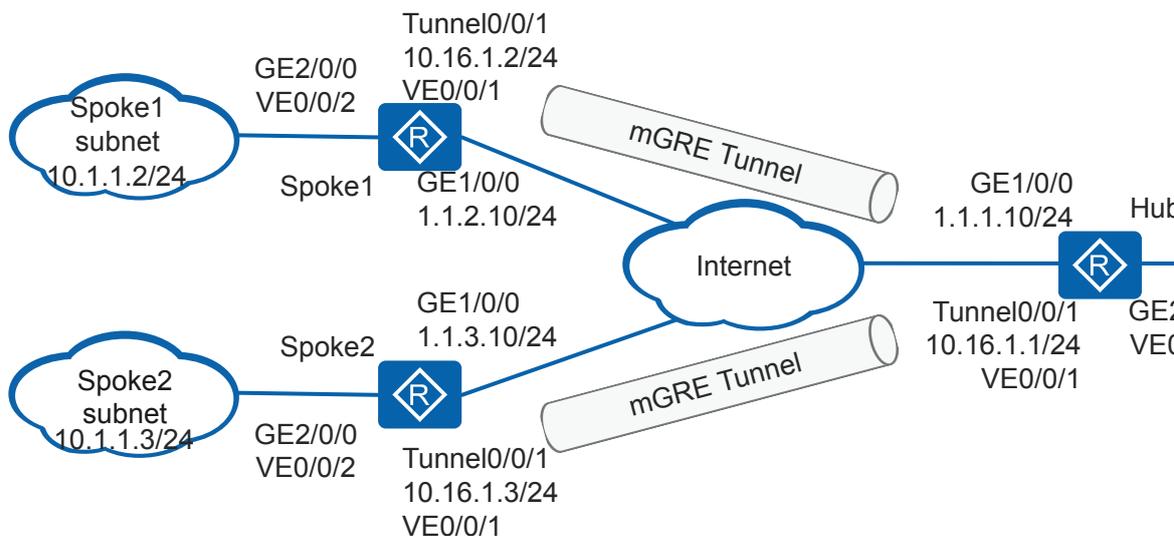
3.8.11 Example for Configuring an Ethernet over mGRE Tunnel

Networking Requirements

In [Figure 3-27](#), a medium-sized enterprise has the headquarters (Hub) and two branches (Spoke1 and Spoke2) located in different areas. The Hub and Spoke subnets use Ethernet networks, and Spokes connect to the public network using dynamic addresses. The enterprise requires that the Hub and Spoke subnets can communicate over the public network.

Assume that the dynamic addresses obtained by Spoke1 and Spoke2 are 1.1.2.10 and 1.1.3.10, respectively.

Figure 3-27 Ethernet over mGRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure DSVPN to implement VPN interconnection between the Spokes because the Spokes connect to the public network using dynamic IP addresses and the Spokes do not know the public IP addresses of each other.
2. Configure the non-shortcut mode because there are only two Spokes.
3. Configure Ethernet over mGRE to enable communication between the Hub and Spoke subnets that are Ethernet networks.

Procedure

Step 1 Configure an IP address for each physical interface.

Configure an IP address for the interface on the Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface gigabitethernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 1.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
```

Configure an IP address for the interface on Spoke1.

```
<Huawei> system-view
[Huawei] sysname Spoke1
[Spoke1] interface gigabitethernet 1/0/0
[Spoke1-GigabitEthernet1/0/0] ip address 1.1.2.10 255.255.255.0
[Spoke1-GigabitEthernet1/0/0] quit
```

Configure an IP address for the interface on Spoke2.

```
<Huawei> system-view
[Huawei] sysname Spoke2
[Spoke2] interface gigabitethernet 1/0/0
[Spoke2-GigabitEthernet1/0/0] ip address 1.1.3.10 255.255.255.0
[Spoke2-GigabitEthernet1/0/0] quit
```

Step 2 Configure OSPF to ensure reachable routes between the routers over the public network.

Configure OSPF on the Hub.

```
[Hub] ospf 2
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 1.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 1.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 1.1.3.0 0.0.0.255
```

```
[Spoke2-ospf-2-area-0.0.0.1] quit  
[Spoke2-ospf-2] quit
```

Step 3 Configure tunnel interfaces and create mGRE tunnels.

Configure tunnel interfaces on the Hub and Spokes and configure the static NHRP peer entry of the Hub on Spoke1 and Spoke2.

Configure a tunnel interface on the Hub.

```
[Hub] interface tunnel 0/0/1  
[Hub-Tunnel0/0/1] tunnel-protocol gre p2mp  
[Hub-Tunnel0/0/1] ip address 10.16.1.1 255.255.255.0  
[Hub-Tunnel0/0/1] source gigabitethernet 1/0/0  
[Hub-Tunnel0/0/1] nhrp entry multicast dynamic  
[Hub-Tunnel0/0/1] quit
```

Configure a tunnel interface and a static NHRP peer entry of the Hub on Spoke1.

```
[Spoke1] interface tunnel 0/0/1  
[Spoke1-Tunnel0/0/1] tunnel-protocol gre p2mp  
[Spoke1-Tunnel0/0/1] ip address 10.16.1.2 255.255.255.0  
[Spoke1-Tunnel0/0/1] source gigabitethernet 1/0/0  
[Spoke1-Tunnel0/0/1] nhrp entry 10.16.1.1 1.1.1.10 register  
[Spoke1-Tunnel0/0/1] quit
```

Configure a tunnel interface and a static NHRP peer entry of the Hub on Spoke2.

```
[Spoke2] interface tunnel 0/0/1  
[Spoke2-Tunnel0/0/1] tunnel-protocol gre p2mp  
[Spoke2-Tunnel0/0/1] ip address 10.16.1.3 255.255.255.0  
[Spoke2-Tunnel0/0/1] source gigabitethernet 1/0/0  
[Spoke2-Tunnel0/0/1] nhrp entry 10.16.1.1 1.1.1.10 register  
[Spoke2-Tunnel0/0/1] quit
```

Step 4 Configure Ethernet over mGRE.

The configurations on Spoke1 and Spoke2 are the same as that on the Hub. The configuration on the Hub is used as an example.

Configure a Layer 2 VE interface VE0/0/2 and bind it to the LAN-side physical Ethernet interface GE2/0/0.

```
[Hub] vlan 100  
[Hub-vlan100] quit  
[Hub] interface virtual-ethernet 0/0/2  
[Hub-Virtual-Ethernet0/0/2] portswitch  
[Hub-Virtual-Ethernet0/0/2] port link-type access  
[Hub-Virtual-Ethernet0/0/2] port default vlan 100  
[Hub-Virtual-Ethernet0/0/2] quit  
[Hub] interface gigabitethernet 2/0/0  
[Hub-GigabitEthernet2/0/0] map interface virtual-ethernet 0/0/2  
[Hub-GigabitEthernet2/0/0] quit
```

Configure a Layer 2 VE interface VE0/0/1 and bind it to the WAN-side tunnel interface Tunnel0/0/1.

```
[Hub] interface virtual-ethernet 0/0/1  
[Hub-Virtual-Ethernet0/0/1] portswitch  
[Hub-Virtual-Ethernet0/0/1] port link-type trunk  
[Hub-Virtual-Ethernet0/0/1] port trunk allow-pass vlan 100  
[Hub-Virtual-Ethernet0/0/1] quit  
[Hub] interface tunnel 0/0/1  
[Hub-Tunnel0/0/1] map interface virtual-ethernet 0/0/1  
[Hub-Tunnel0/0/1] quit
```

Step 5 Verify the configuration.

After the configurations are complete, check the NHRP peer entries on Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1.

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
10.16.1.1      32    1.1.1.10       10.16.1.1     static    hub
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2.

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
10.16.1.1      32    1.1.1.10       10.16.1.1     static    hub
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

The output of the **display nhrp peer all** command indicates that only the static NHRP peer entry of the Hub is displayed on Spoke1 and Spoke2.

On the Hub, check registration information about Spoke1 and Spoke2.

Run the **display nhrp peer all** command on the Hub.

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
10.16.1.2      32    1.1.2.10       10.16.1.2     dynamic   route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
10.16.1.3      32    1.1.3.10       10.16.1.3     dynamic   route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 6 Run the **ping** command and check the configuration result.

The subnet addresses of the Hub, Spoke1, and Spoke2 can successfully ping each other. Spoke1 and Spoke2 can obtain the dynamic NHRP peer of each other.

Run the **display nhrp peer all** command on Spoke1.

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
10.16.1.1      32    1.1.1.10       10.16.1.1     static    hub
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:46:35
```

```

Expire time      : --
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
10.16.1.3       32   1.1.3.10      10.16.1.3     dynamic        route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:00:28
Expire time     : 01:59:32
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
10.16.1.2       32   1.1.2.10      10.16.1.2     dynamic        local
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:00:28
Expire time     : 01:59:32

Number of nhrp peers: 3
  
```

Run the **display nhrp peer all** command on Spoke2.

```

[Spoke2] display nhrp peer all
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
10.16.1.1       32   1.1.1.10      10.16.1.1     static         hub
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:43:32
Expire time     : --
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
10.16.1.2       32   1.1.2.10      10.16.1.2     dynamic        route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
10.16.1.3       32   1.1.3.10      10.16.1.3     dynamic        local
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:00:47
Expire time     : 01:59:13

Number of nhrp peers: 3
  
```

---End

Configuration Files

- Hub configuration file

```

#
sysname Hub
#
vlan batch 100
#
interface GigabitEthernet1/0/0
ip address 1.1.1.10 255.255.255.0
#
interface GigabitEthernet2/0/0
map interface Virtual-Ethernet0/0/2
#
interface Virtual-Ethernet0/0/1
portswitch
  
```

```
port link-type trunk
port trunk allow-pass vlan 100
#
interface Virtual-Ethernet0/0/2
portswitch
port link-type access
port default vlan 100
#
interface Tunnel0/0/1
ip address 10.16.1.1 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
map interface Virtual-Ethernet0/0/1
nhrp entry multicast dynamic
#
ospf 2
area 0.0.0.1
network 1.1.1.0 0.0.0.255
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
#
vlan batch 100
#
interface GigabitEthernet1/0/0
ip address 1.1.2.10 255.255.255.0
#
interface GigabitEthernet2/0/0
map interface Virtual-Ethernet0/0/2
#
interface Virtual-Ethernet0/0/1
portswitch
port link-type trunk
port trunk allow-pass vlan 100
#
interface Virtual-Ethernet0/0/2
portswitch
port link-type access
port default vlan 100
#
interface Tunnel0/0/1
ip address 10.16.1.2 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
map interface Virtual-Ethernet0/0/1
nhrp entry 10.16.1.1 1.1.1.10 register
#
ospf 2
area 0.0.0.1
network 1.1.2.0 0.0.0.255
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
#
vlan batch 100
#
interface GigabitEthernet1/0/0
ip address 1.1.3.10 255.255.255.0
#
interface GigabitEthernet2/0/0
map interface Virtual-Ethernet0/0/2
#
interface Virtual-Ethernet0/0/1
portswitch
port link-type trunk
```

```
port trunk allow-pass vlan 100
#
interface Virtual-Ethernet0/0/2
 portswitch
 port link-type access
 port default vlan 100
#
interface Tunnel0/0/1
 ip address 10.16.1.3 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 map interface Virtual-Ethernet0/0/1
 nhrp entry 10.16.1.1 1.1.1.10 register
#
ospf 2
 area 0.0.0.1
  network 1.1.3.0 0.0.0.255
#
return
```

3.9 Troubleshooting GRE

This section describes common faults caused by incorrect configurations and provides the troubleshooting procedure.

3.9.1 Failed to Ping the IP Address of the Remote Tunnel Interface

Fault Description

Failed to ping the IP address of the remote tunnel interface.

Procedure

- When the network layer protocol of one or both ends of a tunnel is Down
 - a. Check that interfaces on both ends of the tunnel use the same tunnel encapsulation mode.

Run the **display this interface** command in the tunnel interface view on both ends of the tunnel to check whether interfaces on both ends use the same tunnel encapsulation mode. If **Tunnel protocol/transport GRE/IP** is displayed, the tunnel encapsulation mode is GRE.

- If the two interfaces use different tunnel encapsulation modes, run the **tunnel-protocol** command in the tunnel interface view to reconfigure the tunnel encapsulation mode of one interface to be the same as that of the other interface.

NOTE

After reconfiguring the tunnel encapsulation mode, reconfigure the tunnel source and destination addresses because configurations of the original source and destination addresses are lost.

- If the two interfaces use the same tunnel encapsulation mode, go to step 2.
- b. Check that an IP address, a tunnel source address (or interface), and a tunnel destination address are configured for interfaces on both ends of the tunnel.

Check whether the local tunnel source address (or interface) is the peer tunnel destination address and the local tunnel destination address is the peer tunnel source

address (or interface). If not, no tunnel can be established between the two interfaces. A tunnel source address (or interface) and a tunnel destination address uniquely identify a tunnel.

Run the **display this** command in the tunnel interface view to check the interface configuration. Ensure that the local tunnel source address (or interface) is the peer tunnel destination address and the local tunnel destination address is the peer tunnel source address.

- If the tunnel source (or interface) and destination addresses are incorrect, run the **source** and **destination** commands in the tunnel interface view to reconfigure the tunnel source (or interface) and destination addresses.
 - If the tunnel source (or interface) and destination addresses are correct, go to step 3.
- c. Check that there are reachable routes between the tunnel source and destination addresses.

If the interface configurations on both ends are correct but the tunnel status is still Down, check whether there are reachable routes between interfaces on both ends of the tunnel.

- If the tunnel is established between two indirectly connected interfaces, check whether there are reachable routes between the two interfaces.
- If the tunnel is established between two directly connected interfaces, a direct route exists.

Run the **display ip routing-table** command to view the IP routing table. If the IP routing table is correct, run the **display fib** command to check the forwarding table (FIB table) and check whether data is correctly forwarded.

If there is no reachable route between the tunnel source (or interface) and destination addresses, configure static routes between the tunnel source and destination addresses or configure a dynamic routing protocol to calculate reachable routes.

- When the network layer protocol of interfaces on both ends of a tunnel is Up
 - a. Check that GRE key configurations of interfaces on both ends are consistent.

Run the **display interface tunnel** command on the two interfaces to check whether their GRE key configurations are consistent. Ensure that:

- Neither of the two interfaces is configured with a GRE key.
- The two interfaces are configured with the same key.

If GRE key configurations of interfaces on both ends are consistent but the fault persists, go to step 2.

- b. Check IP addresses of interfaces on both ends of the tunnel.

If the network protocol status of the two interfaces is Up but they cannot ping each other, check whether their IP addresses are on the same network segment. If the IP addresses are on different network segments, configure static routes between the two devices or configure a dynamic routing protocol to calculate reachable routes.

----End

3.9.2 Tunnel Interface Alternates Between Up and Down States

Fault Description

After a GRE tunnel is configured, the tunnel interface alternates between Up and Down states.

Troubleshooting Procedure

Run the **display fib destination-address** command in any view multiple times to check whether the outbound interface in the FIB entry to the specified destination network segment changes. For example, the command outputs are displayed as follows:

```
<Huawei> display fib 10.2.1.0
Route Entry Count: 1
Destination/Mask  Nexthop      Flag  TimeStamp      Interface  TunnelID
10.2.1.0/24      10.3.1.2     SU    t[631838]      Tun0/0/1  0x0
<Huawei> display fib 10.2.1.0
Route Entry Count: 1
Destination/Mask  Nexthop      Flag  TimeStamp      Interface  TunnelID
10.2.1.0/24      1.1.1.2     GSU   t[631840]      GE1/0/0   0x0
```

The command outputs show that the outbound interface to 10.2.1.0 changes. As a result, the tunnel interface flaps.

If the outbound interface changes, adjust the network topology to ensure that the outbound interface is not a GRE tunnel interface. For VLL, PWE3, and BGP/MPLS IP VPN services, you are advised to run the **tunnel-policy (system view)** command to select the GRE tunnel by tunnel policy.

3.10 FAQ About GRE

This section describes the FAQ about GRE.

3.10.1 Can the MTU of the GRE Tunnel Interface Take Effect?

If you set an MTU value on a GRE tunnel interface, forwarding of data packets through the GRE tunnel will be affected. If the packet length exceeds the MTU value on the tunnel interface, the device fragments the packet.

3.11 References for GRE

This section lists references for GRE.

The following table lists the references for GRE.

Table 3-3 References for GRE

Document	Description	Remarks
RFC1701	Generic Routing Encapsulation (GRE)	-
RFC1702	Generic Routing Encapsulation over IPv4 networks	-

Document	Description	Remarks
RFC2784	Generic Routing Encapsulation (GRE)	-

4 SVPN Configuration

About This Chapter

An enterprise can use the smart virtual private network (SVPN) function to bind multiple WAN links to realize high-bandwidth and highly-reliable access to a public network, providing high-quality communication.

[4.1 Overview of SVPN](#)

This section describes the definition, purpose, and benefits of Smart Virtual Private Network (SVPN).

[4.2 Understanding SVPN](#)

This section describes the SVPN working mode and service forwarding process.

[4.3 Application Scenarios for SVPN](#)

This section describes the application scenarios for SVPN.

[4.4 Summary of SVPN Configuration Tasks](#)

SVPN supports two modes: Lone Ranger and Hub-Spoke.

[4.5 Licensing Requirements and Limitations for SVPN](#)

This section describes SVPN configuration notes.

[4.6 Default Settings for SVPN](#)

This section describes the default settings for SVPN.

[4.7 Configuring SVPN](#)

This section describes the procedures for configuring SVPN.

[4.8 Configuration Examples for SVPN](#)

This section provides configuration examples about SVPN, including networking requirements, configuration roadmap, configuration procedure, and configuration files.

4.1 Overview of SVPN

This section describes the definition, purpose, and benefits of Smart Virtual Private Network (SVPN).

Definition

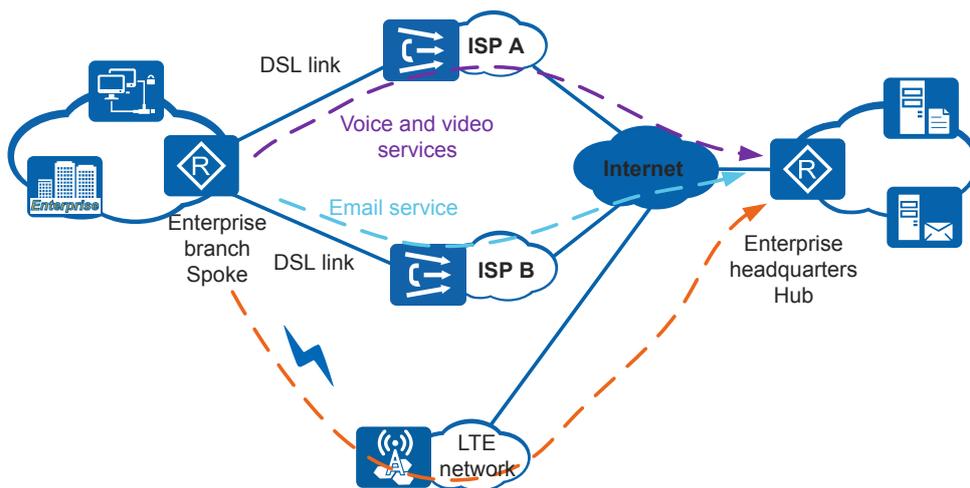
SVPN binds multiple WAN access lines to provide high bandwidth and reliable networks for customers.

Purpose

With increasing enterprise branches and industry sites and the rapid development of the Internet of Things (IoT), there are higher demands for Internet access. Digital Subscriber Line (DSL) access lines gradually become unable to meet customer requirements such as high bandwidth. Dedicated lines can be used to solve this problem. However, most companies consider that dedicated lines are expensive and they are in urgent need for a cost-effective Internet access solution.

To reduce the cost on leased lines, the majority of enterprises purchase multiple DSL links to access the Internet. As shown in **Figure 4-1**, an enterprise purchases a DSL link from ISP A and the other DSL link from ISP B, and an LTE link as the secondary link. When both DSL links are faulty, services are switched to the LTE link to ensure service continuity.

Figure 4-1 Accessing the Internet through multiple WAN links



According to the enterprise planning, voice and video services and the email service which is Big Data service are transmitted over the two DSL links respectively. However, the following problems exist in an actual application:

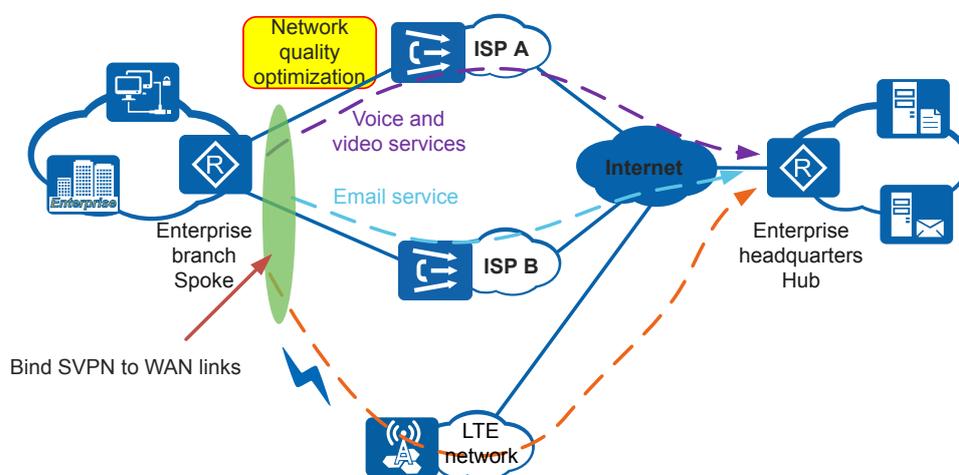
- If the voice and video services are to be transmitted over the optimal link all along, the configuration and maintenance are complicated.
- If data of the email service bursts and then load balanced to the DSL link of ISP A, the transmission quality of voice and video services cannot be guaranteed.

SVPN can be used to solve the preceding problems and provide a cost-effective Internet access solution for customers. SVPN binds multiple WAN access links and schedules service flows to the corresponding WAN links for transmission. SVPN can provide differentiated services for various data flows.

- Through link detection (associated NQA instances or the default link quality measurement method), SVPN can ensure that the voice and video services are transmitted over the optimal link all along.
- By allocating link bandwidth for each service type to implement load balancing, SVPN guarantees the transmission quality of the voice and video services and fully uses each link.

In addition, SVPN has a high reliability as it allows the links to back up each other. When either link is faulty, services are switched to the other link to ensure service continuity.

Figure 4-2 Binding multiple WAN access links of an enterprise using SVPN



As shown in **Figure 4-2**, after SVPN is deployed, the email service of the enterprise is preferentially transmitted over the DSL link of ISP B (While ISP A provide a DSL link which reserves 1 Mbit/s bandwidth for the service), and the voice and video services are transmitted over the DSL link of ISP A with high network quality. When data of the email service bursts, SVPN load balances the data to the DSL link of ISP A on the premise that the transmission quality of voice and video services are guaranteed. When both DSL links are faulty, services are switched to the LTE link to ensure service continuity.

Benefits

SVPN fully uses the existing WAN access links of an enterprise to reduce the cost on leased lines. Meanwhile, it implements high-bandwidth and reliable Internet access to provide high-quality communication for users.

4.2 Understanding SVPN

This section describes the SVPN working mode and service forwarding process.

Working Mode

SVPN has two working modes:

- Lone Ranger mode

This mode applies when an enterprise egress gateway connects to the public network through multiple links. For details, see [4.3.1 Fully Using WAN Links for Internet Access Through Lone Ranger SVPN](#) in Applications.

After SVPN is deployed on the enterprise egress gateway, the gateway uses SVPN to forward traffic to the public network, without the need for SVPN tunnel negotiation.

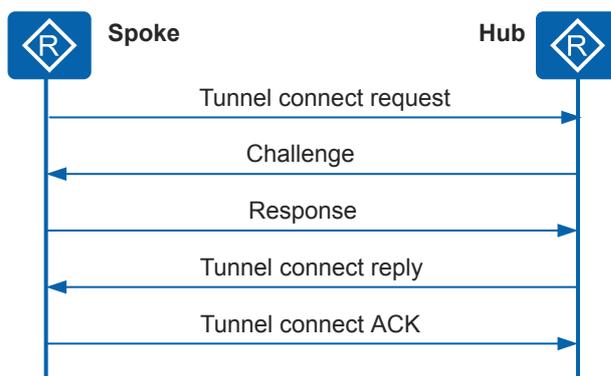
- Hub-Spoke mode

This mode applies when the hub and spoke devices are connected through multiple links. For details, see [4.3.2 Fully Using WAN Links for Implementing Interworking Between the Enterprise Branch and Headquarters Through Hub-Spoke SVPN](#) in Applications.

After a spoke device successfully connects to the public network, the device initiates a tunnel registration process along each link to the hub to establish connections with the hub device. The tunnels along multiple links form a logical tunnel for end-to-end service transmission.

Figure 4-3 shows the SVPN tunnel registration process in Hub-Spoke mode.

Figure 4-3 Tunnel registration process



- The spoke sends a Tunnel Connect Request message carrying parameters such as the tunnel ID and SVPN domain to the hub.
- After receiving the Tunnel Connect Request message, the hub sends a Challenge message to the spoke.
- After receiving the Challenge message, the spoke encrypts the message, encapsulates the generated cipher text and its own user name in a Response message, and sends the Response message to the hub.
- The hub encrypts the Challenge message using the locally saved password and compares the obtained cipher text with that carried in the Response message. If they are the same, authentication succeeds. The hub then sends a Tunnel Connect Reply message carrying parameters such as the tunnel ID and SVPN domain to the spoke.
- After receiving the Tunnel Connect Reply message, the spoke sends a Tunnel Connect ACK message to the hub.

After the preceding process, an SVPN tunnel is established successfully.

Service Traffic Forwarding

You need to configure an SVPN proposal to implement service traffic forwarding over an SVPN tunnel. An SVPN proposal is a basic SVPN component that defines various parameters, including the encapsulation type for SVPN packets, service classification, and scheduling policy.

Packet Encapsulation

SVPN supports two packet encapsulation types:

- No encapsulation: The original IP packet structure is not changed.
- GRE encapsulation: A GRE header and an IP header are added to change the original IP packet structure. For details on GRE encapsulation, see "Principles" in [GRE Configuration](#).

Service Classification

SVPN distinguishes service flows by service type. Two methods are available for classifying service flows:

- Access Control List (ACL): The device obtains the service type by matching service packets with corresponding ACL and then schedules the service flows accordingly. For details, see "ACL Configuration" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Security*.
- Smart Application Control (SAC): The device uses the service awareness (SA) technology to obtain the application type of service flows and then schedules the service flows accordingly. For details, see "SAC Configuration" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - QoS*.

Scheduling Policy

After services are classified, the device needs to schedule service flows using specified scheduling policies. You can specify a scheduling policy for each service. Two types of scheduling policies are supported:

- Overflow mode: The device schedules service flows based on the path bandwidth. Packets of one service flow can be transmitted over different paths. If bandwidth of the first available path is used up, packets are distributed to a second available path, and so on.
 - In practice, SVPN adjusts bandwidth of the forwarding paths in real time based on the packet loss rate calculated using the default link quality measurement method to provide better service. When the packet loss rate on a path is high, you can decrease bandwidth of this path to allow service flows to be forwarded through a second available path. When the packet loss rate is low, you can increase bandwidth of the path to switch some service flows back to the path.
 - When SVPN selects the second available path, if the link quality measurement result (obtained through NAQ or the default measurement method) shows that the delay of the second available path is much larger than the delay of the first available path, SVPN does not use this path but searches for another path that meets the link quality requirement.
- Priority mode: The device schedules service flows based on path priorities. Packets of one service flow can be transmitted over only one optimal path. When selecting the

optimal path, the device figures out qualified forwarding paths according to the link quality measurement results (obtained through NQA or the default measurement method) and then selects the first available path from these paths.

The path quality is measured by the delay (D), jitter (J), packet loss rate (L), and Composite Measure Indicator (CMI).

If any of the following situations occurs on a path, the path does not meet the requirement.

- The CMI value calculated using the CMI calculation formula is smaller than the configured CMI threshold.
- The path delay is larger than the configured delay threshold.
- The path jitter is larger than the configured jitter threshold.
- The packet loss rate value is larger than the configured packet loss rate threshold.

Different services have different requirements on the delay, jitter, loss, and CMI. If a service does not have high requirements on one of these indicators, you do not need to set a threshold for the indicator.

Select the Overflow mode when link bandwidth needs to be fully used for service transmission. Select the Priority mode when you want to transmit a service along the optimal path.

 **NOTE**

In Lone Ranger mode, SVPN does not support the default link quality measurement method.

In HUB-Spoke mode, SVPN does not support the default link quality measurement method when no encapsulation is used.

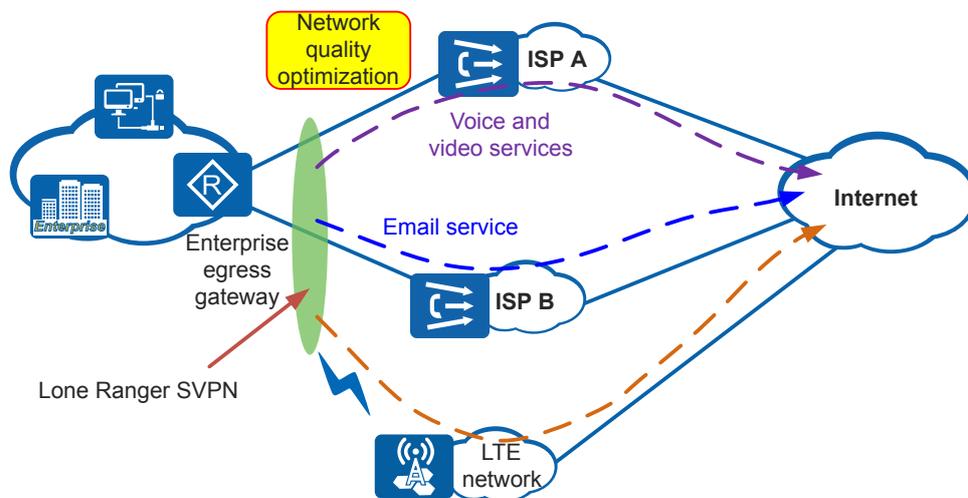
4.3 Application Scenarios for SVPN

This section describes the application scenarios for SVPN.

4.3.1 Fully Using WAN Links for Internet Access Through Lone Ranger SVPN

An enterprise gateway can connect to the Internet through multiple WAN links. Lone Ranger SVPN can be used to bind these WAN links to improve link bandwidth or schedule service flows to the optimal WAN link, making full use of the WAN links.

Figure 4-4 Fully using WAN links for Internet access through Lone Ranger SVPN

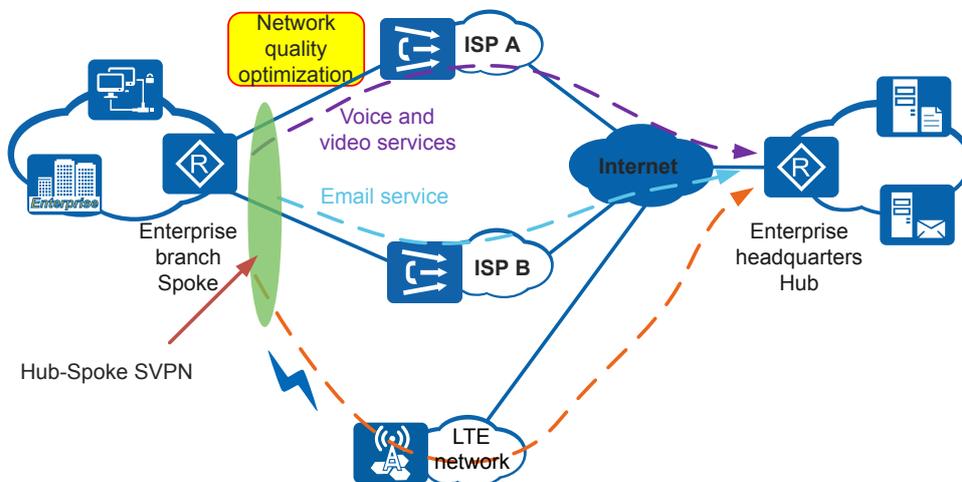


Lone Ranger SVPN can be used to schedule service flows destined for the Internet to transmit the voice and video services over the optimal path and the email service over the path with the minimum cost and then other available paths.

4.3.2 Fully Using WAN Links for Implementing Interworking Between the Enterprise Branch and Headquarters Through Hub-Spoke SVPN

In a Hub-Spoke networking, you can configure Hub-Spoke SVPN to effectively use the WAN links for the hub and spoke and implement high-bandwidth and high-quality communication between the headquarters and branch.

Figure 4-5 Fully using WAN links for implementing interworking between the enterprise branch and headquarters through Hub-Spoke SVPN



Hub-Spoke SVPN can be used to schedule service flows destined for the enterprise headquarters to transmit the voice and video services over the optimal path and the email service over the path with the minimum cost and then other available paths.

4.4 Summary of SVPN Configuration Tasks

SVPN supports two modes: Lone Ranger and Hub-Spoke.

Table 4-1 lists summary of SVPN configuration tasks.

Table 4-1 SVPN configuration tasks

Scenario	Description	Task
Configure Lone Ranger SVPN to connect to a public network	When the device functions as the enterprise egress gateway, it can connect to a public network through multiple WAN links. Lone Ranger SVPN can be configured in this scenario to effectively use these WAN links. For example, multiple WAN links can be bound together to improve link bandwidth, or service flows can be transmitted through the optimal WAN link.	4.7.1 Configuring Lone Ranger SVPN

Scenario	Description	Task
Configure Hub-Spoke SVPN to connect the headquarters and branch VPNs over a public network	In a Hub-Spoke networking, the hub in the headquarters and the spoke in the branch can connect to a public network through WAN links. Hub-Spoke SVPN can be configured in this scenario to effectively use these WAN links and implement high-bandwidth and high-quality communication between the headquarters and branch.	4.7.2 Configuring Hub-Spoke SVPN

4.5 Licensing Requirements and Limitations for SVPN

This section describes SVPN configuration notes.

Involved Network Elements

None

Feature Limitations

- The feature is just for beta test, and is not for commercial use. If the feature is required in the test, contact Huawei technical support personnel.
- Fully using WAN links for implementing interworking between the enterprise branch and headquarters through Hub-Spoke SVPN, when traffic transmitted between a branch and the headquarters over the public network passes through a NAT device, SVPN does not support NAT traversal.

4.6 Default Settings for SVPN

This section describes the default settings for SVPN.

[Table 4-2](#) describes the default settings for SVPN.

Table 4-2 Default settings for SVPN

Parameter	Default Setting
Encapsulation type for SVPN packets	NULL
SVPN zone for an SVPN tunnel interface	0
Delay for switching a service flow between forwarding paths	5s

Parameter	Default Setting
Size of the receive or transmit buffer for SVPN packets	64

4.7 Configuring SVPN

This section describes the procedures for configuring SVPN.

4.7.1 Configuring Lone Ranger SVPN

When the device functions as the enterprise egress gateway, it can connect to a public network through multiple WAN links. Lone Ranger SVPN can be configured in this scenario to effectively use these WAN links.

Pre-configuration Tasks

Before configuring Lone Ranger SVPN, complete the following task:

- Connecting the devices to a public network through DSL and LTE links

NOTE

Lone Ranger SVPN only processes service flows destined from the local device to the public network but does not apply to service flows destined from the public network to the local device.

Configuration Process

Perform the following operations on the device to configure Lone Ranger SVPN.

4.7.1.1 Configuring a Tunnel Interface

Context

To implement the Lone Ranger SVPN function, create tunnel interfaces and set the interface type to SVPN. There is no need to specify a source or destination address for a tunnel interface. You can bind an SVPN proposal to each tunnel interface to specify the forwarding path.

Perform the following operations on the devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

Step 3 Run **ip address** *ip-address* { *mask* | *mask-length* }

An IP address is configured for the tunnel interface.

Step 4 Run **tunnel-protocol svpn**

The tunnel encapsulation type is set to SVPN.

By default, the tunnel encapsulation type is none, indicating that packets are not encapsulated.

Step 5 (Optional) Run **svpn-zone zone-number**

The SVPN zone is specified for an SVPN tunnel interface.

By default, a tunnel interface belongs to SVPN zone 0.

When multiple SVPN tunnel interfaces are configured on a device, run this command to configure different SVPN zones for these interfaces to ensure proper data forwarding.

If SVPN packets are encapsulated using GRE, the two devices connected must be configured with the same SVPN zone.

Step 6 (Optional) Run **mtu mtu**

An MTU is configured for the tunnel interface.

By default, the MTU of a tunnel interface is 1500 bytes.

If the original data packet contains more than 1518 bytes, the physical outbound interface on the device re-fragments the data packet. In this case, the device may fail to reassemble the packet. It is recommended that the MTU value on the tunnel interface be smaller than or equal to 1474.

NOTE

To change the MTU of a tunnel interface, run the **shutdown** command and then the **undo shutdown** command on the interface to make the new MTU effective.

----End

4.7.1.2 Configuring an SVPN Proposal

Context

An SVPN proposal is a basic component of SVPN and defines various parameters, including the encapsulation type for SVPN packets, service type, and forwarding paths for a service flow.

SVPN distinguishes service flows based on the service type in an SVPN proposal. Two path scheduling modes are available for a service flow.

- **Overflow mode:** The device schedules the service flow based on the path bandwidth. Packets of one service flow can be transmitted over different paths. If the bandwidth of the first available path is occupied, packets are scheduled to a second available path, and so on.

When the Overflow mode is used:

- You can specify the bandwidth for a forwarding path. If bandwidth is not specified, the actual physical bandwidth of the source interface is used. In practice, SVPN adjusts the bandwidth of the forwarding paths based on the packet loss rate calculated using link detection (associated the default link quality measurement method) in real time to provide a better service.

- If the link detection result shows that the delay of a second available path is much larger than the delay of the first available path, SVPN does not use this path but searches for another path that meets the link quality requirement.
- Priority mode: The device schedules the service flow based on the path priority. Packets of one service flow can be transmitted over only one optimal path. When selecting the optimal path, the device calculates forwarding paths that meet the criteria using link detection (associated NQA instances or the default link quality measurement method) and then selects the first available path from these paths.

To make full use of the link bandwidth, select the configuration task "Configuring an SVPN Proposal of the Overflow Mode." To transmit service flows through the link with the optimal network quality, select the configuration task "Configuring an SVPN Proposal of the Priority Mode."

 **NOTE**

A maximum of eight service types can be configured in an SVPN proposal, and the services can use different forwarding path scheduling modes.

When multiple service types are configured in an SVPN proposal, you only need to repeat the service type-related configurations in the corresponding configuration task, including specifying the service flow forwarding paths to be used and the scheduling mode, packet matching mode, and composite measure indicator (CMI) threshold for a path.

4.7.1.3 Binding an SVPN Proposal to a Tunnel Interface

Prerequisite

A tunnel interface has been configured and an SVPN proposal has been created.

Context

After an SVPN proposal is created, you need to bind the SVPN proposal to an SVPN tunnel interface. After the SVPN proposal is bound to a tunnel interface, packets received by this interface are forwarded along the path for SVPN service packets.

Perform the following operations on the devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 3 Run **svpn-proposal** *svpn-proposal-name*

An SVPN proposal is bound to a tunnel interface.

---End

4.7.1.4 Importing Service Flows to an SVPN Tunnel

Context

After an SVPN tunnel is created, service flows cannot be automatically imported to the tunnel. You need to configure a mode for importing service flows. In Lone Ranger SVPN mode, you only need to configure static routes to import service flows to an SVPN tunnel.

Perform the following operations on the devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ip route-static** *ip-address* { *mask* | *mask-length* } **tunnel** *interface-number* [**description** *text*]

A static route is configured.

---End

4.7.2 Configuring Hub-Spoke SVPN

In a Hub-Spoke networking, you can configure Hub-Spoke SVPN to effectively use the WAN links for the hub and spoke and implement high-bandwidth and high-quality communication between the headquarters and branch.

Pre-configuration Tasks

Before configuring Hub-Spoke SVPN, complete the following tasks:

- Connecting the devices to a public network through DSL and LTE links
- Configuring reachable routes between the hub and spoke over the public network

Configuration Process

Perform the following operations on devices to configure Hub-Spoke SVPN.

4.7.2.1 Configuring a Tunnel Interface

Context

To implement the Hub-Spoke SVPN function, create tunnel interfaces and set the interface type to P2P SVPN or P2MP SVPN. There is no need to specify a source or destination address for a tunnel interface. You can bind an SVPN proposal to each tunnel interface to specify the forwarding path.

Perform the following operations on the spoke and hub.

Procedure

- Configuring the spoke
 - a. Run **system-view**

The system view is displayed.

b. Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

c. Run **ip address** *ip-address2* { *mask* | *mask-length* }

An IP address is configured for the tunnel interface.

d. Run **tunnel-protocol svpn p2p**

The tunnel encapsulation type is set to P2P SVPN.

By default, the tunnel encapsulation type is none, indicating that packets are not encapsulated.

e. (Optional) Run **svpn-zone** *zone-number*

The SVPN zone is specified for an SVPN tunnel interface.

By default, a tunnel interface belongs to SVPN zone 0.

When multiple SVPN tunnel interfaces are configured on a device, run this command to configure different SVPN zones for these interfaces to ensure proper data forwarding.

 **NOTE**

In Hub-Spoke mode, SVPN zone for the tunnel interface of the hub must be the same as that of the spoke.

● **Configuring the hub**

a. Run **system-view**

The system view is displayed.

b. Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

c. Run **ip address** *ip-address1* { *mask* | *mask-length* }

An IP address is configured for the tunnel interface.

d. Run **tunnel-protocol svpn p2mp**

The tunnel encapsulation type is set to P2MP SVPN.

By default, the tunnel encapsulation type is none, indicating that packets are not encapsulated.

e. (Optional) Run **svpn-zone** *zone-number*

The SVPN zone is specified for an SVPN tunnel interface.

By default, a tunnel interface belongs to SVPN zone 0.

When multiple SVPN tunnel interfaces are configured on a device, run this command to configure different SVPN zones for these interfaces to ensure proper data forwarding.

 **NOTE**

In Hub-Spoke mode, SVPN zone for the tunnel interface of the hub must be the same as that of the spoke.

----End

4.7.2.2 Configuring an SVPN Proposal

Context

An SVPN proposal is a basic component of SVPN and defines various parameters, including the encapsulation type for SVPN packets, service type, and forwarding paths for a service flow. In Hub-Spoke SVPN mode, the service type-related configurations on the hub and spoke are different. On the spoke, you only need to specify the forwarding paths and scheduling mode for service flows. On the hub, you only need to configure a matching mode for service flows of a specified service type configured on the spoke.

SVPN distinguishes service flows based on the service type in an SVPN proposal. Two path scheduling modes are available for a service flow.

- **Overflow mode:** The device schedules the service flow based on the path bandwidth. Packets of one service flow can be transmitted over different paths. If the bandwidth of the first available path is occupied, packets are scheduled to a second available path, and so on.

When the Overflow mode is used:

- You can specify the bandwidth for a forwarding path. If bandwidth is not specified, the actual physical bandwidth of the source interface is used. In practice, SVPN adjusts the bandwidth of the forwarding paths based on the packet loss rate calculated using link detection (associated the default link quality measurement method) in real time to provide a better service.
- If the link detection result shows that the delay of a second available path is much larger than the delay of the first available path, SVPN does not use this path but searches for another path that meets the link quality requirement.

- **Priority mode:** The device schedules the service flow based on the path priority. Packets of one service flow can be transmitted over only one optimal path. When selecting the optimal path, the device calculates forwarding paths that meet the criteria using link detection (associated NQA instances or the default link quality measurement method) and then selects the first available path from these paths.

To make full use of the link bandwidth, select the configuration task "Configuring an SVPN Proposal of the Overflow Mode." To transmit service flows through the link with the optimal network quality, select the configuration task "Configuring an SVPN Proposal of the Priority Mode."

NOTE

A maximum of eight service types can be configured in an SVPN proposal, and the services can use different forwarding path scheduling modes.

When multiple service types are configured in an SVPN proposal, you only need to repeat the service type-related configurations in the corresponding configuration task, including specifying the service flow forwarding paths to be used and the scheduling mode, packet matching mode, and composite measure indicator (CMI) threshold for a path.

4.7.2.3 Binding an SVPN Proposal to a Tunnel Interface

Prerequisite

A tunnel interface has been configured and an SVPN proposal has been created.

Context

After an SVPN proposal is created, you need to bind the SVPN proposal to an SVPN tunnel interface. After the SVPN proposal is bound to a tunnel interface, packets received by this interface are forwarded along the path for SVPN service packets.

Perform the following operations on the devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 3 Run **svpn-proposal** *svpn-proposal-name*

An SVPN proposal is bound to a tunnel interface.

---End

4.7.2.4 Importing Service Flows to an SVPN Tunnel

Context

After an SVPN tunnel is created, service flows cannot be automatically imported to the tunnel. You need to configure a mode for importing service flows. In Hub-Spoke mode, you can configure static routes or a dynamic routing protocol to import service flows to an SVPN tunnel.

When there are multiple spokes in Hub-Spoke mode, two methods are available for configuring static routes or a dynamic routing protocol:

- Branches learn routes from each other
This method uses the tunnel address of a destination branch as the next hop to the destination subnet. This deployment has a low requirement on the performance of the hub and spokes because the devices only have to learn a small number of routes.
- Branches have only summarized routes to the central office
On a large-sized network with many branch subnets, spokes need to learn many routes from other branches. If the preceding method is used, the spokes must save routing information on the entire network. This requires spokes to maintain a large routing table and provide high performance because many CPU and memory resources are consumed for computing of dynamic routing protocols. To reduce the number of routes saved on spokes, branches have only summarized routes to the central office. In this deployment, the next hop to a destination subnet is the tunnel address of the hub.

Perform the following operations on the spoke and hub.

Procedure

- Configuring static routes

a. Run **system-view**

The system view is displayed.

b. Run **ip route-static** *ip-address* { *mask* | *mask-length* } *nexthop-address* [**description** *text*]

A static route is configured.

 **NOTE**

- Branches learn routes from each other
 You need to configure static routes between the hub and spoke and between spokes and specify the peer tunnel address as the next-hop address.
- Branches have only summarized routes to the central office
 You need to configure static routes on the hub and spokes. The next-hop address of the hub is the peer tunnel address of a spoke, and the next-hop address of a spoke is the tunnel address of the hub.

● Configuring dynamic routes

a. Run **system-view**

The system view is displayed.

b. Configure dynamic routes.

Dynamic routing can be implemented using OSPF, RIP, or BGP. For the configuration of a dynamic routing protocol, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Unicast Routing*.

When configuring different dynamic routing protocols, note the following points:

Deployment Mode and Routing Protocol	RIP	OSPF	BGP
Branches learn routes from each other	Disable the split horizon and automatic route aggregation functions on the SVPN interface of the hub.	Run the ospf network-type broadcast command to set the OSPF network type to broadcast on the hub and spokes.	Do not configure route aggregation on the hub.
Branches have only summarized routes to the central office	Enable the split horizon and automatic route aggregation functions on the SVPN interface of the hub.	Run the ospf network-type p2mp command to set the OSPF network type to P2MP on the hub and spokes.	Configure route aggregation on the hub.

----End

4.8 Configuration Examples for SVPN

This section provides configuration examples about SVPN, including networking requirements, configuration roadmap, configuration procedure, and configuration files.

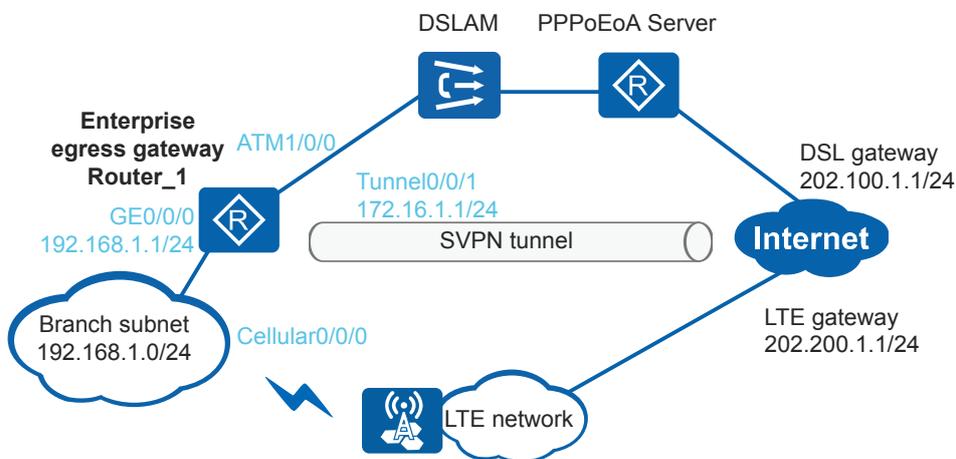
4.8.1 Example for Configuring Lone Ranger SVPN of the Overflow Mode

Networking Requirements

As shown in [Figure 4-6](#), an enterprise purchased two WAN links: a DSL link with 2 Mbit/s bandwidth and an LTE link which is a yearly-package service. Users in the enterprise automatically dial up to connect to the Internet. The enterprise requires to fully use bandwidth of the DSL and LTE links and transmit a service flow preferentially over the DSL link. When DSL link bandwidth is occupied, the service flow can be transmitted over the LTE link.

Lone Ranger SVPN of Overflow mode can be configured to meet this requirement. After a service flow is imported to an SVPN tunnel, SVPN schedules the service flow based on the Overflow mode to fully use the DSL link.

Figure 4-6 Configuring Lone Ranger SVPN of the Overflow mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Connect the enterprise gateway to the Internet over the DSL and LTE links. Connect the branch gateway to the enterprise gateway.

2. Configure Lone Ranger SVPN of the Overflow mode. Configure two forwarding paths (DSL link and LTE link) for a service flow and specify the scheduling mode to fully use the DSL link.

Procedure

- Step 1** Connect the enterprise gateway to the Internet and the branch subnet to the enterprise gateway.

Configure a DSL link for the enterprise gateway Router_1.

```
<Huawei> system-view
[Huawei] sysname Router_1
[Router_1] dialer-rule
[Router_1-dialer-rule] dialer-rule 10 ip permit
[Router_1-dialer-rule] quit
[Router_1] interface dialer 1
[Router_1-Dialer1] dialer user u1
[Router_1-Dialer1] dialer-group 10
[Router_1-Dialer1] dialer bundle 12
[Router_1-Dialer1] ip address ppp-negotiate
[Router_1-Dialer1] link-protocol ppp
[Router_1-Dialer1] ppp chap user huawei
[Router_1-Dialer1] ppp chap password cipher huawei@1234
[Router_1-Dialer1] quit
[Router_1] interface virtual-ethernet 0/0/0
[Router_1-Virtual-Ethernet0/0/0] pppoe-client dial-bundle-number 12
[Router_1-Virtual-Ethernet0/0/0] quit
[Router_1] interface atm 1/0/0
[Router_1-Atm1/0/0] pvc pppoeoa 2/45
[Router_1-atm-pvc-Atm1/0/0-2/45-pppoeoa] map bridge virtual-ethernet 0/0/0
[Router_1-atm-pvc-Atm1/0/0-2/45-pppoeoa] quit
[Router_1-Atm1/0/0] quit
[Router_1] ip route-static 202.100.1.0 24 dialer 1
```

For details on how to configure the DSLAM, see the DSLAM documentation.

Assign IP address 202.1.10.1 to the PPPoEoA server and configure the server to assign IP address 202.1.10.2 to the client. Set the authentication mode to CHAP authentication, and set the user name and password to be the same as those configured on the client.

Configure an LTE link for the enterprise gateway Router_1. The enterprise obtains the following information from the carrier: the APN is ltenet, and the dialer number is *99#.

```
[Router_1] apn profile lteprofile
[Router_1-apn-profile-lteprofile] apn ltenet
[Router_1-apn-profile-lteprofile] quit
[Router_1] interface cellular 0/0/0
[Router_1-Cellular0/0/0] mode lte auto
[Router_1-Cellular0/0/0] dialer enable-circular
[Router_1-Cellular0/0/0] apn-profile lteprofile
[Router_1-Cellular0/0/0] ip address negotiate
[Router_1-Cellular0/0/0] dialer-group 1
[Router_1-Cellular0/0/0] dialer number *99# autodial
[Router_1-Cellular0/0/0] dialer timer autodial 20
[Router_1-Cellular0/0/0] shutdown
[Router_1-Cellular0/0/0] undo shutdown
[Router_1-Cellular0/0/0] quit
[Router_1] ip route-static 202.200.1.0 24 cellular 0/0/0
```

After the device successfully connects to the LTE network, Cellular0/0/0 obtains a dynamic IP address. Assume that the IP address is 202.1.20.2/24 in this example.

Connect the branch subnet to the enterprise gateway.

```
[Router_1] interface gigabitethernet 0/0/0
[Router_1-GigabitEthernet0/0/0] ip address 192.168.1.1 24
[Router_1-GigabitEthernet0/0/0] quit
```

Step 2 Configure a tunnel interface.

```
[Router_1] interface tunnel 0/0/1
[Router_1-Tunnel0/0/1] ip address 172.16.1.1 24
[Router_1-Tunnel0/0/1] tunnel-protocol svpn
[Router_1-Tunnel0/0/1] quit
```

Step 3 Configure an SVPN proposal of the Overflow mode.

```
[Router_1] acl number 3001
[Router_1-acl-adv-3001] rule 5 permit ip
[Router_1-acl-adv-3001] rule 10 permit icmp
[Router_1-acl-adv-3001] quit
[Router_1] svpn-proposal p1
[Router_1-svpn-proposal-p1] source dialer1 destination 202.100.1.1 bandwidth 2048
[Router_1-svpn-proposal-p1] source cellular0/0/0 destination 202.200.1.1
bandwidth 10240
[Router_1-svpn-proposal-p1] service s1 id 1
[Router_1-svpn-proposal-p1-service-s1] schedule-type overflow
[Router_1-svpn-proposal-p1-service-s1] match acl 3001
[Router_1-svpn-proposal-p1-service-s1] source dialer1
[Router_1-svpn-proposal-p1-service-s1] source cellular0/0/0
[Router_1-svpn-proposal-p1-service-s1] quit
[Router_1-svpn-proposal-p1] quit
```

Step 4 Bind the SVPN proposal to the tunnel interface.

```
[Router_1] interface tunnel 0/0/1
[Router_1-Tunnel0/0/1] svpn-proposal p1
[Router_1-Tunnel0/0/1] quit
```

Step 5 Import service flows to an SVPN tunnel.

```
[Router_1] ip route-static 0.0.0.0 0 tunnel 0/0/1
```

Step 6 Verify the configuration.

After the configurations are complete, run the **display ip routing-table** command on Router_1. You can see that routes destined for the Internet pass through the SVPN tunnel.

Connect Port1 on the tester to Router_1 and Port2 on the tester to the Internet (the IP address is 202.10.10.10/24). Inject a 5 Mbit/s service flow from Port1 to Router_1 (the source and destination addresses of the service flow are 192.168.1.100/24 and 202.10.10.10/24). Some packets of the service flow are transmitted over the DSL link at a rate of 2 Mbit/s, and the other packets are transmitted over the LTE link at a rate of 3 Mbit/s.

----End

Configuration Files

Configuration file of Router_1.

```
#
 sysname Router_1
#
acl number 3001
 rule 5 permit ip
 rule 10 permit icmp
#
interface Dialer1
 link-protocol ppp
 ppp chap user huawei
 ppp chap password cipher
 %^%#'&=6Q(|7-#|.]EB`mK$(h7[CY`2m)-YT)Q=Oh2~2%^%#
 ip address ppp-negotiate
```

```
dialer user u1
dialer bundle 12
dialer-group 10
#
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0
#
interface Cellular0/0/0
 dialer enable-circular
 dialer-group 1
 apn-profile lteprofile
 dialer timer autodial 20
 dialer number *99# autodial
 ip address negotiate
#
interface Atm1/0/0
 pvc pppoea 2/45
 map bridge Virtual-Ethernet0/0/0
#
interface Virtual-Ethernet0/0/0
 pppoe-client dial-bundle-number 12
#
svpn-proposal p1
 source Dialer1 destination 202.100.1.1 bandwidth 2048
 source Cellular0/0/0 destination 202.200.1.1 bandwidth 10240
 service s1 id 1
  schedule-type overflow
  match acl 3001
  source Dialer1
  source Cellular0/0/0
#
interface Tunnel0/0/1
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol svpn
 svpn-proposal p1
#
dialer-rule
 dialer-rule 10 ip permit
#
apn profile lteprofile
 apn ltenet
#
ip route-static 0.0.0.0 0.0.0.0 Tunnel0/0/1
ip route-static 202.100.1.0 255.255.255.0 Dialer1
ip route-static 202.200.1.0 255.255.255.0 Cellular0/0/0
#
return
```

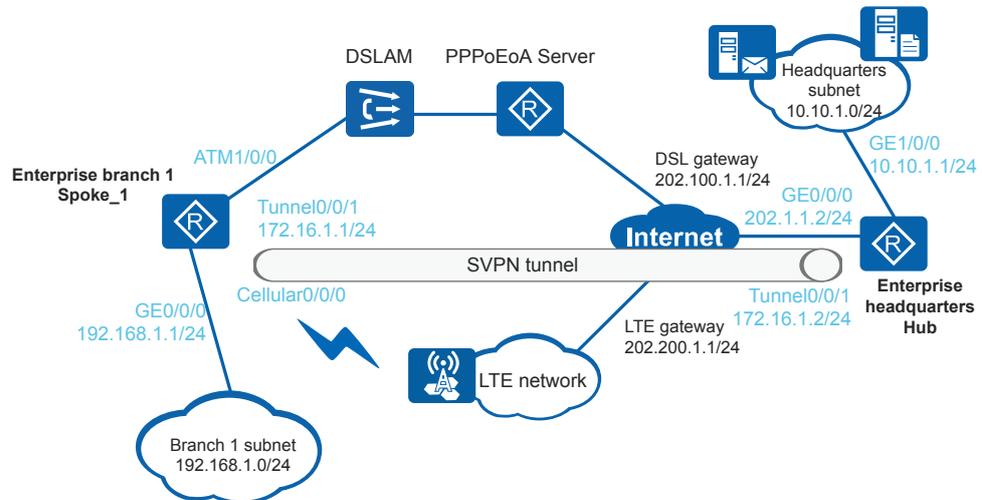
4.8.2 Example for Configuring Hub-Spoke svpn of the Priority Mode

Networking Requirements

As shown in [Figure 4-7](#), an enterprise purchased two WAN links: a DSL link with 2 Mbit/s bandwidth and an LTE link which is a yearly-package service. Users in the enterprise automatically dial up to connect to the Internet. The enterprise requires that service flows between the headquarters and branch be always transmitted over the optimal path.

Hub-Spoke SVPN of Priority mode can be configured to meet this requirement. After a service flow is imported to an SVPN tunnel, SVPN schedules the service flow in the Priority mode to fully use the optimal link.

Figure 4-7 Configuring Hub-Spoke SVPN of the Priority mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Connect the enterprise subnet to the Internet over the DSL and LTE links.
2. Configure Hub-Spoke SVPN of the Priority mode. Configure two forwarding paths (DSL link and LTE link) for a service flow and configure NQA test to ensure that the optimal link is used for communication between the headquarters and branch.

Procedure

Step 1 Connect the enterprise branch and the headquarters.

Configure a DSL link for the enterprise branch Spoke_1.

```
<Huawei> system-view
[Huawei] sysname Spoke_1
[Spoke_1] dialer-rule
[Spoke_1-dialer-rule] dialer-rule 10 ip permit
[Spoke_1-dialer-rule] quit
[Spoke_1] interface dialer 1
[Spoke_1-Dialer1] dialer user u1
[Spoke_1-Dialer1] dialer-group 10
[Spoke_1-Dialer1] dialer bundle 12
[Spoke_1-Dialer1] ip address ppp-negotiate
[Spoke_1-Dialer1] link-protocol ppp
[Spoke_1-Dialer1] ppp chap user huawei
[Spoke_1-Dialer1] ppp chap password cipher huawei@1234
[Spoke_1-Dialer1] quit
[Spoke_1] interface virtual-ethernet 0/0/0
[Spoke_1-Virtual-Ethernet0/0/0] pppoe-client dial-bundle-number 12
[Spoke_1-Virtual-Ethernet0/0/0] quit
[Spoke_1] interface atm 1/0/0
[Spoke_1-Atm1/0/0] pvc pppoeoa 2/45
[Spoke_1-atm-pvc-Atm1/0/0-2/45-pppoeoa] map bridge virtual-ethernet 0/0/0
[Spoke_1-atm-pvc-Atm1/0/0-2/45-pppoeoa] quit
[Spoke_1-Atm1/0/0] quit
```

```
[Spoke_1] ip route-static 202.100.1.0 24 dialer 1
[Spoke_1] ip route-static 202.1.1.0 24 202.100.1.1
```

For details on how to configure the DSLAM, see the DSLAM documentation.

Assign IP address 202.1.10.1 to the PPPoEoA server and configure the server to assign IP address 202.1.10.2 to the client. Set the authentication mode to CHAP authentication, and set the user name and password to be the same as those configured on the client.

Configure an LTE link for the enterprise branch Spoke_1. The enterprise obtains the following information from the carrier: the APN is ltenet, and the dialer number is *99#.

```
[Spoke_1] apn profile lteprofile
[Spoke_1-apn-profile-lteprofile] apn ltenet
[Spoke_1-apn-profile-lteprofile] quit
[Spoke_1] interface cellular 0/0/0
[Spoke_1-Cellular0/0/0] mode lte auto
[Spoke_1-Cellular0/0/0] dialer enable-circular
[Spoke_1-Cellular0/0/0] apn-profile lteprofile
[Spoke_1-Cellular0/0/0] ip address negotiate
[Spoke_1-Cellular0/0/0] dialer-group 1
[Spoke_1-Cellular0/0/0] dialer number *99# autodial
[Spoke_1-Cellular0/0/0] dialer timer autodial 20
[Spoke_1-Cellular0/0/0] shutdown
[Spoke_1-Cellular0/0/0] undo shutdown
[Spoke_1-Cellular0/0/0] quit
[Spoke_1] ip route-static 202.200.1.0 24 cellular 0/0/0
[Spoke_1] ip route-static 202.1.1.0 24 202.200.1.1
```

After the device successfully connects to the LTE network, Cellular0/0/0 obtains a dynamic IP address. Assume that the IP address is 202.1.20.2/24 in this example.

Connect the branch subnet to the enterprise branch.

```
[Spoke_1] interface gigabitethernet 0/0/0
[Spoke_1-GigabitEthernet0/0/0] ip address 192.168.1.1 24
[Spoke_1-GigabitEthernet0/0/0] quit
```

Connect the headquarters Hub to the Internet.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface gigabitethernet 0/0/0
[Hub-GigabitEthernet0/0/0] ip address 202.1.1.2 24
[Hub-GigabitEthernet0/0/0] quit
[Hub] ip route-static 0.0.0.0 0 202.1.1.1
```

Connect the headquarters subnet to the enterprise headquarters.

```
[Hub] interface gigabitethernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 10.10.1.1 24
[Hub-GigabitEthernet1/0/0] quit
```

Step 2 Configure a tunnel interface.

Configure Spoke_1.

```
[Spoke_1] interface tunnel 0/0/1
[Spoke_1-Tunnel0/0/1] ip address 172.16.1.1 24
[Spoke_1-Tunnel0/0/1] tunnel-protocol svpn p2p
[Spoke_1-Tunnel0/0/1] quit
```

Configure the hub.

```
[Hub] interface tunnel 0/0/1
[Hub-Tunnel0/0/1] ip address 172.16.1.2 24
[Hub-Tunnel0/0/1] tunnel-protocol svpn p2mp
[Hub-Tunnel0/0/1] quit
```

Step 3 Configure an SVPN Proposal of the Priority mode.

In Hub-Spoke mode, you need to set the encapsulation type for SVPN packets to GRE. To implement priority-based scheduling, you also need to configure NQA to detect link quality on the branch device.

1. Configure NQA.

Configure the hub.

```
[Hub] nqa-server udpecho 202.1.1.2 9000
```

Configure Spoke_1.

```
[Spoke_1] nqa test-instance admin dsl
[Spoke_1-nqa-admin-dsl] test-type jitter
[Spoke_1-nqa-admin-dsl] destination-address ipv4 202.1.1.2
[Spoke_1-nqa-admin-dsl] destination-port 9000
[Spoke_1-nqa-admin-dsl] frequency 15
[Spoke_1-nqa-admin-dsl] source-interface dialer1
[Spoke_1-nqa-admin-dsl] start now
[Spoke_1-nqa-admin-dsl] quit
[Spoke_1] nqa test-instance admin lte
[Spoke_1-nqa-admin-lte] test-type jitter
[Spoke_1-nqa-admin-lte] destination-address ipv4 202.1.1.2
[Spoke_1-nqa-admin-lte] destination-port 9000
[Spoke_1-nqa-admin-lte] frequency 15
[Spoke_1-nqa-admin-lte] source-interface cellular0/0/0
[Spoke_1-nqa-admin-lte] start now
[Spoke_1-nqa-admin-lte] quit
```

2. Configure an SVPN proposal.

Configure Spoke_1.

```
[Spoke_1] acl number 3001
[Spoke_1-acl-adv-3001] rule 5 permit ip source 192.168.1.0 0.0.0.255
destination 10.10.1.0 0.0.0.255
[Spoke_1-acl-adv-3001] rule 10 permit icmp
[Spoke_1-acl-adv-3001] rule 15 permit ospf
[Spoke_1-acl-adv-3001] quit
[Spoke_1] svpn-proposal p1
[Spoke_1-svpn-proposal-p1] authentication user-name Huawei password cipher
Huawei@1234
[Spoke_1-svpn-proposal-p1] encapsulation gre
[Spoke_1-svpn-proposal-p1] source dialer1 destination 202.1.1.2 track nqa
admin dsl
[Spoke_1-svpn-proposal-p1] source cellular0/0/0 destination 202.1.1.2 track
nqa admin lte
[Spoke_1-svpn-proposal-p1] service s1 id 1
[Spoke_1-svpn-proposal-p1-service-s1] schedule-type priority
[Spoke_1-svpn-proposal-p1-service-s1] match acl 3001
[Spoke_1-svpn-proposal-p1-service-s1] source dialer1
[Spoke_1-svpn-proposal-p1-service-s1] source cellular0/0/0
[Spoke_1-svpn-proposal-p1-service-s1] quit
[Spoke_1-svpn-proposal-p1] recovery-delay 50
[Spoke_1-svpn-proposal-p1] quit
```

Configure the hub.

```
[Hub] acl number 3001
[Hub-acl-adv-3001] rule 5 permit ip source 10.10.1.0 0.0.0.255 destination
192.168.1.0 0.0.0.255
[Hub-acl-adv-3001] rule 10 permit icmp
[Hub-acl-adv-3001] rule 15 permit ospf
[Hub-acl-adv-3001] quit
[Hub] svpn-proposal p1
[Hub-svpn-proposal-p1] authentication user-name Huawei password cipher
Huawei@1234
```

```
[Hub-svpn-proposal-p1] encapsulation gre
[Hub-svpn-proposal-p1] service s1 id 1
[Hub-svpn-proposal-p1-service-s1] match acl 3001
[Hub-svpn-proposal-p1-service-s1] quit
[Hub-svpn-proposal-p1] quit
```

Step 4 Bind the SVPN proposal to the tunnel interface.

In Hub-Spoke mode, SVPN zone for the tunnel interface of the hub must be the same as that of the spoke.

Configure Spoke_1.

```
[Spoke_1] interface tunnel 0/0/1
[Spoke_1-Tunnel0/0/1] svpn-zone 1
[Spoke_1-Tunnel0/0/1] svpn-proposal p1
[Spoke_1-Tunnel0/0/1] quit
```

Configure the hub.

```
[Hub] interface tunnel 0/0/1
[Hub-Tunnel0/0/1] svpn-zone 1
[Hub-Tunnel0/0/1] svpn-proposal p1
[Hub-Tunnel0/0/1] quit
```

Step 5 Import service flows to an SVPN tunnel.

Configure OSPF on the headquarters and branch devices to advertise branch subnet routes to the remote end.

Configure Spoke_1.

```
[Spoke_1] ospf 1
[Spoke_1-ospf-1] area 0.0.0.0
[Spoke_1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke_1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke_1-ospf-1-area-0.0.0.0] quit
[Spoke_1-ospf-1] quit
[Spoke_1] interface tunnel 0/0/1
[Spoke_1-Tunnel0/0/1] ospf network-type broadcast
[Spoke_1-Tunnel0/0/1] ospf dr-priority 0
[Spoke_1-Tunnel0/0/1] quit
```

Configure the hub.

```
[Hub] ospf 1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 10.10.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
[Hub] interface tunnel 0/0/1
[Hub-Tunnel0/0/1] ospf network-type broadcast
[Hub-Tunnel0/0/1] ospf dr-priority 100
[Hub-Tunnel0/0/1] quit
```

Step 6 Verify the configuration.

After the configurations are complete, run the **display ip routing-table** command on Spoke_1. You can see that routes destined for the headquarters pass through the SVPN tunnel.

Connect Port1 on the tester to Spoke_1 and Port2 on the tester to the hub. Inject a service flow from Port1 to Spoke_1 (the source and destination addresses of the service flow are 192.168.1.100/24 and 10.10.1.10/24). Check the NQA test results on the DSL and LTE links. You can find that the service flow is always transmitted over the path with a lower jitter.

----End

Configuration Files

- Configuration file of Spoke_1

```
#
sysname Spoke_1
#
acl number 3001
 rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 10.10.1.0
 0.0.0.255
 rule 10 permit icmp
 rule 15 permit ospf
#
interface Dialer1
 link-protocol ppp
 ppp chap user huawei
 ppp chap password cipher
%^%#'&=6Q(|7-#|. )EB`mK$(h7 [CY`2m)-YT)Q=Oh2~2%^%#
 ip address ppp-negotiate
 dialer user ul
 dialer bundle 12
 dialer-group 10
#
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0
#
interface Cellular0/0/0
 dialer enable-circular
 dialer-group 1
 apn-profile lteprofile
 dialer timer autodial 20
 dialer number *99# autodial
 ip address negotiate
#
interface Atm1/0/0
 pvc pppoea 2/45
 map bridge Virtual-Ethernet0/0/0
#
interface Virtual-Ethernet0/0/0
 pppoe-client dial-bundle-number 12
#
svpn-proposal p1
 encapsulation gre
 recovery-delay 50
 source Dialer1 destination 202.1.1.2 track nqa admin dsl
 source Cellular0/0/0 destination 202.1.1.2 track nqa admin lte
 authentication user-name Huawei password cipher %@%@] )&a,3$|5@WjAd8V+tRV,.R$
%@%@
 service s1 id 1
  schedule-type priority
  match acl 3001
  source Dialer1
  source Cellular0/0/0
#
interface Tunnel0/0/1
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol svpn p2p
 ospf network-type broadcast
 ospf dr-priority 0
 svpn-zone 1
 svpn-proposal p1
#
dialer-rule
 dialer-rule 10 ip permit
#
apn profile lteprofile
 apn ltenet
#
ospf 1
 area 0.0.0.0
```

```
network 172.16.1.0 0.0.0.255
network 192.168.1.0 0.0.0.255
#
ip route-static 202.1.1.0 255.255.255.0 202.200.1.1
ip route-static 202.1.1.0 255.255.255.0 202.100.1.1
ip route-static 202.100.1.0 255.255.255.0 Dialer1
ip route-static 202.200.1.0 255.255.255.0 Cellular0/0/0
#
nqa test-instance admin dsl
test-type jitter
destination-address ipv4 202.1.1.2
destination-port 9000
frequency 15
source-interface Dialer1
start now
nqa test-instance admin lte
test-type jitter
destination-address ipv4 202.1.1.2
destination-port 9000
frequency 15
source-interface Cellular0/0/0
start now
#
return
```

● Configuration file of the hub

```
#
sysname Hub
#
acl number 3001
rule 5 permit ip source 10.10.1.0 0.0.0.255 destination 192.168.1.0
0.0.0.255
rule 10 permit icmp
rule 15 permit ospf
#
interface GigabitEthernet0/0/0
ip address 202.1.1.2 255.255.255.0
#
interface GigabitEthernet1/0/0
ip address 10.10.1.1 255.255.255.0
#
svpn-proposal p1
encapsulation gre
authentication user-name Huawei password cipher %@%@ZiE!m99L#RV]*'T=!63,&i+
%@%@
service s1 id 1
match acl 3001
#
interface Tunnel0/0/1
ip address 172.16.1.2 255.255.255.0
tunnel-protocol svpn p2mp
ospf network-type broadcast
ospf dr-priority 100
svpn-zone 1
svpn-proposal p1
#
ospf 1
area 0.0.0.0
network 10.10.1.0 0.0.0.255
network 172.16.1.0 0.0.0.255
#
nqa-server udpecho 202.1.1.2 9000
#
ip route-static 0.0.0.0 0.0.0.0 202.1.1.1
#
return
```

5 DSVPN Configuration

About This Chapter

- [5.1 Overview of DSVPN](#)
- [5.2 Understanding DSVPN](#)
- [5.3 Application Scenarios for DSVPN](#)
- [5.4 Licensing Requirements and Limitations for DSVPN](#)
- [5.5 Default Settings for DSVPN](#)
- [5.6 Configuring DSVPN](#)
- [5.7 Maintaining DSVPN](#)
- [5.8 Configuration Examples for DSVPN](#)
- [5.9 Troubleshooting DSVPN](#)
- [5.10 References for DSVPN](#)

5.1 Overview of DSVPN

Definition

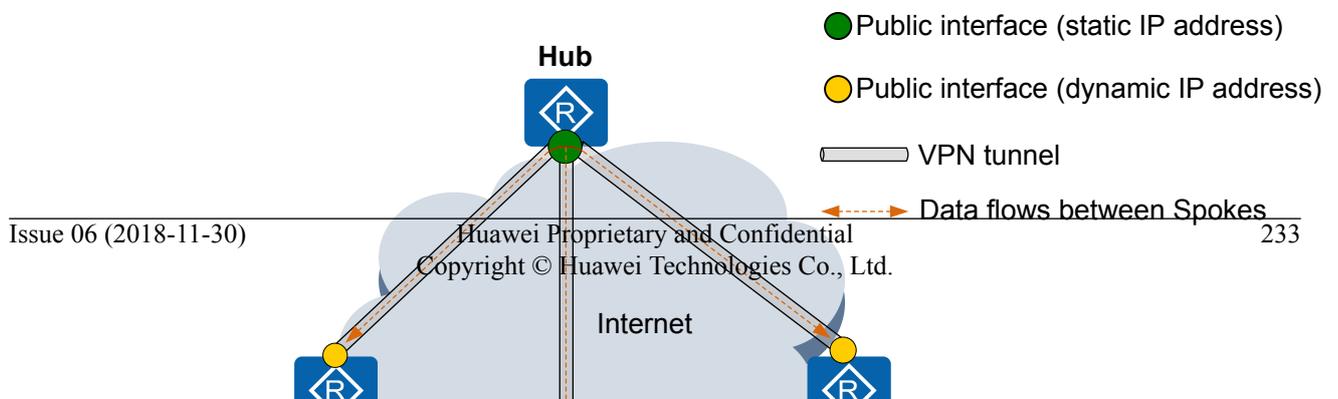
Dynamic Smart Virtual Private Network (DSVPN) establishes VPN tunnels between Spokes with dynamically variable public addresses in the Hub-Spoke model.

Purpose

More enterprises want to build the IPsec VPN in Hub-Spoke model to connect the Hub to Spokes in different geographical locations. This enhances enterprise communication security and reduces communication costs. When the Hub uses the static public address to connect to the Internet and Spokes use dynamic public addresses to connect to the Internet, Spokes cannot communicate with each other directly if traditional IPsec or GRE over IPsec is used to build the VPN. This is because Spokes cannot learn the public addresses of the remote ends in

advance and tunnels cannot be set up between Spokes. In this case, communication data between Spokes must be forwarded by the Hub. **Figure 5-1** shows the networking.

Figure 5-1 Typical Hub-Spoke networking without DSVPN enabled



When all communication data between Spokes is forwarded by the Hub, the following problems may occur:

- The Hub consumes many CPU and memory resources to forward data flows between Spokes, causing resource shortage.
- The Hub needs to encapsulate and decapsulate data flows between Spokes, which causes extra delay.

When the IPsec network scale increases continuously, dynamic routing protocols need to be deployed to reduce route configuration and maintenance. There is a common issue when IPsec and dynamic routing protocols are deployed. Dynamic routing protocols use multicast or broadcast packets for route update, whereas IPsec does not support transmission of broadcast and multicast packets.

Huawei proposes the DSVPN solution that integrates the Next Hop Resolution Protocol (NHRP) and Multipoint Generic Routing Encapsulation (mGRE) with IPsec to solve the preceding issue.

- DSVPN uses NHRP to dynamically collect, maintain, and advertise dynamic public network addresses of nodes, and allows the source Spoke to obtain the public network address of the destination Spoke so that a dynamic VPN tunnel can be set up between Spokes. Spokes can communicate with each other directly, reducing the burden of the Hub and preventing the network delay.
- DSVPN uses mGRE technology to transmit multicast and broadcast packets over VPN tunnels that can be established between one tunnel interface and multiple remote devices, reducing the tunnel configuration workload. In addition, when a Spoke is added or the public address of a Spoke changes, DSVPN automatically maintains the tunnel between the Hub and Spoke. There is no need to change the tunnel configuration of the Hub, and DSVPN facilitates network maintenance.

Figure 5-2 shows a VPN using DSVPN.

Figure 5-2 Hub-Spoke networking enabled with DSVPN

Benefits

- Reduce costs on VPN construction.
DSVPN implements dynamic connections between the Hub and Spokes, and between Spokes. Spokes do not need to purchase static public network addresses.
- Simplify configuration on the Hub and Spokes.
The Hub and Spokes use an mGRE tunnel interface but not multiple GRE tunnel interfaces to establish tunnels. When a new Spoke is added to the network, the network administrator does not need to change configurations on the Hub or any existing Spokes. The administrator only needs to configure the new Spoke, and then the Spoke dynamically registers with the Hub.
- Reduce the forwarding delay between Spokes.
Spokes can dynamically establish tunnels to directly exchange service data, reducing the forwarding delay and improving forwarding performance and efficiency.

5.2 Understanding DSVPN

5.2.1 Basic Concepts

Figure 5-3 shows the typical network architecture of the DSVPN solution. An enterprise connects the **Hub** to **Spokes** in different geographical locations through the public network. The Hub uses the static public address, and Spokes use dynamic public addresses.

On the network, when the source Spoke sends data packets to the destination Spoke, the source Spoke obtains the public address of the destination Spoke by exchanging **NHRP packets** over the **static mGRE tunnel** between itself and the Hub and establishes a **dynamic mGRE tunnel** with the destination Spoke. After the tunnel is set up, data packets between Spokes are sent to the remote end over the dynamic mGRE tunnel, but are not forwarded by the Hub.

Figure 5-3 Typical enterprise networking

Subnet

DSVPN Node

DSVPN involves the following entities:

A DSVPN node is a device on which DSVPN is deployed, which can be a Spoke or Hub.

- **Spoke**

A Spoke is the network gateway of a branch. Generally, a Spoke uses a dynamic public network address.

- **Hub**

A Hub is the gateway in the headquarters and receives registration packets from Spokes. On a DSVPN network, the Hub can use a fixed public network address or a domain name.

mGRE, mGRE Tunnel Interface, and mGRE Tunnel

mGRE is a point-to-multipoint GRE technology developed based on GRE. It extends traditional P2P tunnel interfaces to P2MP mGRE tunnel interfaces. One tunnel interface can be used to establish tunnels with multiple remote devices by changing the interface type. Therefore, only one tunnel interface needs to be configured on the Hub or Spoke, reducing the GRE tunnel configuration workload.

The mGRE tunnel interface has the following attributes:

- Source tunnel address: is the source address of a GRE encapsulated packet, that is, public network address of one end in [Figure 5-3](#).
- Destination tunnel address: is the destination address of a GRE encapsulated packet, that is, public network address of the other end in [Figure 5-3](#). This address is based on [NHRP](#), which is different from the manually specified destination address of the GRE tunnel interface.
- Tunnel interface IP address: is the tunnel address in [Figure 5-3](#). Similar to an IP address of a physical interface, a tunnel interface IP address is used for communication between devices, for example, routing information is obtained.

 **NOTE**

mGRE tunnel interfaces do not support keepalive detection of the GRE interface.

A GRE tunnel established using an mGRE tunnel interface is called an mGRE tunnel. mGRE tunnels fall into static and dynamic mGRE tunnels:

- A static mGRE tunnel is set up between a Spoke and the Hub and always exists. The Spoke sends registration packets to the Hub periodically. When receiving a registration packet a Spoke, the Hub resets the aging timer of the matching NHRP mapping entry to maintain the tunnel with the Spoke.
- A dynamic mGRE tunnel is established between Spokes. It is automatically torn down if no packet is forwarded through it within a period.

NHRP and NHRP Mapping Entry

On a DSVPN network, NHRP is used to establish and resolve the mapping between the protocol address (tunnel address in [Figure 5-3](#) or subnet address) and Non-Broadcast Multiple Access (NBMA) address (public address in [Figure 5-3](#)). By doing this, the source Spoke can obtain the dynamic public address of the destination Spoke.

The NHRP mapping table contains the entries that are generated based on the mapping between protocol addresses and NBMA addresses. NHRP entries fall into static and dynamic entries based on the entry generation mode:

- **Static NHRP mapping entry:** is manually configured by an administrator. When a Spoke needs to establish a static mGRE tunnel with the Hub, the administrator needs to manually configure the tunnel address and public network address of the Hub on the Spoke.
- **Dynamic NHRP mapping entry:** is dynamically generated by NHRP. For example, the Hub obtains the tunnel address and public address of each Spoke from an NHRP Registration packet and generates an NHRP mapping table. Each Spoke obtains the tunnel address or subnet address and public address of the remote Spoke from an NHRP Resolution packet and generates an NHRP mapping table.

For details about NHRP Registration and Resolution packets, see RFC 2332.

5.2.2 Implementation

DSVPN establishes tunnels based on mGRE and NHRP to implement direct communication between Spokes. Unlike GRE, mGRE does not need to define the **destination tunnel address** during tunnel setup. Instead, mGRE obtains the destination tunnel address through NHRP, making it possible to set up tunnels between Spokes with dynamic addresses.

When the device forwards an IP packet, it sends the IP packet to the mGRE tunnel interface based on the routing table. mGRE queries and obtains the remote public address mapping the next-hop address in the **NHRP mapping table**. mGRE adds a new IP header to the IP packet in which the destination address is the remote public address. Then the IP packet can be sent to the remote end over the tunnel.

The NHRP mapping table and routing table are the basis for tunnel setup when mGRE and NHRP are deployed. If a Spoke has a route to the remote Spoke and the NHRP mapping entry between the tunnel address or subnet address of the remote Spoke and public address, an mGRE tunnel can be set up between Spokes. At the beginning, a Spoke has only the route to the Hub and one static NHRP mapping entry between the tunnel address of the Hub and public address. Spokes cannot establish tunnels directly. They have to learn routes to each other through the Hub and generate NHRP mapping entries between tunnel addresses or subnet addresses and public addresses. There are three processes:

1. **Establishing mGRE Tunnels Between Spokes and the Hub**

After mGRE tunnels are set up between Spokes and the Hub, packets of one Spoke can be forwarded to the remote Spoke through the Hub.

DSVPN establishes a static mGRE tunnel between a Spoke and the Hub. This tunnel always exists regardless of whether there is traffic between the Spoke and Hub.

2. **Learning Routes Between Spokes**

The route from one Spoke to the remote Spoke is generated.

3. **Establishing mGRE Tunnels Between Spokes**

mGRE tunnels are set up so that Spokes can communicate directly. When one Spoke forwards data packets to the other Spoke and the source Spoke cannot find the public address of the destination Spoke, DSVPN establishes an mGRE tunnel between Spokes.

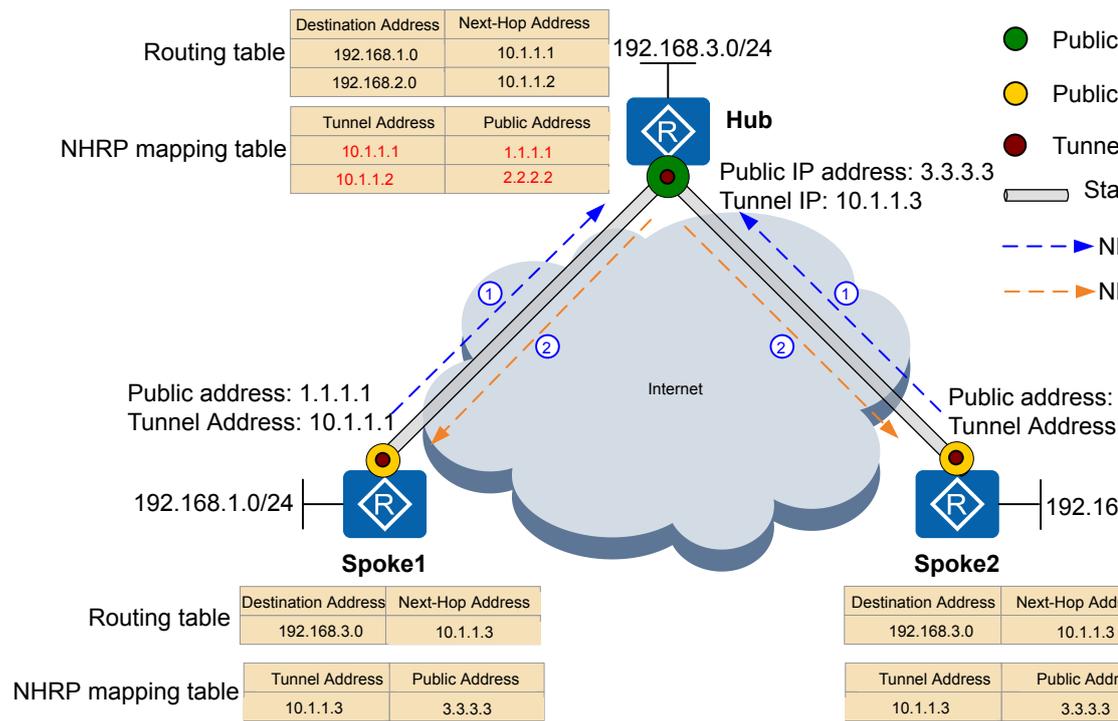
The mGRE tunnel between Spokes is a dynamic tunnel. When there is traffic between Spokes, the tunnel keeps connected automatically. When there is no traffic between Spokes during a period of time, the tunnel is terminated automatically.

When an mGRE tunnel is established between Spokes, data packets between Spokes are directly forwarded over this mGRE tunnel and do not pass through the Hub.

Establishing mGRE Tunnels Between Spokes and the Hub

At the beginning, the Hub has no NHRP mapping entry, and the Spoke has the route to the Spoke and a static mapping entry between the tunnel address of the Hub and public address. To establish mGRE tunnels between Spokes and the Hub, the Hub needs to generate NHRP mapping entries between tunnel addresses of the Spokes and public addresses. The Spokes initiate NHRP registration to the Hub so that entries can be generated. [Figure 5-4](#) shows the process.

Figure 5-4 Establishing mGRE tunnels between Spokes and the Hub



1. Spokes send NHRP Registration Request packets to the Hub.
After the administrator manually configures the Hub's tunnel address and public address on the Spokes, the Spokes periodically send NHRP Registration Request packets to the Hub. The packets carry the Spokes' tunnel addresses and public addresses.
2. The Hub responds to NHRP Registration Request packets of the Spokes.
The Hub obtains the Spokes' tunnel addresses and public addresses from NHRP Registration Request packets (fonts in red in [Figure 5-4](#)), generates NHRP mapping entries, and establishes mGRE tunnels.

The Spokes periodically send NHRP Registration Request packets to the Hub. When receiving a registration packet from a Spoke, the Hub resets the aging timer of the matching NHRP mapping entry to maintain the tunnel with the Spoke.

Learning Routes Between Spokes

DSVPN supports the following route learning modes:

- **Route learning between Spokes (non-shortcut mode)**
The next-hop address of the route from the source Spoke to the destination Spoke is the tunnel address of the destination Spoke (see the routing table in [Figure 5-5](#)), and each Spoke needs to learn the route to the remote end. This consumes many CPU and memory resources and requires large routing tables and high performance on Spokes. In practice, the Spokes have low performance and store a limited number of routes. The route learning solution applies to small- and medium-sized networks where there are fewer network nodes and a small number of routes.
- **Spoke routes summarized to the Hub (shortcut mode)**
The next-hop address of the route from the source Spoke to the destination Spoke is the tunnel address of the Hub (see the routing table in [Figure 5-6](#)), and Spokes only need to store routes to the Hub. The number of routes of Spokes is reduced, so the route learning solution applies to large-sized networks with many Spokes.

Establishing mGRE Tunnels Between Spokes

After the preceding two processes are complete, each Spoke has the route to the remote Spoke, but has no NHRP mapping entry between the destination Spoke's tunnel address or subnet address and public address. To establish mGRE tunnels between Spokes, NHRP is used to generate NHRP mapping entries based on learned routes. When different route learning modes are used, Spokes learn different routes and NHRP mapping entry generation processes are also different:

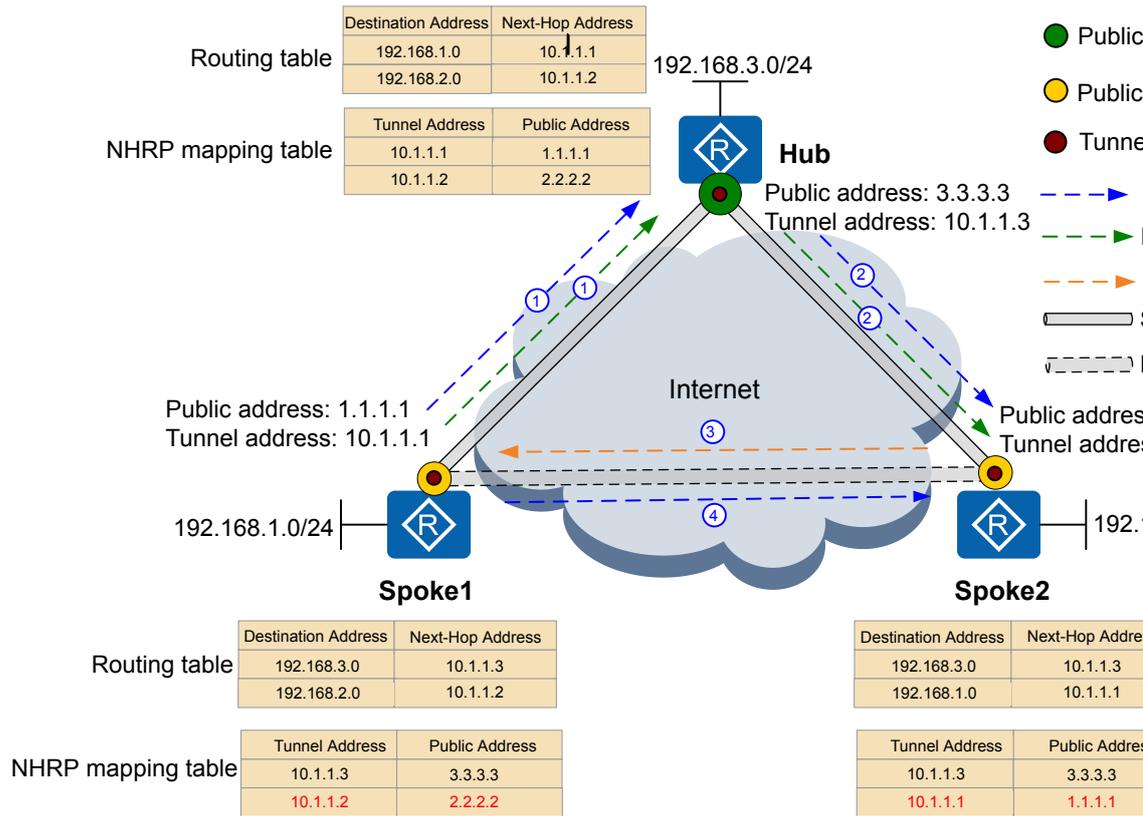
- **Non-shortcut:** The source Spoke can learn the tunnel address of the destination Spoke. The source Spoke can query the destination Spoke's public address based on destination Spoke's tunnel address and generate the NHRP mapping entry between the destination Spoke's tunnel address and public address.
- **Shortcut:** Next-hop addresses of all Spokes are the Hub's tunnel address, and the source Spoke cannot learn the tunnel address of the destination Spoke. The source Spoke can query the destination Spoke's public address based on destination address of packets and generate the NHRP mapping entry between the destination Spoke's subnet address and public address.

[Figure 5-5](#) and [Figure 5-6](#) describe the processes.

Establishing an mGRE tunnel between Spokes in non-shortcut mode

Figure 5-5 shows the mGRE tunnel setup between Spokes in non-shortcut mode.

Figure 5-5 Establishing an mGRE tunnel between Spokes in non-shortcut mode



When a user on Spoke1 first accesses a user on Spoke2, the setup of a dynamic mGRE tunnel between Spoke1 and Spoke2 is triggered. The tunnel setup process is as follows.

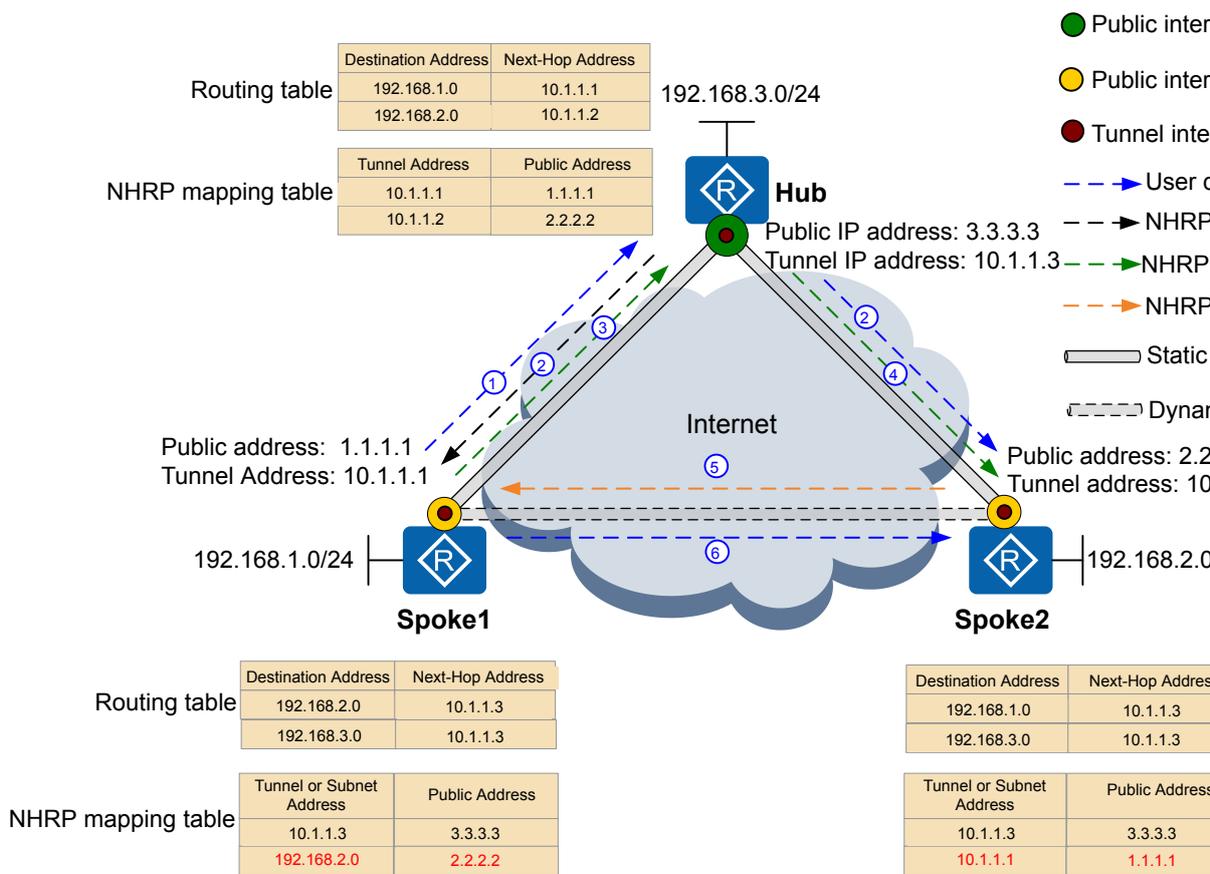
1. After Spoke1 receives a data packet destined for Spoke2:
 - Spoke1 finds next-hop address 10.1.1.2 (tunnel address of Spoke2) based on destination address 192.168.2.0 of the data packet, but no NHRP mapping entry defines the public address mapping 10.1.1.2. Spoke1 directly forwards the data packet to the Hub by default.
 - Spoke1 constructs and sends an NHRP Resolution Request packet to the Hub, requesting for the public address mapping 10.1.1.2.
2. After the Hub receives the data packet and NHRP Resolution Request packet from Spoke1, it forwards the packets to Spoke2 through the mGRE tunnel between them.
3. After Spoke2 receives the NHRP Resolution Request packet:
 - Spoke2 obtains the tunnel address and public address of Spoke1 from the NHRP Resolution Request packet, and updates such information in its NHRP mapping table (see fonts in red in [Figure 5-5](#)).
 - Spoke2 constructs and sends an NHRP Resolution Reply packet that carries tunnel address 10.1.1.2 and public address 2.2.2.2 of Spoke2.
4. After Spoke1 receives the NHRP Resolution Reply packet, it obtains the tunnel address and public address of Spoke2 from the NHRP Resolution Reply packet, and updates its NHRP mapping table (see fonts in red in [Figure 5-5](#)). A dynamic mGRE tunnel between Spoke1 and Spoke2 is set up.

When Spoke1 receives a data packet destined for Spoke2 again, it finds next-hop address 10.1.1.2 in the routing table based on destination address 192.168.2.0 of the data packet and public address 2.2.2.2 mapping 10.1.1.2 in the NHRP mapping table. Then Spoke1 adds an mGRE header to the data packet based on public address 2.2.2.2 and directly forwards it to Spoke2.

Establishing an mGRE tunnel between Spokes in shortcut mode

[Figure 5-6](#) shows the mGRE tunnel setup between Spokes in shortcut mode.

Figure 5-6 Establishing an mGRE tunnel between Spokes in shortcut mode



When a user on Spoke1 first accesses a user on Spoke2, the setup of a dynamic mGRE tunnel between Spoke1 and Spoke2 is triggered. The tunnel setup process is as follows.

1. When Spoke1 receives a data packet destined for Spoke2, it finds next-hop address 10.1.1.3 (tunnel address of the Hub) based on destination address 192.168.2.0 of the data packet and public address 3.3.3.3 mapping 10.1.1.3 (public address of the Hub) in the NHRP mapping table. Then Spoke1 forwards the data packet to the Hub.
2. After the Hub receives the data packet forwarded by Spoke1:
 - The Hub forwards the data packet to Spoke2 over the mGRE tunnel between the Hub and Spoke2.
 - The Hub finds that the tunnel interfaces receiving and sending the data packet belong to the same NHRP domain (see `nhrp network-id`). It constructs and sends an NHRP Redirect packet to Spoke1. The packet carries only the tunnel address and public address of the Hub.
3. After Spoke1 receives the NHRP Redirect packet, it constructs and sends an NHRP Resolution Request packet to the Hub. The packet carries tunnel address 10.1.1.1 and public address 1.1.1.1 of Spoke1, and destination address 192.168.2.0 of the data packet to be resolved.
4. The Hub forwards the received NHRP Resolution Request packet to Spoke2.
5. After Spoke2 receives the NHRP Resolution Request packet:
 - Spoke2 obtains the subnet address and public address of Spoke1 from the NHRP Resolution Request packet, and updates its NHRP mapping table (see fonts in red in [Figure 5-6](#)).

- Spoke2 constructs and sends an NHRP Resolution Reply packet that carries the subnet address 192.168.2.0, tunnel address 10.1.1.2, and public address 2.2.2.2 of Spoke2.
6. After Spoke1 receives the NHRP Resolution Reply packet, it obtains the subnet address and public address of Spoke2 from the NHRP Resolution Reply packet, and updates its NHRP mapping table (see fonts in red in [Figure 5-6](#)). A dynamic mGRE tunnel between Spoke1 and Spoke2 is set up.

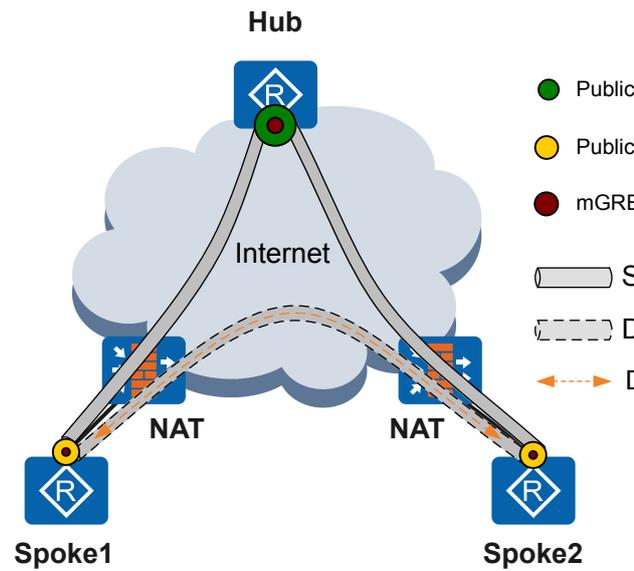
When Spoke1 receives a data packet destined for Spoke2 again, it searches the NHRP mapping table for Spoke2's public address 2.2.2.2 based on destination address 192.168.2.0 of the data packet. Then Spoke1 adds an mGRE header to the data packet based on public address 2.2.2.2 and directly forwards it to Spoke2.

When a user on Spoke2 first accesses a user on Spoke1, the setup of a dynamic mGRE tunnel between Spoke2 and Spoke1 is also triggered. The tunnel setup process is similar to that when a user on Spoke1 first accesses a user on Spoke2.

5.2.3 DSVPN NAT Traversal

In [Figure 5-7](#), when private networks of Spokes connect to the Hub through Network Address Translation (NAT), NAT traversal must be implemented to establish VPN tunnels between the Hub and Spokes, and between Spokes. DSVPN NAT traversal can be deployed so that Spokes can directly communicate across the NAT device.

Figure 5-7 DSVPN NAT traversal



DSVPN NAT traversal is implemented by encapsulating original and translated addresses of Spokes in NAT extension fields of NHRP Registration Reply packets and NHRP Resolution Request or Reply packets. The implementation is as follows:

1. The Spokes send NHRP Registration Request packets to the Hub. The NHRP Registration Request packets carry public or private network addresses of the Spokes.
2. NHRP on the Hub detects whether a NAT device exists between the Hub and Spokes. If the NAT device exists, the Hub encapsulates translated public addresses of Spokes in NAT extension fields of NHRP Registration Reply packets and sends the packets to the Spokes.
3. The source Spoke sends an NHRP Resolution Request packet to the destination Spoke. The packet carries the original address and translated public address in NAT extension fields of the source Spoke.
4. The destination Spoke sends an NHRP Resolution Reply packet to the source Spoke. The packet carries the original address and translated public address in NAT extension fields of the destination Spoke.
5. The source and destination Spokes learn the original address and translated public network address of each other and establish an mGRE tunnel based on the translated public address. By doing this, Spokes can directly communicate across the NAT device.

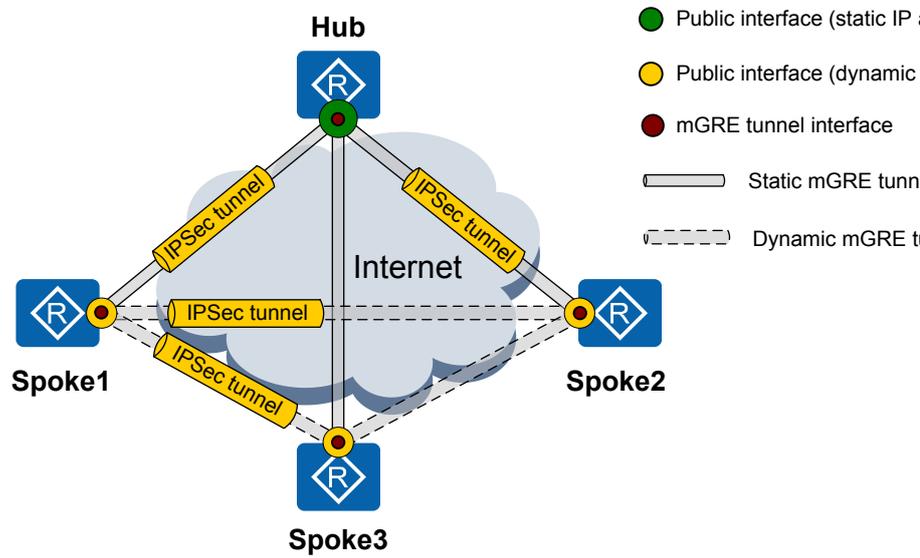
 **NOTE**

- NAT traversal cannot be implemented on a DSVPN network if two Spokes use the same NAT device and their original addresses are translated to the same public network address.
- NAT traversal cannot be implemented if two Spokes are behind different NAT devices, and Port Address Translation (PAT) is enabled on the NAT device.
- When branches need to communicate with each other, the NAT device must be configured with an NAT server or static NAT. NAT traversal cannot be implemented if outbound NAT is configured on the NAT device.
- When you deploy IPSec on a DSVPN network, the IPSec encapsulation mode can only be transport if two branches are connected to different NAT devices or the headquarters is connected to a NAT device.

5.2.4 DSVPN Protected by IPSec

DSVPN uses an mGRE tunnel to transmit data, but data is not encrypted over the mGRE tunnel and data transmission on the Internet is insecure. You are advised to deploy IPSec to ensure secure communication data transmission between Spokes when DSVPN is used.

Figure 5-8 DSVPN protected by IPSec



On a DSVPN network, IPsec profiles are configured on the Hub and Spokes and bound to mGRE tunnel interfaces. mGRE tunnel setup will trigger IPsec tunnel setup. The implementation is as follows:

1. All the Spokes on the network send NHRP Registration Request packets to the Hub and report the NHRP mapping entries to IPsec. The Internet Key Exchange (IKE) modules of the Spokes and the Hub negotiate with each other for IPsec tunnel parameters.
2. The Hub generates local NHRP mapping entries between tunnel addresses and public network addresses of the Spokes based on the NHRP Registration Request packets received. The Hub then sends NHRP Registration Reply packets to the Spokes.
3. The Spokes trigger an mGRE tunnel immediately when they transmit traffic. For details about how to establish an mGRE tunnel, see [Establishing mGRE Tunnels Between Spokes](#).
4. After the Spokes establish an mGRE tunnel, the IPsec module obtains NHRP mapping entries, adds or deletes IPsec peers based on the mapping entries, and triggers the Spokes to dynamically establish an IPsec tunnel.
5. After an IPsec tunnel is established between the Spokes, packets are routed based on the destination IP addresses. If the outbound interface is an mGRE interface, the Spoke searches the NHRP mapping table for the public network address mapping the next hop private address. After obtaining the public network address, the Spoke searches for the IPsec security association (SA) matching the public network address to encrypt the packets and send them.

Compared with IPsec in traditional Hub-Spoke networking, integrating DSVPN and IPsec has the following advantages:

- Traditional IPsec uses ACLs to identify unicast traffic to be encrypted. The ACL configuration is complex and its maintenance is difficult. In DSVPN scenarios, you only need to bind mGRE tunnel interfaces to IPsec profiles, without defining complex ACLs. The network deployment is more simple.
- Because an IPsec tunnel is dynamically established between Spokes, IPsec packets transmitted between Spokes are not decrypted or encrypted by the Hub. This shortens the packet forwarding delay.

 **NOTE**

When you deploy IPsec on a DSVPN network, the IPsec encapsulation mode can only be transport if two branches are connected to different NAT devices or the headquarters is connected to a NAT device.

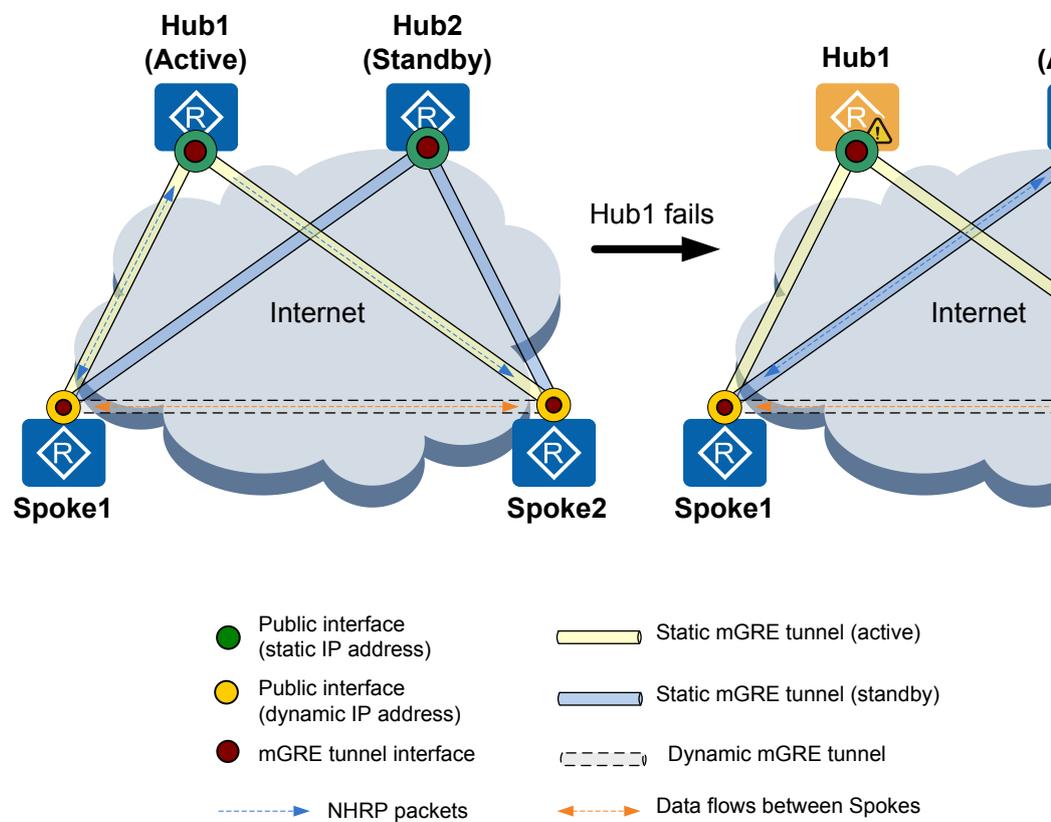
5.2.5 DSVPN Reliability

5.2.5.1 Dual Hubs in Active/Standby Mode

In basic DSVPN networking, all Spokes are connected to one Hub. Spokes cannot communicate with each other if the Hub fails. Multiple Hubs can be deployed to improve network reliability.

In [Figure 5-9](#), Hub1 and Hub2 are deployed in the headquarters. Routing policies are deployed on Spokes so that routes to Hub1 have a higher priority than those to Hub2. Normally, Hub1 is the active Hub and Hub2 is the standby Hub.

Figure 5-9 Dual Hubs in active/standby mode



The working mechanism is as follows:

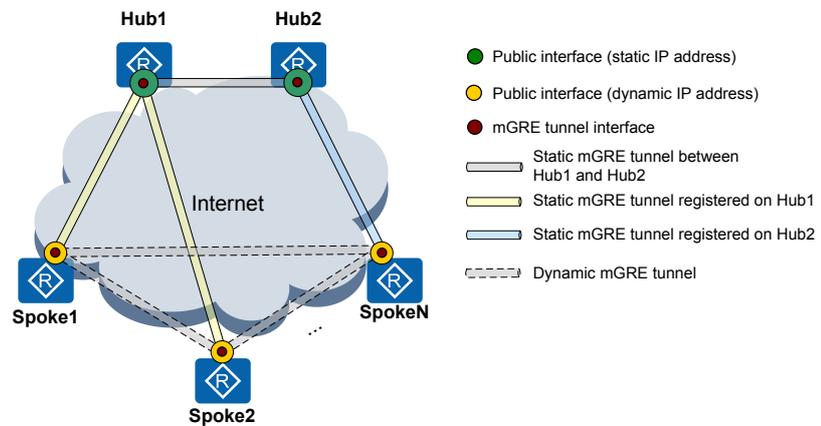
1. All Spokes are registered with Hub1 and Hub2, and establish active and standby static mGRE tunnels with Hub1 and Hub2 respectively.
2. The source Spoke sends an NHRP Resolution Request packet to request the public address of the destination Spoke, which is used to establish a dynamic mGRE tunnel.
 - When Hub1 and Hub2 work properly, the NHRP Resolution Request packet is sent to the Hub over the active static mGRE tunnel because the route from the source Spoke to Hub1 has a higher priority. Hub2 forwards the NHRP Resolution Request packet to the destination Spoke.
 - When Hub1 fails, the priority of the route from the source Spoke to Hub1 is reduced and the NHRP Resolution Request packet is sent to Hub2 over the standby static mGRE tunnel. Hub2 then sends the NHRP Resolution Request packet to the destination Spoke.
 - When Hub1 recovers, the NHRP Resolution Request packet is forwarded by Hub1. This is because the route from each Spoke to Hub1 has a higher priority than the route from each Spoke to Hub2.
3. The destination Spoke sends an NHRP Resolution Reply packet to the source Spoke, and a dynamic mGRE tunnel is set up.
4. After the dynamic mGRE tunnel is set up, Spokes can directly communicate with each other. In this case, the Hub running status does not affect service flows between Spokes. If the dynamic mGRE tunnel between branch Spokes is torn down because no traffic passes through the tunnel for a long period of time, the Spokes need to reestablish a dynamic mGRE tunnel. The Spokes then determine the Hub to which they send NHRP Resolution Request packets based on the route priority.

5.2.5.2 Dual Hubs in Load Balancing Mode

A single Hub can connect to a certain number of Spokes due to its performance limitation. When there are many Spokes on a network, you can deploy two or more Hubs to improve the processing capability of the headquarters.

In [Figure 5-10](#), Hub1 and Hub2 are deployed in the headquarters. Not all Spokes can register with one Hub because there are many Spokes, so some Spokes are registered with Hub1 and Hub2 to implement load balancing.

Figure 5-10 DSVPN load balancing



The principle of direct communication between Spokes connected to the same Hub is similar to [Principles](#). Static mGRE tunnels need to be set up between Hubs to allow Spokes connected to different Hubs to directly communicate with each other.

The direct communication process between Spokes connected to different Hubs is as follows:

1. A source Spoke1 sends an NHRP Resolution Request packet to Hub1 to request the public network address of destination SpokeN.
2. Hub1 forwards the NHRP Resolution Request packet to Hub2 over the static mGRE tunnel between Hub1 and Hub2.
3. Hub2 forwards the NHRP Resolution Request packet to the destination SpokeN.
4. The destination SpokeN obtains the public network address of the source Spoke1 from the NHRP Resolution Request packet and sends an NHRP Resolution Reply packet to the source Spoke1.
5. The source Spoke1 obtains the public network address of the destination SpokeN from the NHRP Resolution Reply packet and establishes a dynamic mGRE tunnel with the destination SpokeN.

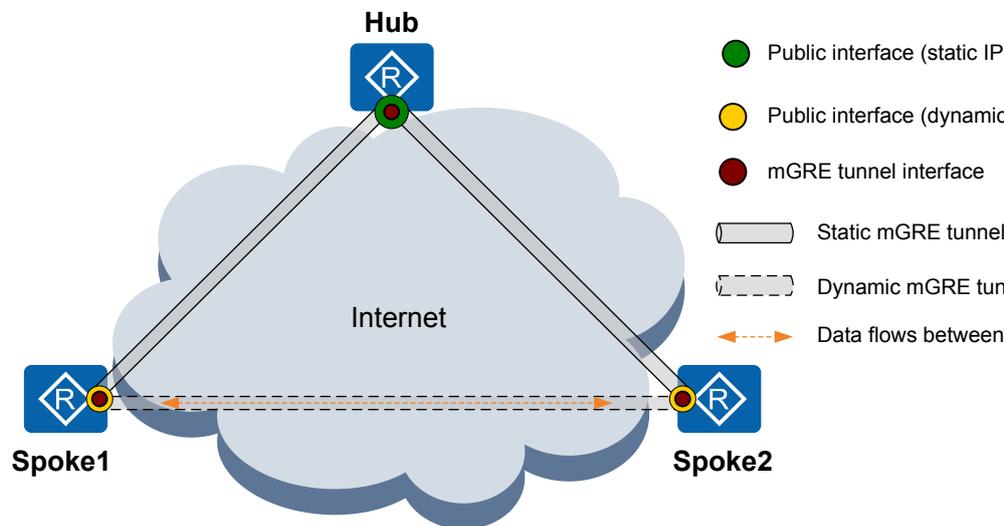
After the dynamic mGRE tunnel is set up, Spokes connected to the two Hubs can directly communicate with each other.

5.3 Application Scenarios for DSVPN

5.3.1 DSVPN Deployment on a Small- or Medium-sized Network

Small- and medium-sized networks have only a few branches, and the branches can dynamically establish VPNs by deploying Non-Shortcut Scenario of DSVPN.

Figure 5-11 DSVPN deployment on a small- or medium-sized network

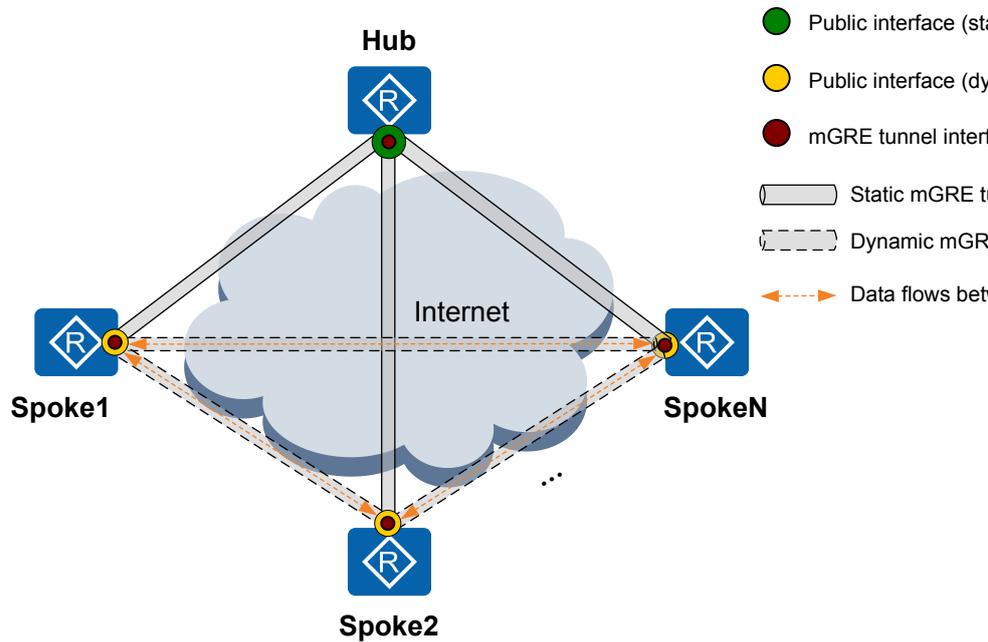


As shown in [Figure 5-11](#), Spoke1 and Spoke2 connect to the Hub through the public network. DSVPN is deployed to enable Spoke1 and Spoke2 to learn routes from each other. Spoke1 and Spoke2 can communicate with each other directly because they are each other's next hop.

5.3.2 DSVPN Deployment on a Large-sized Network

A large-sized network has a large number of branch offices. The deployment of Non-Shortcut Scenario of DSVPN requires the Spokes to have a large routing table and high forwarding performance. Shortcut Scenario of DSVPN can be deployed without upgrading the Spokes. This deployment reduces the routing entries on the Spokes, lowering the requirements on the Spokes' routing table size and forwarding performance.

Figure 5-12 DSVPN deployment on a large-sized network



As shown in [Figure 5-12](#), all the Spokes only have routes to the Hub. When two Spokes need to communicate with each other, the first packet is sent to the Hub. After that, a tunnel is established between the Spokes, and the Spokes can directly exchange data traffic.

5.3.3 Deploying DSVPN in Hierarchical Hub Networking

[Figure 5-13](#) shows a network topology of an enterprise. The organizations of the enterprise are hierarchical. DSVPN is deployed on the network. Some intermediate nodes function as both the Spokes and Hubs. For example, Hub1 and Hub2 function as Hubs for its downstream Spokes and serve as Spokes for the Hub in the headquarters.

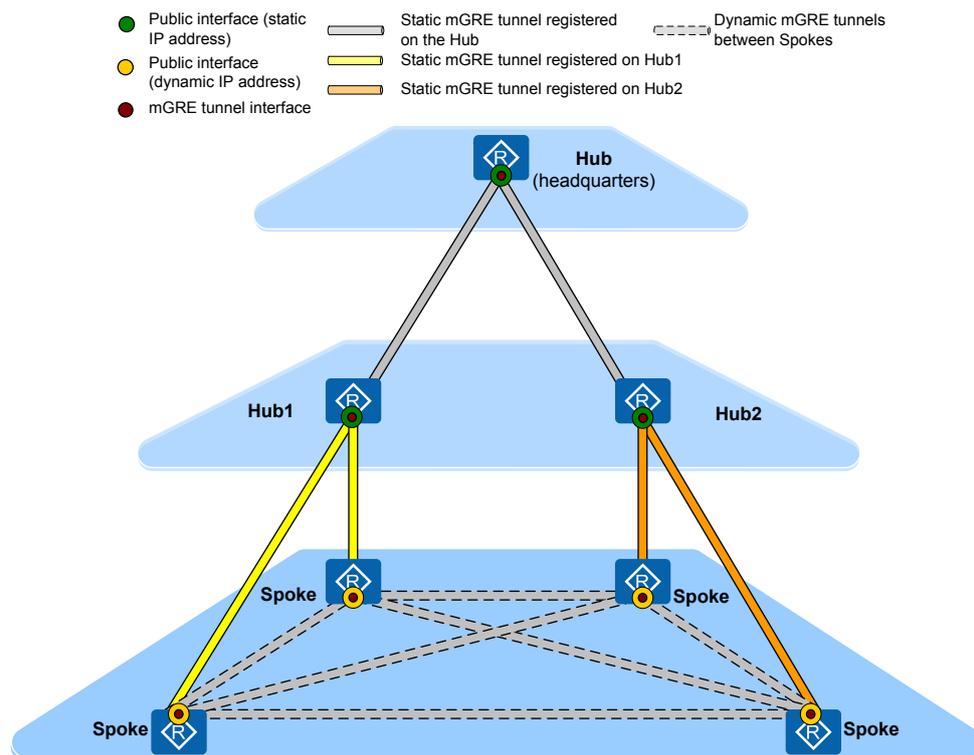
The principles of establishing dynamic mGRE tunnels between Spokes of Hub1 and Hub2 are similar to [Principles](#). When a Spoke of Hub1 needs to establish a dynamic mGRE tunnel with a Spoke of Hub2, the source Spoke sends an NHRP Resolution Request packet to Hub1. Hub1 sends the NHRP Resolution Request packet to the Hub, the Hub sends the NHRP Resolution Request packet to Hub2, and Hub2 sends the NHRP Resolution Request packet to the destination Spoke. Finally, a dynamic mGRE tunnel is set up between Spokes in hierarchical Hub networking.

NOTE

When DSVPN is deployed in hierarchical Hub networking, branches can learn routes from each other in shortcut mode only.

During the DSVPN deployment, two tunnel interfaces need to be created on Hub1 and Hub2 separately, so that Hub1 and Hub2 can set up static mGRE tunnels with the Hub and corresponding Spoke. Hub1 and Hub2 can be regarded as Spokes of the Hub.

Figure 5-13 Deploying DSVPN in hierarchical Hub networking



5.4 Licensing Requirements and Limitations for DSVPN

Involved Network Elements

None

License Requirements

For DSVPN-capable devices, their licensing requirements for the DSVPN function are as follows:

NOTE

DSVPN is a Huawei proprietary protocol and can only be used to interconnect AR routers.

- AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-SS&AR1200-S series: DSVPN is a basic feature of the device and is not under license control.
- AR2200&AR3200 series: By default, DSVPN function is disabled on a new device. To use the DSVPN function, apply for and purchase the following license from the Huawei local office.
- AR2200-S series: AR2200 Value-Added Security Package
- AR3200-S series: AR3200 Value-Added Security Package

Feature Limitations

When IPsec tunnels are deployed on the DSVPN network, rapidly updating the NHRP mapping table will cause IKE re-negotiation and may even interrupt services. Do not update the NHRP mapping table frequently.

5.5 Default Settings for DSVPN

Parameter	Default Setting
NHRP domain of an interface	0
NHRP authentication	Unspecified
Time interval at which a Spoke registers with the Hub	1800 seconds
Holding time of NHRP mapping entries	7200 seconds
NHRP redirect function	Disabled
NHRP shortcut function	Disabled
Adding dynamically registered branches to the NHRP multicast member table	Unspecified
Method to process conflicting NHRP mapping entries during NHRP registration	Not overridden
Referencing an IKE peer in the IPsec profile	Not referenced
Referencing an IPsec proposal in the IPsec profile	Not referenced
Using PFS in IPsec negotiation	Unused

5.6 Configuring DSVPN

Pre-configuration Tasks

Before configuring DSVPN, configure public network addresses to ensure that routes between nodes are reachable.

After DSVPN is configured, a Spoke can dynamically obtain the public network address of its peer device and establish a tunnel with the peer device to exchange data.

Configuration Procedure

Perform the following operations on the Hub and Spokes to configure DSVPN. Configuring an IPSec profile is optional. You are advised to perform this operation to protect packets against attacks because NHRP does not provide the encryption and decryption functions.

5.6.1 Configuring mGRE

Context

To implement DSVPN, create a tunnel interface and set the interface type to Multipoint GRE (mGRE). You only need to configure the source address or source interface but not the destination address on the mGRE interface. An mGRE tunnel interface has multiple remote ends and allows multiple GRE tunnels to be established on the interface. This simplifies GRE configuration on devices.

Perform the following operations on the Hub and Spokes.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface tunnel** *interface-number*

A tunnel interface is created and the tunnel interface view is displayed.

Step 3 Run **ip address** *ip-address* { *mask* | *mask-length* }

The IP address of the tunnel interface is configured.

Step 4 Run **tunnel-protocol gre p2mp**

The tunnel encapsulation mode is set to mGRE.



NOTICE

Changing the encapsulation mode of a tunnel interface deletes other parameters of the tunnel interface, including the source address or source interface configured for the tunnel interface, and NHRP parameters.

Step 5 Run **source** { [**vpn-instance** *vpn-instance-name*] *source-ip-address* | *interface-type interface-number* }

The source address or source interface is configured for the tunnel interface.



NOTICE

Changing the **source** command configuration will cause the IPsec configuration on the tunnel interface to be deleted.

Step 6 (Optional) Run **gre key** { **plain** *key-number* | [**cipher**] *plain-cipher-text* }

The key number of a tunnel interface is set.

By default, no key number is set for a tunnel interface.

When multiple mGRE tunnel interfaces are configured with the same source address or source interface, run this command to set a key number for each interface.



NOTICE

If **plain** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

----End

5.6.2 Configuring Routes

Context

The routes forwarded by a tunnel must be available on Spokes and the Hub so that packets encapsulated with mGRE can be forwarded correctly. These routes can be static routes or dynamic routes.

DSVPN supports the following route learning modes:

- Route learning between Spokes (non-shortcut)

The next-hop address of the route from the source Spoke to the destination Spoke is the tunnel address of the destination Spoke, and each Spoke needs to learn the route to the remote end. This consumes many CPU and memory resources and requires large routing tables and high performance on Spokes. In practice, the Spokes have low performance and store a limited number of routes. The route learning solution applies to small- and medium-sized networks where there are fewer network nodes and a small number of routes.

mGRE tunnels between Spokes are set up in non-shortcut mode.

- Spoke routes summarized to the Hub (shortcut mode)

The next-hop address of the route from the source Spoke to the destination Spoke is the tunnel address of the Hub, and Spokes only need to store routes to the Hub. The number

of routes of Spokes is reduced, so the route learning solution applies to large-sized networks with many Spokes.

mGRE tunnels between Spokes are set up in shortcut mode.

Static routes and dynamic routing protocols can be deployed in the preceding two route learning modes. DSVPN supports RIP, OSPF and BGP.

Perform the following configurations on the Hub and Spokes.

Procedure

- **Configure a static route.**

- a. Run **system-view**

The system view is displayed.

- b. Run **ip route-static** *ip-address* { *mask* | *mask-length* } *nexthop-address* [**description** *text*]

A static route is configured.

 **NOTE**

- When Spokes learn routes from each other, the next-hop address of the static route from a Spoke to the Hub or another Spoke is the tunnel address of the remote device.
- When routes of Spokes are summarized to the Hub, the next-hop address of the static route from a Spoke to the Hub or another Spoke is the tunnel address of the Hub.

- **Configuring a dynamic route**

- a. Run **system-view**

The system view is displayed.

- b. Configure a dynamic route.

Dynamic routes can be implemented using RIP, OSPF, or BGP. For the configuration of a dynamic routing protocol, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IP Unicast Routing*.

When configuring dynamic routing protocols, pay attention to the following points:

Scenario and Routing Protocol	RIP	OSPF	BGP
Non-Shortcut Scenario of DSVPN	Disable the split horizon and automatic route aggregation functions on the mGRE interface of the Hub.	Configure the OSPF network type to multicast using the ospf network-type broadcast command on the Hub and Spokes.	Route aggregation cannot be configured on the Hub.

Scenario and Routing Protocol	RIP	OSPF	BGP
Shortcut Scenario of DSVPN	Enable the split horizon and automatic route aggregation functions on the mGRE interface of the Hub.	Configure the OSPF network type to Point-to-Multipoint (P2MP) using the ospf network-type p2mp command on the Hub and Spokes.	Configure route aggregation on the Hub.

----End

5.6.3 Configuring NHRP

Context

NHRP enables a source Spoke on a public network to dynamically obtain the public network address of a destination Spoke. When a Spoke connects to a public network, it sends NHRP Registration Request packets to the Hub by using the public network address of the outbound interface. The Hub creates or updates NHRP mapping entries based on the packets received. Two Spokes exchange NHRP Resolution Request and Reply packets to create or update NHRP mapping entries between them.



NOTICE

When configuring the NHRP authentication string, if **simple** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

Perform the following operations on the Hub and Spokes.

Procedure

- Configure the Hub.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **interface tunnel interface-number**

The tunnel interface view is displayed.
 - c. (Optional) Run **nhrp network-id number**

An NHRP domain is configured for the tunnel interface.

By default, a tunnel interface belongs to NHRP domain 0.

- d. **Run `nhrp entry multicast dynamic`**

The dynamically registered Spoke is added to the NHRP multicast member table.

By default, no dynamically registered Spoke is added to the NHRP multicast member table.
- e. **Run `nhrp authentication { simple string | cipher cipher-string }`**

The NHRP authentication string is configured.

By default, no NHRP authentication string is configured.
- f. (Optional) **Run `nhrp entry holdtime seconds seconds`**

The aging time of NHRP mapping entries is configured.

By default, the aging time of NHRP mapping entries is 7200 seconds.
- g. (Optional) **Run `nhrp redirect`**

The NHRP redirect function is enabled.

This configuration is required only when the shortcut mode is used. By default, the NHRP redirect function is disabled.
- **Configure the Spokes.**
 - a. **Run `system-view`**

The system view is displayed.
 - b. **Run `interface tunnel interface-number`**

The tunnel interface view is displayed.
 - c. (Optional) **Run `nhrp network-id number`**

An NHRP domain is configured for the tunnel interface.

By default, a tunnel interface belongs to NHRP domain 0.
 - d. **Run `nhrp entry protocol-address { dns-name | nbma-address } [register [preference preference-value]] [track apn apn-name]`**

An NHRP mapping entry is configured.

When the **track apn** parameter is specified, whether the NHRP mapping entry takes effect depends on the APN status. If the APN is valid, the NHRP mapping entry takes effect; otherwise, the configuration is saved but the NHRP mapping entry does not take effect.
 - e. (Optional) **Run `nhrp registration no-unique`**

The device is configured to send NHRP packets that carry the no-unique flag to instruct the remote end to overwrite conflicting NHRP peer entries.

By default, the device sends NHRP packets that do not carry the no-unique flag to instruct the remote end not to overwrite conflicting NHRP peer entries.
 - f. **Run `nhrp authentication { simple string | cipher cipher-string }`**

The NHRP authentication string is configured.

By default, no NHRP authentication string is configured.

 **NOTE**

If the NHRP authentication string is configured on the Hub, it must also be configured on the Spoke.

- g. (Optional) Run **nhrp registration interval seconds**

The NHRP registration interval is configured.

By default, a Spoke registers with the Hub at an interval of 1800 seconds.

- h. (Optional) Run **nhrp entry holdtime seconds seconds**

The aging time of NHRP mapping entries is configured.

By default, the aging time of NHRP mapping entries is 7200 seconds.

- i. (Optional) Run **nhrp shortcut**

The NHRP shortcut function is enabled.

This configuration is required only when the shortcut mode is used. By default, the NHRP shortcut function is disabled.

----End

5.6.4 (Optional) Configuring an IPSec Profile

Context

Data transmitted between the central office and a branch, and between branches can be encrypted to increase data security. Binding an IPSec profile to DSVPN can dynamically establish an mGRE over IPSec tunnel.

Before configuring an IPSec profile for DSVPN, you need to perform the following operations:

- Create an IKE peer. For details, see [6.10.2 Configuring an IKE Peer](#).
- Create an IPSec proposal. For details, see [6.8.1 Configuring an IPSec Proposal](#).

After completing the preceding configuration, perform the following operations on the Hub and Spokes.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec profile profile-name**

An IPSec profile is created and the IPSec profile view is displayed.

Step 3 Run **ike-peer peer-name**

An IKE peer is bound to the IPSec profile.

Step 4 Run **proposal proposal-name**

An IPSec proposal is bound to the IPSec profile.

Step 5 (Optional) Run **pfs { dh-group1 | dh-group2 | dh-group5 | dh-group14 | dh-group19 | dh-group20 | dh-group21 }**

The perfect forward secrecy (PFS) feature is used in IPsec negotiation.

By default, PFS is not used in IPsec negotiation.

 **NOTICE**

If PFS is specified on the local end, you also need to specify PFS on the remote peer. The Diffie-Hellman groups specified on the two ends must be the same. Otherwise, the negotiation fails.

Step 6 Run **quit**

Return to the system view.

Step 7 Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Step 8 Run **tunnel-protocol gre p2mp**

The tunnel encapsulation mode is configured.

Step 9 Run **ipsec profile** *profile-name*

The tunnel interface is bound to an IPsec profile.

----End

5.6.5 Verifying the DSVPN Configuration

Prerequisites

All DSVPN configurations have been completed.

Procedure

- Run the **display nhrp peer** command to check NHRP mapping entries.
- Run the **display nhrp peer maximum-history** command to check the history statistics on NHRP peer entries.
- Run the **display ipsec profile [brief | name *profile-name*]** command to check the IPsec profile configuration.
- Run the **display ipsec sa profile *profile-name*** command to check the information of IPsec SA.

----End

5.7 Maintaining DSVPN

5.7.1 Clearing DSVPN Running Statistics

Context



NOTICE

Statistics cannot be restored after being cleared. Exercise caution when you run the command.

Procedure

- Run the **reset nhrp statistics interface tunnel** *interface-number* command in the user view to clear NHRP packet statistics on a specified tunnel interface.
- Run the **reset nhrp peer maximum-history** command in the user view to clear historical statistics on NHRP peer entries.

----End

5.7.2 Monitoring DSVPN Running Statistics

Prerequisites

All DSVPN configurations have been completed.

Procedure

- Run the **display nhrp statistics interface tunnel** *interface-number* command to check NHRP packet statistics.
- Run the **display nhrp peer maximum-history** command to check historical statistics on NHRP peer entries.

----End

5.8 Configuration Examples for DSVPN

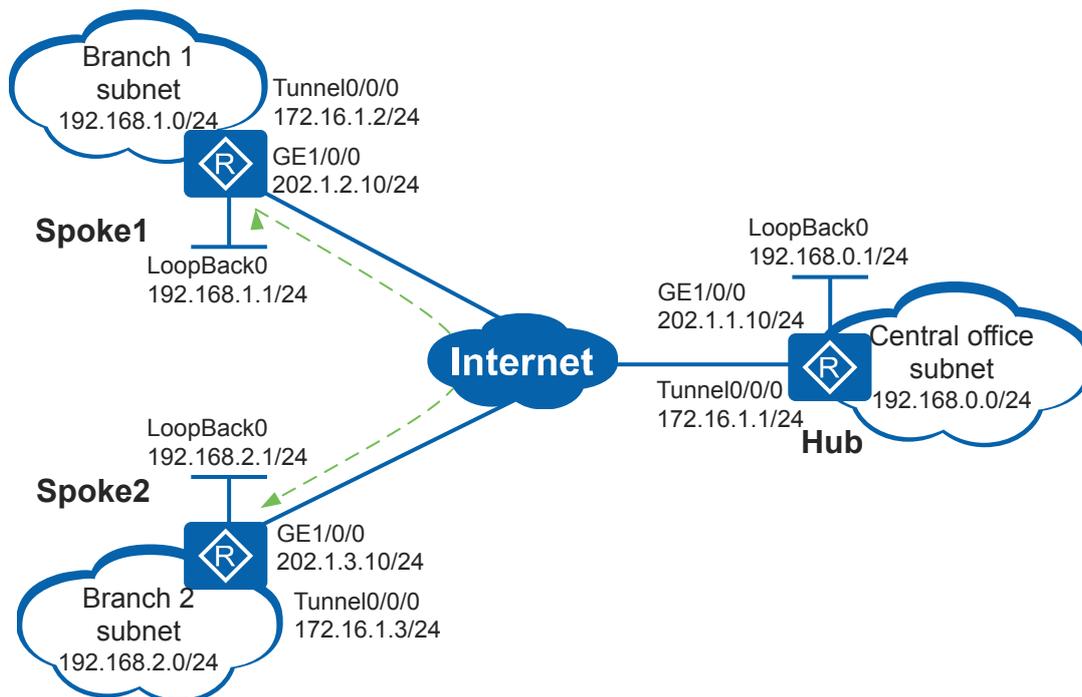
5.8.1 Example for Configuring Non-Shortcut Scenario of DSVPN (Static Route)

Networking Requirements

A small enterprise has a central office (Hub) and two branches (Spoke1 and Spoke2) which are located in different areas. The network between the Hub and Spokes is stable. The Spokes use dynamic addresses to connect to the public network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-14 Networking diagram for the Non-Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Non-Shortcut Scenario of DSVPN is implemented because the enterprise has a small number of branches.
3. Static routes can be configured to realize communication between the Hub and Spokes because the network is stable.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface gigabitethernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of Spoke1 and Spoke2 as shown in [Figure 5-14](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure static routes.

Configure Hub.

```
[Hub] ip route-static 192.168.1.0 255.255.255.0 172.16.1.2
[Hub] ip route-static 192.168.2.0 255.255.255.0 172.16.1.3
```

Configure Spoke1.

```
[Spoke1] ip route-static 192.168.0.0 255.255.255.0 172.16.1.1
[Spoke1] ip route-static 192.168.2.0 255.255.255.0 172.16.1.3
```

Configure Spoke2.

```
[Spoke2] ip route-static 192.168.0.0 255.255.255.0 172.16.1.1
[Spoke2] ip route-static 192.168.1.0 255.255.255.0 172.16.1.2
```

Step 4 Configure tunnel interfaces.

Configure tunnel interfaces on Hub and Spokes and configure static NHRP peer entries of Spoke1 and Spoke2.

Configure a tunnel interface on Hub.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source gigabitethernet 1/0/0
[Hub-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP peer entry of Hub on Spoke1.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP mapping entry of Hub on Spoke2.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
```

```
[Spoke2-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

NOTE

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 6 Check the static routes.

Check the static routes on Hub.

Run the **display ip routing-table protocol static** command on Hub. The command output is as follows:

```
[Hub] display ip routing-table protocol static
Route Flags: R - relay, D - download to fib
-----
Public routing table : Static
      Destinations : 2          Routes : 2          Configured Routes : 2

Static routing table status : <Active>
      Destinations : 2          Routes : 2

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
192.168.1.0/24     Static 60   0        RD   172.16.1.2            Tunnel0/0/0
192.168.2.0/24     Static 60   0        RD   172.16.1.3            Tunnel0/0/0

Static routing table status : <Inactive>
      Destinations : 0          Routes : 0
```

Check the static routes on the Spokes.

Run the **display ip routing-table protocol static** command on Spoke1. The command output is as follows:

```
[Spoke1] display ip routing-table protocol static
Route Flags: R - relay, D - download to fib
-----
Public routing table : Static
      Destinations : 2          Routes : 2          Configured Routes : 2

Static routing table status : <Active>
      Destinations : 2          Routes : 2

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
192.168.0.0/24     Static 60   0        RD   172.16.1.1            Tunnel0/0/0
192.168.2.0/24     Static 60   0        RD   172.16.1.3            Tunnel0/0/0

Static routing table status : <Inactive>
      Destinations : 0          Routes : 0
```

Run the **display ip routing-table protocol static** command on Spoke2. The command output is as follows:

```
[Spoke2] display ip routing-table protocol static
Route Flags: R - relay, D - download to fib
-----
Public routing table : Static
      Destinations : 2          Routes : 2          Configured Routes : 2

Static routing table status : <Active>
      Destinations : 2          Routes : 2

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
192.168.0.0/24     Static 60   0        RD   172.16.1.1            Tunnel0/0/0
192.168.1.0/24     Static 60   0        RD   172.16.1.2            Tunnel0/0/0

Static routing table status : <Inactive>
      Destinations : 0          Routes : 0
```

Step 7 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:46:35
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:28
Expire time     : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    local     up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:28
Expire time     : 01:59:32
-----
Number of nhrp peers: 3
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:43:32
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
```

```
172.16.1.3      32      202.1.3.10    172.16.1.3    local      up
```

```
-----  
Tunnel interface: Tunnel0/0/0  
Created time      : 00:00:47  
Expire time       : 01:59:13  
  
Number of nhrp peers: 3
```

----End

Configuration Files

- Hub configuration file

```
#  
sysname Hub  
#  
interface GigabitEthernet1/0/0  
 ip address 202.1.1.10 255.255.255.0  
#  
interface LoopBack0  
 ip address 192.168.0.1 255.255.255.0  
#  
interface Tunnel0/0/0  
 ip address 172.16.1.1 255.255.255.0  
 tunnel-protocol gre p2mp  
 source GigabitEthernet1/0/0  
#  
ospf 2 router-id 202.1.1.10  
 area 0.0.0.1  
  network 202.1.1.0 0.0.0.255  
#  
ip route-static 192.168.1.0 255.255.255.0 172.16.1.2  
ip route-static 192.168.2.0 255.255.255.0 172.16.1.3  
#  
return
```

- Spoke1 configuration file

```
#  
sysname Spoke1  
#  
interface GigabitEthernet1/0/0  
 ip address 202.1.2.10 255.255.255.0  
#  
interface LoopBack0  
 ip address 192.168.1.1 255.255.255.0  
#  
interface Tunnel0/0/0  
 ip address 172.16.1.2 255.255.255.0  
 tunnel-protocol gre p2mp  
 source GigabitEthernet1/0/0  
 nhrp entry 172.16.1.1 202.1.1.10 register  
#  
ospf 2 router-id 202.1.2.10  
 area 0.0.0.1  
  network 202.1.2.0 0.0.0.255  
#  
ip route-static 192.168.0.0 255.255.255.0 172.16.1.1  
ip route-static 192.168.2.0 255.255.255.0 172.16.1.3  
#  
return
```

- Spoke2 configuration file

```
#  
sysname Spoke2  
#  
interface GigabitEthernet1/0/0  
 ip address 202.1.3.10 255.255.255.0
```

```
#
interface LoopBack0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
ip route-static 192.168.0.0 255.255.255.0 172.16.1.1
ip route-static 192.168.1.0 255.255.255.0 172.16.1.2
#
return
```

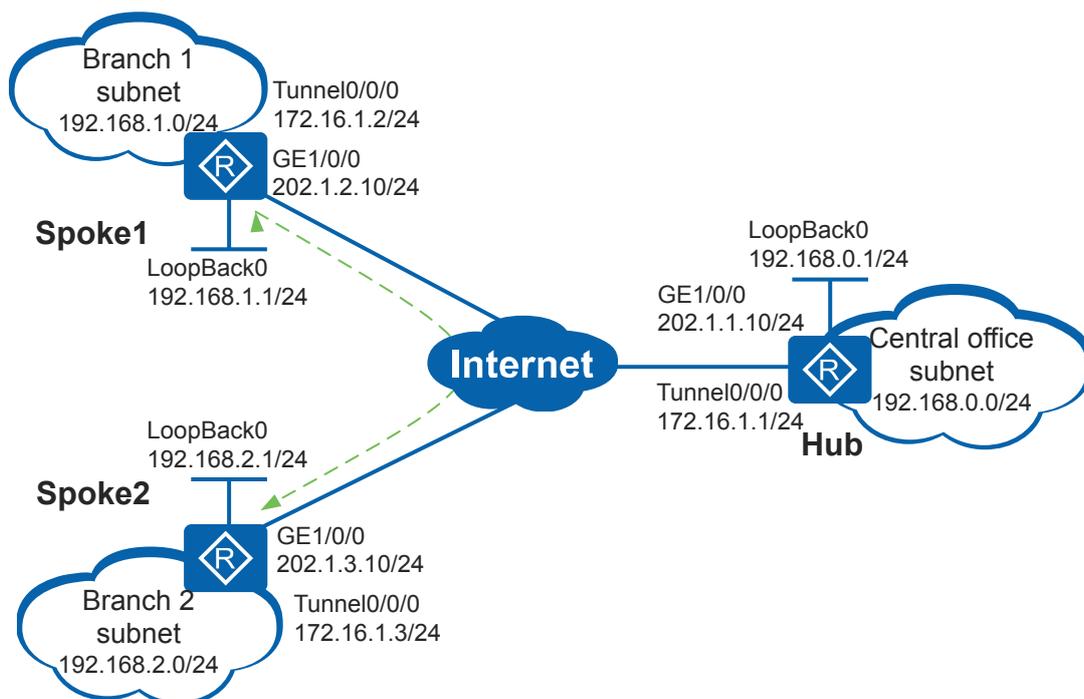
5.8.2 Example for Configuring Non-Shortcut Scenario of DSVPN (RIP)

Networking Requirements

A small enterprise has a central office (Hub) and two branches (Spoke1 and Spoke2) which are located in different areas. The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Routing Information Protocol (RIP) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-15 Networking diagram for the Non-Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Non-Shortcut Scenario of DSVPN is implemented because the enterprise has a small number of branches.
3. The networks of the central office and branches frequently change. RIP is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface gigabitethernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-15](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure the basic RIP functions.

Configure Hub.

```
[Hub] rip 1
[Hub-rip-1] version 2
[Hub-rip-1] undo summary
[Hub-rip-1] network 172.16.0.0
[Hub-rip-1] network 192.168.0.0
[Hub-rip-1] quit
```

Configure Spoke1.

```
[Spoke1] rip 1
[Spoke1-rip-1] version 2
[Spoke1-rip-1] network 172.16.0.0
[Spoke1-rip-1] network 192.168.1.0
[Spoke1-rip-1] quit
```

Configure Spoke2.

```
[Spoke2] rip 1
[Spoke2-rip-1] version 2
[Spoke2-rip-1] network 172.16.0.0
[Spoke2-rip-1] network 192.168.2.0
[Spoke2-rip-1] quit
```

NOTE

The RIP configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 4 Configure tunnel interfaces.

Configure route attributes on Hub to allow Spokes to learn routes from each other. Configure static NHRP mapping entries of Hub on Spoke1 and Spoke2.

Configure a tunnel interface and RIP on Hub.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source gigabitethernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] undo rip split-horizon
[Hub-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP mapping entry of Hub on Spoke1.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP mapping entry of Hub on Spoke2.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
```

```

Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1    32    202.1.1.10    172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:10:58
Expire time    : --
Number of nhrp peers: 1
  
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```

[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1    32    202.1.1.10    172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:07:55
Expire time    : --
Number of nhrp peers: 1
  
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```

[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2    32    202.1.2.10    172.16.1.2    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:02:02
Expire time    : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3    32    202.1.3.10    172.16.1.3    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:01:53
Expire time    : 01:59:35
Number of nhrp peers: 2
  
```

Step 6 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```

[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=1 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=1 ms
  
```

```
--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/1/2 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:46:35
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:28
Expire time     : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:28
Expire time     : 01:59:32
-----
Number of nhrp peers: 3
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:43:32
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:47
Expire time     : 01:59:13
-----
Number of nhrp peers: 3
```

----End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 undo rip split-horizon
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp entry multicast dynamic
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
#
rip 1
 undo summary
 version 2
 network 172.16.0.0
 network 192.168.0.0
#
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 2 router-id 202.1.2.10
 area 0.0.0.1
  network 202.1.2.0 0.0.0.255
#
rip 1
 version 2
 network 172.16.0.0
 network 192.168.1.0
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
 ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.2.1 255.255.255.0
```

```
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
rip 1
version 2
network 172.16.0.0
network 192.168.2.0
#
return
```

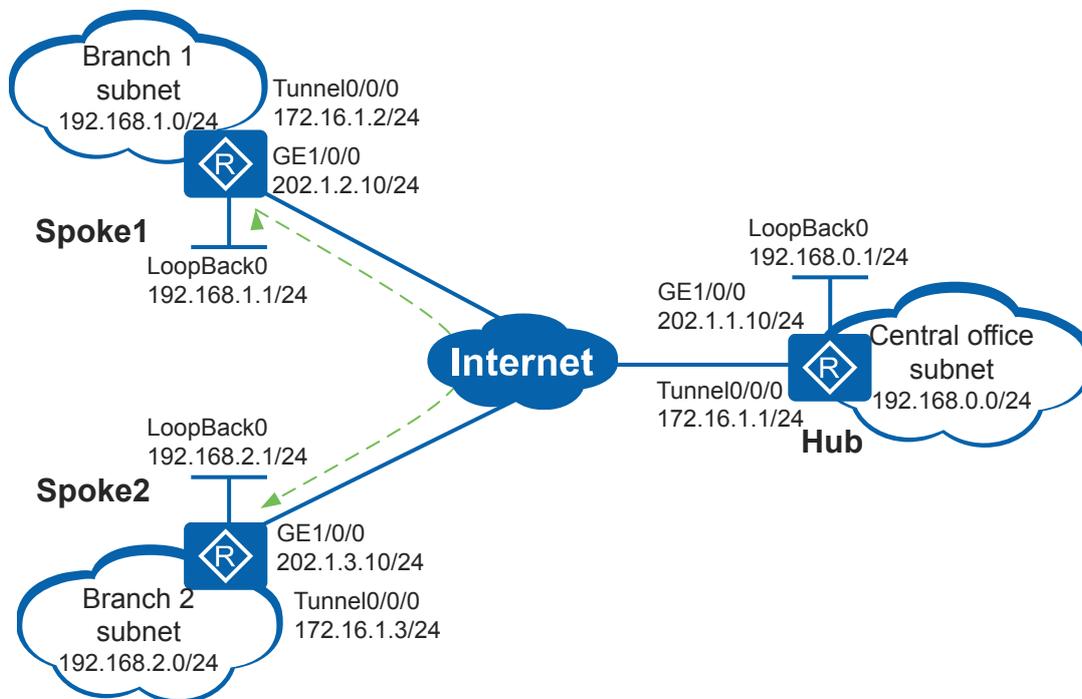
5.8.3 Example for Configuring Non-Shortcut Scenario of DSVPN (OSPF)

Networking Requirements

A small enterprise has a central office (Hub) and two branches (Spoke1 and Spoke2) which are located in different areas. The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Open Shortest Path First (OSPF) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-16 Networking diagram for the Non-Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Non-Shortcut Scenario of DSVPN is implemented because the enterprise has a small number of branches.
3. The networks of the central office and branches frequently change. OSPF is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-16](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure the basic OSPF functions.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

NOTE

The OSPF configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 4 Configure tunnel interfaces.

Set the OSPF network type to broadcast on Hub and Spokes to allow Spokes to learn routes from each other. Configure static NHRP mapping entries of Hub on Spoke1 and Spoke2.

Configure a tunnel interface and OSPF on Hub.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] ospf network-type broadcast
[Hub-Tunnel0/0/0] quit
```

On Spoke1, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] ospf network-type broadcast
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] ospf network-type broadcast
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 6 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
```

```

Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/2/2 ms
  
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```

[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:46:35
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    local     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32

Number of nhrp peers: 3
  
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```

[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:43:32
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:47
Expire time    : 01:59:13
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    local     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:47
Expire time    : 01:59:13
  
```

```
Number of nhrp peers: 3
```

---End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type broadcast
 nhrp entry multicast dynamic
#
ospf 1 router-id 172.16.1.1
 area 0.0.0.0
  network 172.16.1.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
#
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type broadcast
 nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.2
 area 0.0.0.0
  network 172.16.1.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
#
ospf 2 router-id 202.1.2.10
 area 0.0.0.1
  network 202.1.2.0 0.0.0.255
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
```

```
#
interface GigabitEthernet1/0/0
ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type broadcast
nhp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.3
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
return
```

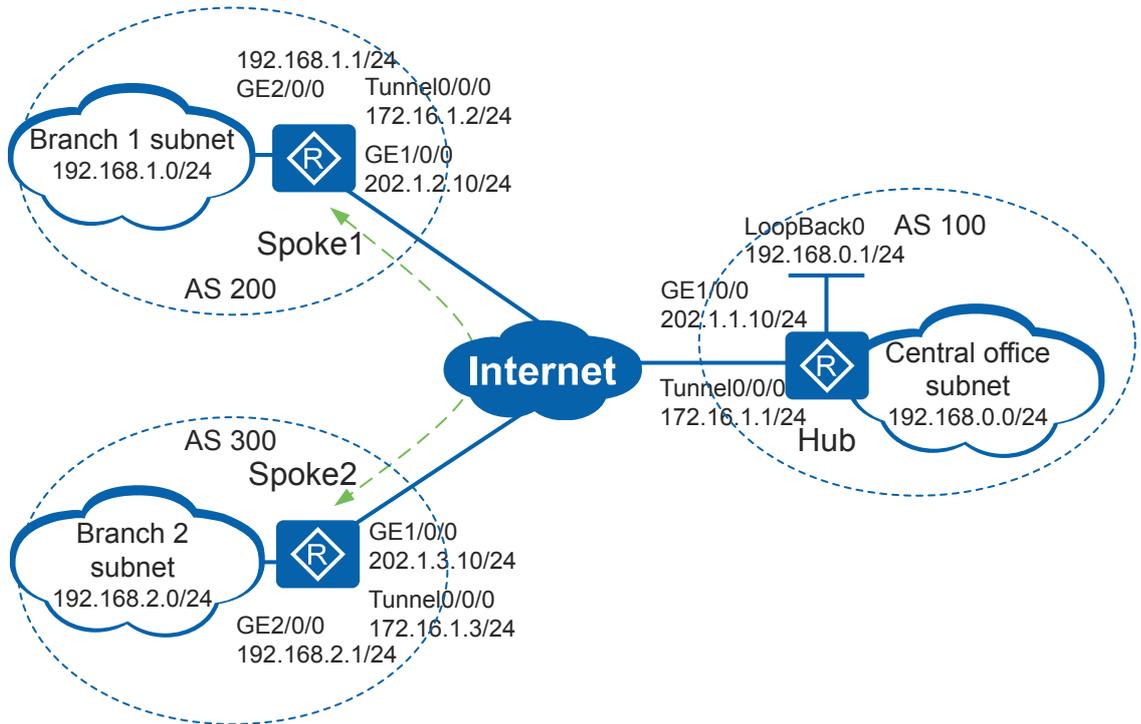
5.8.4 Example for Configuring Non-Shortcut Scenario of DSVPN (BGP)

Networking Requirements

A small enterprise has a central office (Hub) and two branches (Spoke1 and Spoke2) which are located in different areas and belong to different ASs. The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. On the enterprise network, Open Shortest Path First (OSPF) is used for intra-AS routing and External Border Gateway Protocol (EBGP) is used for inter-AS routing.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-17 Networking diagram for the Non-Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Non-Shortcut Scenario of DSVPN is implemented because the enterprise has a small number of branches.
3. The networks of the central office and branches frequently change. BGP is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-17](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure reachable routes between the ASs.

Configure OSPF to implement reachable routes between Hub and Spokes that are located in different ASs.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 4 Configure Basic EBGp Functions

Configure Hub.

```
[Hub] bgp 100
[Hub-bgp] router-id 172.16.1.1
[Hub-bgp] import-route ospf 1
[Hub-bgp] peer 172.16.1.2 as-number 200
```

```
[Hub-bgp] peer 172.16.1.3 as-number 300
[Hub-bgp] quit
```

Configure Spoke1.

```
[Spoke1] bgp 200
[Spoke1-bgp] router-id 172.16.1.2
[Spoke1-bgp] import-route ospf 1
[Spoke1-bgp] peer 172.16.1.1 as-number 100
[Spoke1-bgp] peer 172.16.1.3 as-number 300
[Spoke1-bgp] quit
```

Configure Spoke2.

```
[Spoke2] bgp 300
[Spoke2-bgp] router-id 172.16.1.3
[Spoke2-bgp] import-route ospf 1
[Spoke2-bgp] peer 172.16.1.1 as-number 100
[Spoke2-bgp] peer 172.16.1.2 as-number 200
[Spoke2-bgp] quit
```

NOTE

The basic BGP configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 5 Configure tunnel interfaces.

Configure route attributes on Hub and Spokes to allow Spokes to learn routes from each other. Configure static NHRP mapping entries of Hub on Spoke1 and Spoke2.

NOTE

In the non-shortcut scenario, configure BGP and set relevant attributes in the BGP view.

Configure a tunnel interface on Hub.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP mapping entry of Hub on Spoke1.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] quit
```

Configure a tunnel interface and a static NHRP mapping entry of Hub on Spoke2.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] quit
```

Step 6 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr  NextHop-addr  Type  Flag
```

```
-----
172.16.1.1      32    202.1.1.10    172.16.1.1     hub      up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub      up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

When you run the **display nhrp peer all** command, you can view the static NHRP mapping entries of Hub and dynamic NHRP mapping entries of each other on Spoke1 and Spoke2. Exchange of BGP packets triggers the Spokes to establish a dynamic tunnel.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 7 Run the **ping** command to check the configuration result.

On Spoke1, ping the subnet address 192.168.2.1 of Spoke2.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=255 time=5 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=3 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=3 ms

--- 192.168.2.1 ping statistics ---
```

```
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 2/3/5 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:46:35
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    local     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32

Number of nhrp peers: 3
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:43:32
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    remote    up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:47
Expire time    : 01:59:13
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    local     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:47
Expire time    : 01:59:13

Number of nhrp peers: 3
```

---End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp entry multicast dynamic
#
bgp 100
 router-id 172.16.1.1
 peer 172.16.1.2 as-number 200
 peer 172.16.1.3 as-number 300
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 172.16.1.2 enable
  peer 172.16.1.3 enable
#
ospf 1 router-id 172.16.1.1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
#
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp entry 172.16.1.1 202.1.1.10 register
#
bgp 200
 router-id 172.16.1.2
 peer 172.16.1.1 as-number 100
 peer 172.16.1.3 as-number 300
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 172.16.1.1 enable
  peer 172.16.1.3 enable
#
ospf 1 router-id 172.16.1.2
 area 0.0.0.0
```

```
network 192.168.1.0 0.0.0.255
#
ospf 2 router-id 202.1.2.10
area 0.0.0.1
network 202.1.2.0 0.0.0.255
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
ip address 202.1.3.10 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
nhrp entry 172.16.1.1 202.1.1.10 register
#
bgp 300
router-id 172.16.1.3
peer 172.16.1.1 as-number 100
peer 172.16.1.2 as-number 200
#
ipv4-family unicast
undo synchronization
import-route ospf 1
peer 172.16.1.1 enable
peer 172.16.1.2 enable
#
ospf 1 router-id 172.16.1.3
area 0.0.0.0
network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
return
```

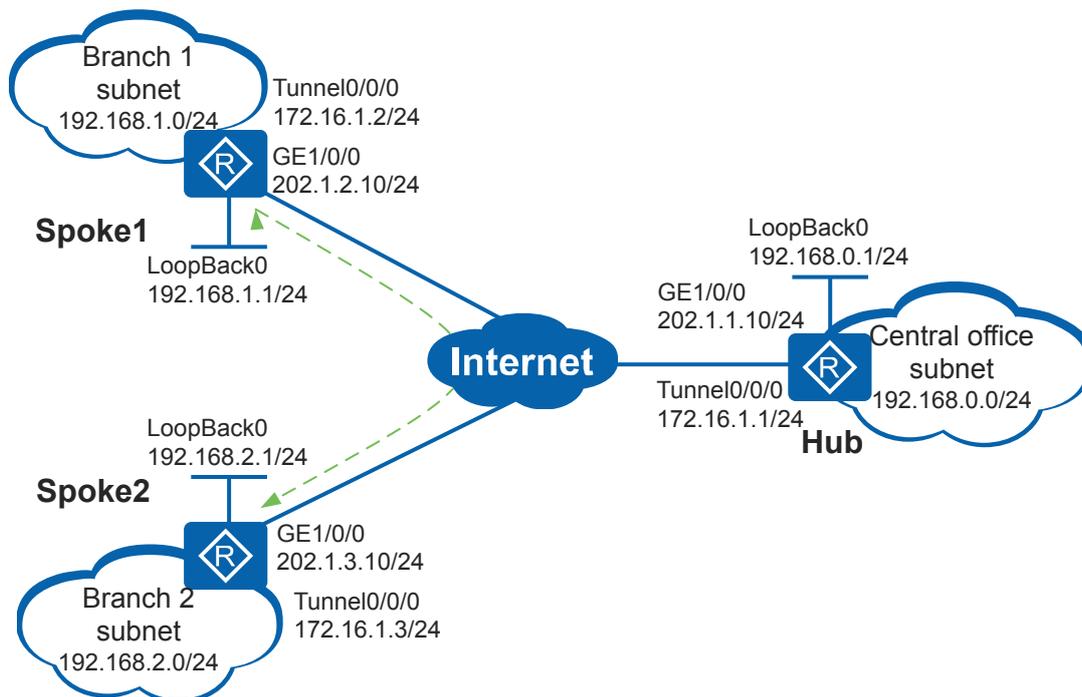
5.8.5 Example for Configuring Shortcut Scenario of DSVPN (RIP)

Networking Requirements

A large-scale enterprise has a central office (Hub) and multiple branches which are located in different areas (this example shows only two Spokes Spoke1 and Spoke2). The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Routing Information Protocol (RIP) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-18 Networking diagram for the Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. RIP is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-18](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure the basic RIP functions.

Configure Hub.

```
[Hub] rip 1
[Hub-rip-1] version 2
[Hub-rip-1] network 172.16.0.0
[Hub-rip-1] network 192.168.0.0
[Hub-rip-1] quit
```

Configure Spoke1.

```
[Spoke1] rip 1
[Spoke1-rip-1] version 2
[Spoke1-rip-1] network 172.16.0.0
[Spoke1-rip-1] network 192.168.1.0
[Spoke1-rip-1] quit
```

Configure Spoke2.

```
[Spoke2] rip 1
[Spoke2-rip-1] version 2
[Spoke2-rip-1] network 172.16.0.0
[Spoke2-rip-1] network 192.168.2.0
[Spoke2-rip-1] quit
```

NOTE

The RIP configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 4 Configure tunnel interfaces.

Configure RIP-2 route summarization on Hub and RIP-2 on the Spokes, so that the Spokes have reachable routes to Hub. Enable the NHRP redirect function on Hub. Configure NHRP mapping entries of Hub and enable the NHRP shortcut function on Spoke1 and Spoke2.

On Hub, configure a tunnel interface, configure RIP, and enable the NHRP redirect function.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] rip version 2 multicast
[Hub-Tunnel0/0/0] rip summary-address 192.168.0.0 255.255.0.0
[Hub-Tunnel0/0/0] nhrp redirect
[Hub-Tunnel0/0/0] quit
```

 **NOTE**

When configuring route summarization, the specified summarized address must exist on the local device. Therefore, a LoopBack address must be configured.

On Spoke1, configure a tunnel interface, RIP, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] rip version 2 multicast
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface, RIP, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] rip version 2 multicast
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
-----
Number of nhrp peers: 2
```

Step 6 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/2/3 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:46:35
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
192.168.2.1    32    202.1.3.10     172.16.1.3   remote-network up
-----
Tunnel interface: Tunnel0/0/0
```

```

Created time      : 00:00:28
Expire time      : 01:59:32
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
172.16.1.3       32   202.1.3.10     172.16.1.3     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:28
Expire time     : 01:59:32
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
172.16.1.2       32   202.1.2.10     172.16.1.2     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:28
Expire time     : 01:59:32

Number of nhrp peers: 4
  
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```

[Spoke2] display nhrp peer all
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
172.16.1.1       32   202.1.1.10     172.16.1.1     hub             up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:43:32
Expire time     : --
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
192.168.1.1      32   202.1.2.10     172.16.1.2     remote-network  up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
172.16.1.2       32   202.1.2.10     172.16.1.2     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr    Mask  NBMA-addr      NextHop-addr    Type            Flag
-----
172.16.1.3       32   202.1.3.10     172.16.1.3     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13

Number of nhrp peers: 4
  
```

----End

Configuration Files

- Hub configuration file

```

#
sysname Hub
#
interface GigabitEthernet1/0/0
  
```

```
ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.1 255.255.255.0
rip version 2 multicast
rip summary-address 192.168.0.0 255.255.0.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
nhrp redirect
nhrp entry multicast dynamic
#
ospf 2 router-id 202.1.1.10
area 0.0.0.1
network 202.1.1.0 0.0.0.255
#
rip 1
version 2
network 172.16.0.0
network 192.168.0.0
#
return
```

● Spoke1 configuration file

```
#
sysname Spoke1
#
interface GigabitEthernet1/0/0
ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.2 255.255.255.0
rip version 2 multicast
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
nhrp shortcut
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 2 router-id 202.1.2.10
area 0.0.0.1
network 202.1.2.0 0.0.0.255
#
rip 1
version 2
network 172.16.0.0
network 192.168.1.0
#
return
```

● Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
rip version 2 multicast
tunnel-protocol gre p2mp
```

```

source GigabitEthernet1/0/0
nhrp shortcut
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
rip 1
version 2
network 172.16.0.0
network 192.168.2.0
#
return
    
```

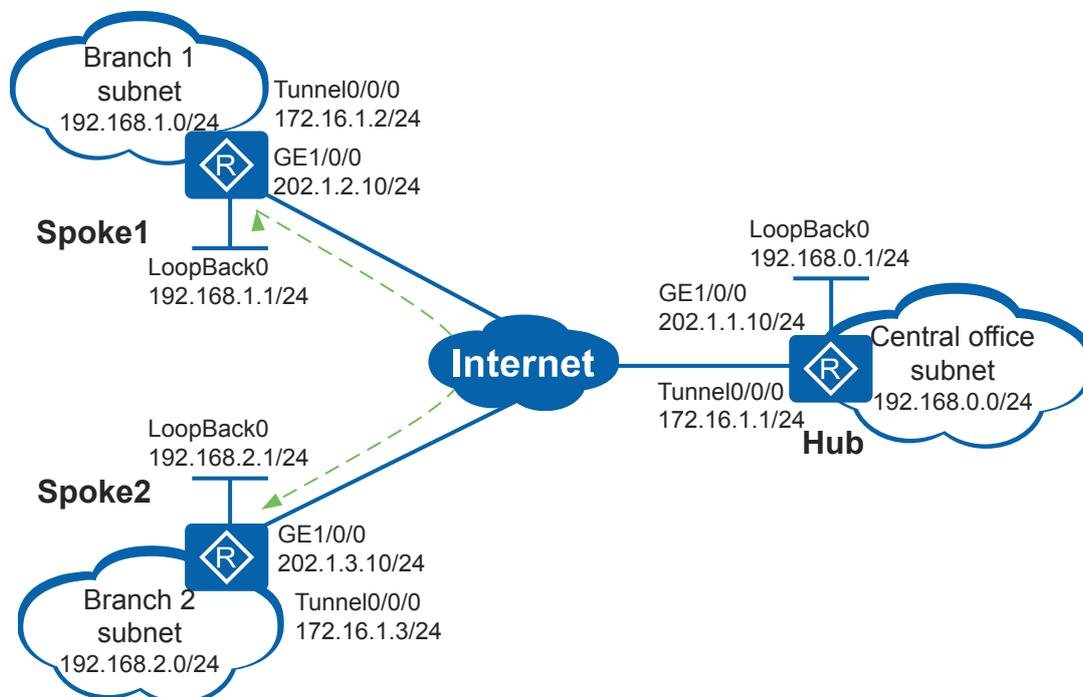
5.8.6 Example for Configuring Shortcut Scenario of DSVPN (OSPF)

Networking Requirements

A large-scale enterprise has a central office (Hub) and multiple branches which are located in different areas (this example shows only two Spokes Spoke1 and Spoke2). The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Open Shortest Path First (OSPF) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-19 Networking diagram for the Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. OSPF is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-19](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure the basic OSPF functions.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
```

```
[Hub-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

NOTE

The OSPF configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 4 Configure tunnel interfaces.

Configure the OSPF network type to Point-to-Multipoint (P2MP) on Hub and Spokes. Enable the NHRP redirect function on Hub. Configure NHRP mapping entries of Hub and enable the NHRP shortcut function on Spoke1 and Spoke2.

On Hub, configure a tunnel interface, configure OSPF, and enable the NHRP redirect function.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] ospf network-type p2mp
[Hub-Tunnel0/0/0] nhrp redirect
[Hub-Tunnel0/0/0] quit
```

On Spoke1, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 6 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=4 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=9 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/3/9 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:46:35
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
192.168.2.1    32    202.1.3.10     172.16.1.3     remote-network up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32

Number of nhrp peers: 4
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:43:32
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr    Type           Flag
-----
192.168.1.1    32    202.1.2.10     172.16.1.2     remote-network up
-----
Tunnel interface: Tunnel0/0/0
```

```

Created time      : 00:00:47
Expire time      : 01:59:13
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.2      32   202.1.2.10     172.16.1.2    remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.3      32   202.1.3.10     172.16.1.3    local          up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13

Number of nhrp peers: 4
  
```

----End

Configuration Files

- Hub configuration file

```

#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type p2mp
 nhrp redirect
 nhrp entry multicast dynamic
#
ospf 1 router-id 172.16.1.1
 area 0.0.0.0
  network 172.16.1.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
#
return
  
```

- Spoke1 configuration file

```

#
sysname Spoke1
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
  
```

```
source GigabitEthernet1/0/0
ospf network-type p2mp
nhrp shortcut
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.2
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.1.0 0.0.0.255
#
ospf 2 router-id 202.1.2.10
area 0.0.0.1
network 202.1.2.0 0.0.0.255
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type p2mp
nhrp shortcut
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.3
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
return
```

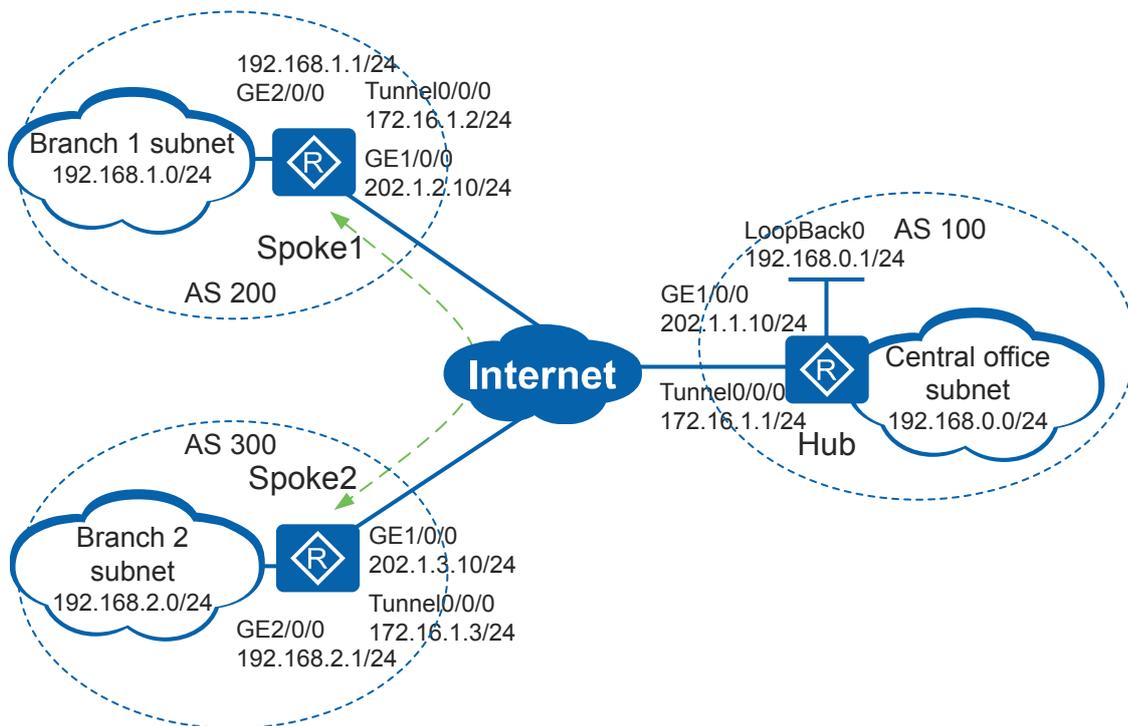
5.8.7 Example for Configuring Shortcut Scenario of DSVPN (BGP)

Networking Requirements

A large-scale enterprise has a central office (Hub) and multiple branches which are located in different areas and belong to different ASs (this example shows only two Spokes Spoke1 and Spoke2). The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. On the enterprise network, Open Shortest Path First (OSPF) is used for intra-AS routing and External Border Gateway Protocol (EBGP) is used for inter-AS routing.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-20 Networking diagram for the Shortcut DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. BGP is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-20](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure reachable routes between the ASs.

Configure OSPF to implement reachable routes between Hub and Spokes that are located in different ASs.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

NOTE

The BGP configuration on a Spoke subnet is given as an example. Perform the same configuration on other Spoke subnets.

When the subnet of a branch changes, you only need to configure the dynamic routing policy on the local device.

Step 4 Configure Basic EBGP Functions

Configure Hub.

```
[Hub] bgp 100
[Hub-bgp] router-id 172.16.1.1
[Hub-bgp] import-route ospf 1
[Hub-bgp] peer 172.16.1.2 as-number 200
[Hub-bgp] peer 172.16.1.3 as-number 300
[Hub-bgp] aggregate 192.168.0.0 16 detail-suppressed
[Hub-bgp] quit
```

NOTE

When configuring route summarization, the specified summarized address must exist on the local device. Therefore, a LoopBack address must be configured.

Configure Spoke1.

```
[Spoke1] bgp 200
[Spoke1-bgp] router-id 172.16.1.2
[Spoke1-bgp] import-route ospf 1
[Spoke1-bgp] peer 172.16.1.1 as-number 100
[Spoke1-bgp] quit
```

Configure Spoke2.

```
[Spoke2] bgp 300
[Spoke2-bgp] router-id 172.16.1.3
[Spoke2-bgp] import-route ospf 1
[Spoke2-bgp] peer 172.16.1.1 as-number 100
[Spoke2-bgp] quit
```

Step 5 Configure tunnel interfaces.

Configure route attributes on Hub and Spokes to ensure that the routes from the Spokes to Hub are reachable. Enable the NHRP redirect function on Hub. Configure NHRP mapping entries of Hub and enable the NHRP shortcut function on Spoke1 and Spoke2.

NOTE

In the shortcut scenario, configure BGP and set relevant attributes in the BGP view.

On Hub, configure a tunnel interface and enable the NHRP redirect function.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] nhrp redirect
[Hub-Tunnel0/0/0] quit
```

On Spoke1, configure a tunnel interface and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] quit
```

Step 6 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3    registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
Number of nhrp peers: 2
```

Step 7 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:46:35
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
192.168.2.1   32    202.1.3.10     172.16.1.3     remote-network up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2     local          up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:28
Expire time    : 01:59:32
-----
Number of nhrp peers: 4
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub            up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:43:32
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
192.168.1.1   32    202.1.2.10     172.16.1.2     remote-network up
-----
Tunnel interface: Tunnel0/0/0
```

```

Created time      : 00:00:47
Expire time      : 01:59:13
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.2      32   202.1.2.10     172.16.1.2    remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.3      32   202.1.3.10     172.16.1.3    local          up
-----
Tunnel interface: Tunnel0/0/0
Created time     : 00:00:47
Expire time     : 01:59:13

Number of nhrp peers: 4
  
```

----End

Configuration Files

- Hub configuration file

```

#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp redirect
 nhrp entry multicast dynamic
#
bgp 100
 router-id 172.16.1.1
 peer 172.16.1.2 as-number 200
 peer 172.16.1.3 as-number 300
#
 ipv4-family unicast
  undo synchronization
  aggregate 192.168.0.0 255.255.0.0 detail-suppressed
  import-route ospf 1
  peer 172.16.1.2 enable
  peer 172.16.1.3 enable
#
ospf 1 router-id 172.16.1.1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
#
return
  
```

- Spoke1 configuration file

```

#
sysname Spoke1
  
```

```
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp shortcut
 nhrp entry 172.16.1.1 202.1.1.10 register
#
bgp 200
 router-id 172.16.1.2
 peer 172.16.1.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 172.16.1.1 enable
#
ospf 1 router-id 172.16.1.2
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
ospf 2 router-id 202.1.2.10
 area 0.0.0.1
  network 202.1.2.0 0.0.0.255
#
return
```

● Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
 ip address 202.1.3.10 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.3 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 nhrp shortcut
 nhrp entry 172.16.1.1 202.1.1.10 register
#
bgp 300
 router-id 172.16.1.3
 peer 172.16.1.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 172.16.1.1 enable
#
ospf 1 router-id 172.16.1.3
 area 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
 area 0.0.0.1
  network 202.1.3.0 0.0.0.255
#
return
```

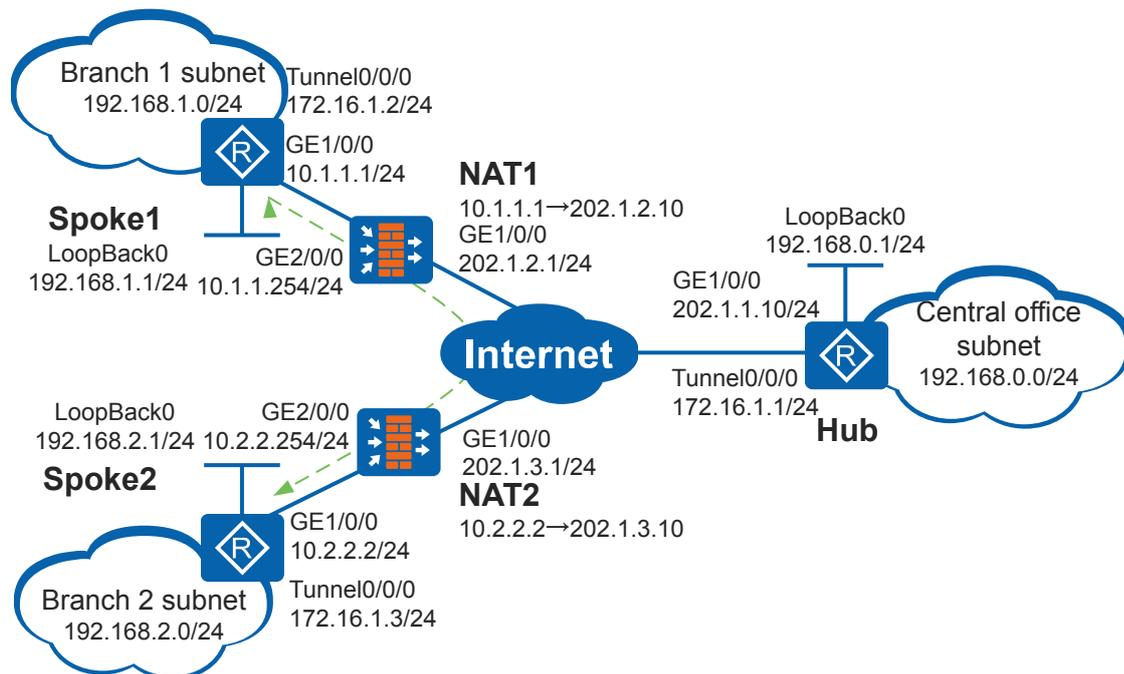
5.8.8 Example for Configuring DSVPN NAT traversal

Networking Requirements

An enterprise has a central office (Hub) and multiple branches which are located in different areas (this example shows only two Spokes Spoke1 and Spoke2). The subnets of the branches frequently change. The Spokes use addresses translated by NAT devices to connect to the public network. Open Shortest Path First (OSPF) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes.

Figure 5-21 Networking diagram for DSVPN NAT traversal configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a translated address to connect to the public network, it does not know the translated public address of the other Spoke. DSVPN NAT traversal is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. OSPF is deployed to realize communication between the Hub and Spokes and to simplify maintenance.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-21](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on NAT1.

```
[NAT1] ospf 2 router-id 202.1.2.1
[NAT1] import-route unr
[NAT1-ospf-2] area 0.0.0.1
[NAT1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[NAT1-ospf-2-area-0.0.0.1] network 10.1.1.0 0.0.0.255
[NAT1-ospf-2-area-0.0.0.1] quit
[NAT1-ospf-2] quit
```

Configure OSPF on NAT2.

```
[NAT2] ospf 2 router-id 202.1.3.1
[NAT2] import-route unr
[NAT2-ospf-2] area 0.0.0.1
[NAT2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[NAT2-ospf-2-area-0.0.0.1] network 10.2.2.0 0.0.0.255
[NAT2-ospf-2-area-0.0.0.1] quit
[NAT2-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 10.1.1.1
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 10.1.1.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 102.2.2.2
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 10.2.2.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure NAT.

Configure addresses before and after NAT traversal.

Configure NAT1.

```
[NAT1] interface GigabitEthernet 1/0/0
[NAT1-GigabitEthernet1/0/0] nat server global 202.1.2.10 inside 10.1.1.1
```

Configure NAT2.

```
[NAT2] interface GigabitEthernet 1/0/0
[NAT2-GigabitEthernet1/0/0] nat server global 202.1.3.10 inside 10.2.2.2
```

NOTE

The NAT devices must be configured with an NAT server or static NAT. NAT traversal cannot be implemented if outbound NAT is configured on the NAT devices.

Step 4 Configure the basic OSPF functions.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 5 Configure tunnel interfaces.

Configure the OSPF network type to Point-to-Multipoint (P2MP) on Hub and Spokes. Enable the NHRP redirect function on Hub. Configure NHRP mapping entries of Hub and enable the NHRP shortcut function on Spoke1 and Spoke2.

On Hub, configure a tunnel interface, configure OSPF, and enable the NHRP redirect function.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] ospf network-type p2mp
[Hub-Tunnel0/0/0] nhrp redirect
[Hub-Tunnel0/0/0] quit
```

On Spoke1, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and enable the NHRP shortcut function.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] quit
```

Step 6 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:07:55
Expire time     : --
Number of nhrp peers: 1
```

 **NOTE**

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

On Hub, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Hub. The command output is as follows:

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.1.2.10     172.16.1.2   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:02:02
Expire time     : 01:57:58
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3   registered up|unique
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:01:53
Expire time     : 01:59:35
```

```
Number of nhrp peers: 2
```

Step 7 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/2 ms
```

Run the **display nhrp peer all** command on Spoke1. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type          Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1     hub           up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:39:32
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type          Flag
-----
192.168.2.1    32    202.1.3.10     172.16.1.3     remote-network up
-----
Tunnel interface: Tunnel0/0/0
Before NAT NBMA-addr: 10.2.2.2
Created time   : 00:00:13
Expire time    : 01:59:47
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type          Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     remote        up
-----
Tunnel interface: Tunnel0/0/0
Before NAT NBMA-addr: 10.2.2.2
Created time   : 00:00:13
Expire time    : 01:59:47
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type          Flag
-----
192.168.1.1    32    10.1.1.1        172.16.1.2     local         up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:13
Expire time    : 01:59:47

Number of nhrp peers: 4
```

Run the **display nhrp peer all** command on Spoke2. The command output is as follows:

```
[Spoke2] display nhrp peer all
```

Protocol-addr	Mask	NBMA-addr	NextHop-addr	Type	Flag
172.16.1.1	32	202.1.1.10	172.16.1.1	hub	up

```
Tunnel interface: Tunnel0/0/0
Created time      : 00:41:08
Expire time       : --
```

Protocol-addr	Mask	NBMA-addr	NextHop-addr	Type	Flag
192.168.1.1	32	202.1.2.10	172.16.1.2	remote-network	up

```
Tunnel interface: Tunnel0/0/0
Before NAT NBMA-addr: 10.1.1.1
Created time      : 00:00:52
Expire time       : 01:59:08
```

Protocol-addr	Mask	NBMA-addr	NextHop-addr	Type	Flag
172.16.1.2	32	202.1.2.10	172.16.1.2	remote	up

```
Tunnel interface: Tunnel0/0/0
Before NAT NBMA-addr: 10.1.1.1
Created time      : 00:00:52
Expire time       : 01:59:08
```

Protocol-addr	Mask	NBMA-addr	NextHop-addr	Type	Flag
192.168.2.1	32	10.2.2.2	172.16.1.3	local	up

```
Tunnel interface: Tunnel0/0/0
Created time      : 00:00:52
Expire time       : 01:59:08
```

Number of nhrp peers: 4

---End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type p2mp
 nhrp redirect
 nhrp entry multicast dynamic
#
ospf 1 router-id 172.16.1.1
 area 0.0.0.0
  network 172.16.1.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
 area 0.0.0.1
  network 202.1.1.0 0.0.0.255
```

```
#  
return
```

- Spoke1 configuration file

```
#  
sysname Spoke1  
#  
interface GigabitEthernet1/0/0  
ip address 10.1.1.1 255.255.255.0  
#  
interface LoopBack0  
ip address 192.168.1.1 255.255.255.0  
#  
interface Tunnel0/0/0  
ip address 172.16.1.2 255.255.255.0  
tunnel-protocol gre p2mp  
source GigabitEthernet1/0/0  
ospf network-type p2mp  
nhrp shortcut  
nhrp entry 172.16.1.1 202.1.1.10 register  
#  
ospf 1 router-id 172.16.1.2  
area 0.0.0.0  
network 192.168.1.0 0.0.0.255  
network 172.16.1.0 0.0.0.255  
#  
ospf 2 router-id 10.1.1.1  
area 0.0.0.1  
network 10.1.1.0 0.0.0.255  
#  
return
```

- Spoke2 configuration file

```
#  
sysname Spoke2  
#  
interface GigabitEthernet1/0/0  
ip address 10.2.2.2 255.255.255.0  
#  
interface LoopBack0  
ip address 192.168.2.1 255.255.255.0  
#  
interface Tunnel0/0/0  
ip address 172.16.1.3 255.255.255.0  
tunnel-protocol gre p2mp  
source GigabitEthernet1/0/0  
ospf network-type p2mp  
nhrp shortcut  
nhrp entry 172.16.1.1 202.1.1.10 register  
#  
ospf 1 router-id 172.16.1.3  
area 0.0.0.0  
network 192.168.2.0 0.0.0.255  
network 172.16.1.0 0.0.0.255  
#  
ospf 2 router-id 10.2.2.2  
area 0.0.0.1  
network 10.2.2.0 0.0.0.255  
#  
return
```

- NAT1 configuration file

```
#  
sysname NAT1  
#  
interface GigabitEthernet1/0/0  
ip address 202.1.2.1 255.255.255.0  
nat server global 202.1.2.10 inside 10.1.1.1  
#  
interface GigabitEthernet2/0/0
```

```
ip address 10.1.1.254 255.255.255.0
#
ospf 2 router-id 202.1.2.1
import-route unr
area 0.0.0.1
 network 10.1.1.0 0.0.0.255
 network 202.1.2.0 0.0.0.255
#
return
```

- NAT2 configuration file

```
#
sysname NAT2
#
interface GigabitEthernet1/0/0
 ip address 202.1.3.1 255.255.255.0
 nat server global 202.1.3.10 inside 10.2.2.2
#
interface GigabitEthernet2/0/0
 ip address 10.2.2.254 255.255.255.0
#
ospf 2 router-id 202.1.3.1
import-route unr
area 0.0.0.1
 network 10.2.2.0 0.0.0.255
 network 202.1.3.0 0.0.0.255
#
return
```

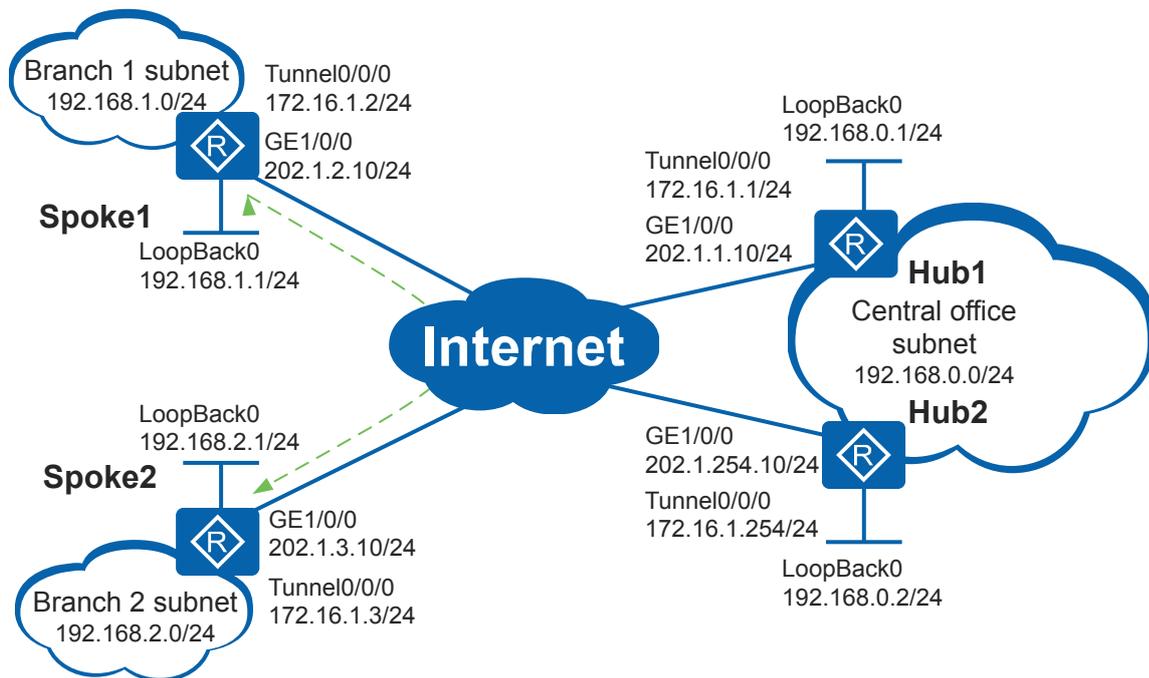
5.8.9 Example for Configuring Dual Hubs in Active/Standby Mode

Networking Requirements

A large-scale enterprise has a central office (Hub1 and Hub2) and multiple branches which are located in different areas (this example shows only two Spokes Spoke1 and Spoke2). The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Open Shortest Path First (OSPF) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes. Hub1 functions as the master device and Hub2 functions as the backup device. Hub2 takes over the services and forwards protocol packets if Hub1 fails. When Hub1 recovers, services are switched back to Hub1.

Figure 5-22 Networking diagram for dual-Hub DSVPN configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. OSPF is deployed to realize communication between the Hub and Spokes and to simplify maintenance.
4. Dual-Hub DSVPN is implemented to provide redundant backup by using Hub2.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub1.

```
<Huawei> system-view
[Huawei] sysname Hub1
[Hub1] interface GigabitEthernet 1/0/0
[Hub1-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
[Hub1-GigabitEthernet1/0/0] quit
[Hub1] interface tunnel 0/0/0
[Hub1-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub1-Tunnel0/0/0] quit
[Hub1] interface loopback 0
[Hub1-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub1-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1, Spoke2 and Hub2 as shown in [Figure 5-22](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub1.

```
[Hub1] ospf 2 router-id 202.1.1.10
[Hub1-ospf-2] area 0.0.0.1
[Hub1-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub1-ospf-2-area-0.0.0.1] quit
[Hub1-ospf-2] quit
```

Configure OSPF on Hub2.

```
[Hub2] ospf 2 router-id 202.1.254.10
[Hub2-ospf-2] area 0.0.0.1
[Hub2-ospf-2-area-0.0.0.1] network 202.1.254.0 0.0.0.255
[Hub2-ospf-2-area-0.0.0.1] quit
[Hub2-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure basic OSPF functions.

Configure Hub1.

```
[Hub1] ospf 1 router-id 172.16.1.1
[Hub1-ospf-1] area 0.0.0.0
[Hub1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub1-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub1-ospf-1-area-0.0.0.0] quit
[Hub1-ospf-1] quit
```

Configure the basic OSPF functions on Hub2.

```
[Hub2] ospf 1 router-id 172.16.1.254
[Hub2-ospf-1] area 0.0.0.0
[Hub2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub2-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub2-ospf-1-area-0.0.0.0] quit
[Hub2-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 4 Configure tunnel interfaces.

Configure the OSPF network type to Point-to-Multipoint (P2MP) on Hubs and Spokes. Enable the NHRP redirect function on Hub1 and Hub2. Configure NHRP mapping entries of Hubs and enable the NHRP shortcut function on Spoke1 and Spoke2.

Configure a tunnel interface and OSPF on Hub1 and enable the NHRP redirect function.

```
[Hub1] interface tunnel 0/0/0
[Hub1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub1-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub1-Tunnel0/0/0] ospf network-type p2mp
[Hub1-Tunnel0/0/0] ospf cost 1000
[Hub1-Tunnel0/0/0] nhrp redirect
[Hub1-Tunnel0/0/0] quit
```

Configure a tunnel interface and OSPF on Hub2 and enable the NHRP redirect function.

```
[Hub2] interface tunnel 0/0/0
[Hub2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub2-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub2-Tunnel0/0/0] ospf network-type p2mp
[Hub2-Tunnel0/0/0] ospf cost 3000
[Hub2-Tunnel0/0/0] nhrp redirect
[Hub2-Tunnel0/0/0] quit
```

Configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hubs on Spoke1, and enable the NHRP shortcut function.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.254 202.1.254.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] nhrp registration interval 300
[Spoke1-Tunnel0/0/0] quit
```

Configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hubs on Spoke2, and enable the NHRP shortcut function.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.254 202.1.254.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] nhrp registration interval 300
[Spoke2-Tunnel0/0/0] quit
```

NOTE

- Configure different OSPF cost values on Hub1 and Hub2 to ensure that the Spokes prefer Hub1 as the next hop device.
- When Hub1 recovers, it restarts to forward OSPF protocol packets when receiving NHRP Registration Request packets from Spokes. The Spokes learn routes to Hub1 after the routes they have already learned are aged out. Set the interval for sending NHRP Registration Request packets to a proper value to ensure that the Spokes can quick detect Hub1 recovery. The interval is set to 1800 seconds by default.

Step 5 Verify the configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke and Hub. Take Spoke1 as an example.

Run the **display nhrp peer all** command on Spoke1.

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub      up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 05:35:50
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.254   32    202.1.254.10   172.16.1.254  hub      up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 04:32:49
Expire time     : --
-----
Number of nhrp peers: 2
```

NOTE

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

Step 6 Run the **ping** command and check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke. Take Spoke1 as an example.

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms
```

Run the **display nhrp peer all** command on Spoke1.

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub      up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 05:42:50
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type      Flag
-----
172.16.1.254   32    202.1.254.10   172.16.1.254  hub      up
-----
```

```
-----
Tunnel interface: Tunnel0/0/0
Created time    : 04:39:49
Expire time     : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
192.168.2.1    32    202.1.3.10     172.16.1.3     remote-network up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:19
Expire time     : 01:59:41
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3     remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:19
Expire time     : 01:59:41
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
192.168.1.1    32    202.1.2.10     172.16.1.2     local         up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:00:19
Expire time     : 01:59:41
-----
Number of nhrp peers: 5
-----
```

Step 7 Shutdown the physical interface GE1/0/0 of Hub1.

Run the **shutdown** command on GE1/0/0 of Hub1.

```
[Hub1] interface GigabitEthernet 1/0/0
[Hub1-GigabitEthernet1/0/0] shutdown
[Hub1-GigabitEthernet1/0/0] quit
```

Step 8 Run the **ping** command and check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.



NOTICE

Before you run the **ping** command, ensure that no default route to Hub1 exists on the local device.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1.

```
[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/2/2 ms
```

Run the **display nhrp peer all** command on Spoke. Take Spoke1 as an example.

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1   hub       down
-----
Tunnel interface: Tunnel0/0/0
Created time   : 05:46:29
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.254   32    202.1.254.10   172.16.1.254 hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 04:43:28
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
192.168.2.1    32    202.1.3.10     172.16.1.3   remote-network up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:22
Expire time    : 01:59:38
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.3     32    202.1.3.10     172.16.1.3   remote     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:22
Expire time    : 01:59:38
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
192.168.1.1    32    202.1.2.10     172.16.1.2   local     up
-----
Tunnel interface: Tunnel0/0/0
Created time   : 00:00:22
Expire time    : 01:59:38
-----
Number of nhrp peers: 5
```

NOTE

Run the **undo nhrp peer** command to clear the NHRP mapping entries existing on the Spokes before running the **ping** command.

---End

Configuration Files

- Hub1 configuration file

```
#
sysname Hub1
#
interface GigabitEthernet1/0/0
ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.1 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
```

```
ospf cost 1000
ospf network-type p2mp
nhrp redirect
nhrp entry multicast dynamic
#
ospf 1 router-id 172.16.1.1
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
area 0.0.0.1
network 202.1.1.0 0.0.0.255
#
return
```

● Hub2 configuration file

```
#
sysname Hub2
#
interface GigabitEthernet1/0/0
ip address 202.1.254.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.0.2 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.254 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf cost 3000
ospf network-type p2mp
nhrp redirect
nhrp entry multicast dynamic
#
ospf 1 router-id 172.16.1.254
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.254.10
area 0.0.0.1
network 202.1.254.0 0.0.0.255
#
return
```

● Spoke1 configuration file

```
#
sysname Spoke1
#
interface GigabitEthernet1/0/0
ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.2 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type p2mp
nhrp shortcut
nhrp registration interval 300
nhrp entry 172.16.1.254 202.1.254.10 register
nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.2
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.1.0 0.0.0.255
```

```
#
ospf 2 router-id 202.1.2.10
 area 0.0.0.1
  network 202.1.2.0 0.0.0.255
#
return
```

- Spoke2 configuration file

```
#
sysname Spoke2
#
interface GigabitEthernet1/0/0
 ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.3 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type p2mp
 nhrp shortcut
 nhrp registration interval 300
 nhrp entry 172.16.1.254 202.1.254.10 register
 nhrp entry 172.16.1.1 202.1.1.10 register
#
ospf 1 router-id 172.16.1.3
 area 0.0.0.0
  network 172.16.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
 area 0.0.0.1
  network 202.1.3.0 0.0.0.255
#
return
```

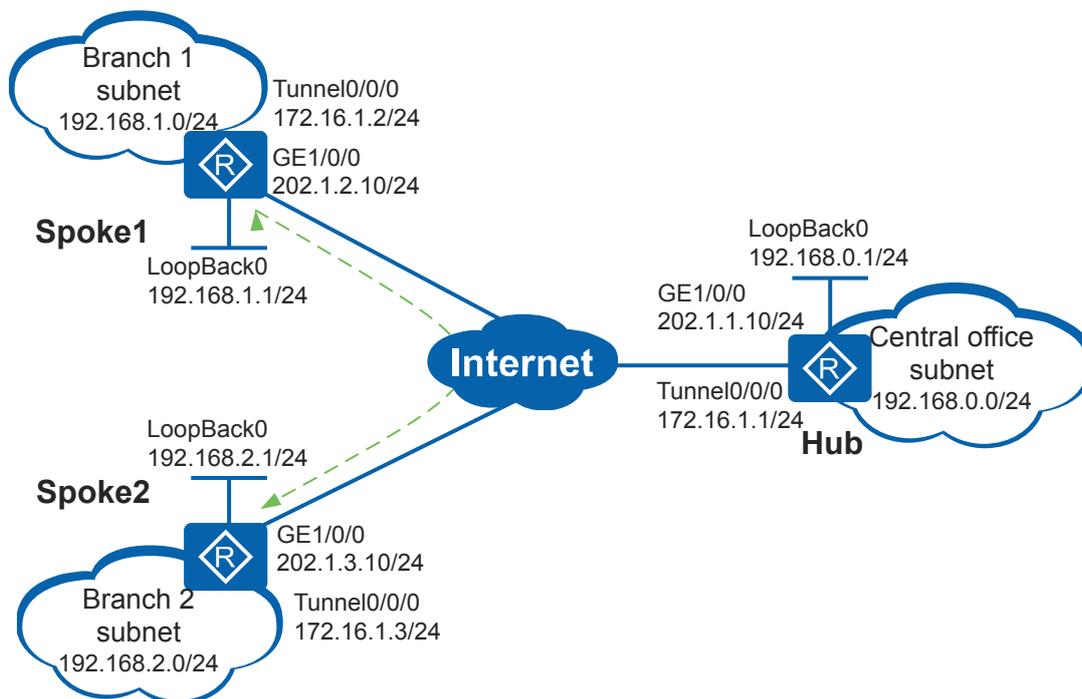
5.8.10 Example for Configuring DSVPN Protected by IPSec

Networking Requirements

A large-scale enterprise has a central office (Hub) and multiple branches which are located in different areas (this example shows only two Spokes Spoke1 and Spoke2). The networks of the central office and branches frequently change. The Spokes use dynamic addresses to connect to the public network. Open Shortest Path First (OSPF) is used on the enterprise network.

The enterprise wants to establish a VPN between the Spokes and encrypt data transmitted between the Hub and Spokes, and between Spokes to increase data security.

Figure 5-23 Networking diagram for DSVPN protected by IPSec configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Because a Spoke uses a dynamic address to connect to the public network, it does not know the public IP address of the other Spoke. DSVPN is implemented to establish a VPN between the Spokes.
2. Shortcut Scenario of DSVPN is implemented because the enterprise has a large number of branches.
3. The networks of the central office and branches frequently change. OSPF is deployed to realize communication between the Hub and Spokes and to simplify maintenance.
4. DSVPN protected by IPSec is implemented to encrypt data transmitted between the central office and branches, and between branches.

NOTE

When you deploy IPSec on a DSVPN network, the IPSec encapsulation mode can only be transport if two branches are connected to different NAT devices or the headquarters is connected to a NAT device.

Procedure

Step 1 Assign an IP address to each interface.

Configure IP addresses for the interfaces of each Router.

Configure IP addresses for interfaces of Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface GigabitEthernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 202.1.1.10 255.255.255.0
```

```
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 172.16.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
[Hub] interface loopback 0
[Hub-LoopBack0] ip address 192.168.0.1 255.255.255.0
[Hub-LoopBack0] quit
```

Configure IP addresses for interfaces of the Spoke1 and Spoke2 as shown in [Figure 5-23](#). The specific configuration is not mentioned here.

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to provide reachable routes to the public network.

Configure OSPF on Hub.

```
[Hub] ospf 2 router-id 202.1.1.10
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 202.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 202.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 202.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 202.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 202.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure the basic OSPF functions.

Configure Hub.

```
[Hub] ospf 1 router-id 172.16.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 172.16.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 172.16.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 4 Configure IKE proposals.

On Hub, Spoke1, and Spoke2, configure IKE proposals and set the same authentication mode.

Configure Hub.

```
[Hub] ike proposal 1
[Hub-ike-proposal-1] dh group5
[Hub-ike-proposal-1] authentication-algorithm sha2-256
[Hub-ike-proposal-1] prf aes-xcbc-128
[Hub-ike-proposal-1] quit
```

Configure Spoke1.

```
[Spoke1] ike proposal 1
[Spoke1-ike-proposal-1] dh group5
[Spoke1-ike-proposal-1] authentication-algorithm sha2-256
[Spoke1-ike-proposal-1] prf aes-xcbc-128
[Spoke1-ike-proposal-1] quit
```

Configure Spoke2.

```
[Spoke2] ike proposal 1
[Spoke2-ike-proposal-1] dh group5
[Spoke2-ike-proposal-1] authentication-algorithm sha2-256
[Spoke2-ike-proposal-1] prf aes-xcbc-128
[Spoke2-ike-proposal-1] quit
```

Step 5 Configure IKE peers.

Configure IKE peers used during IKE negotiation on Hub, Spoke1, and Spoke2.

Configure Hub.

```
[Hub] ike peer hub
[Hub-ike-peer-hub] ike-proposal 1
[Hub-ike-peer-hub] pre-shared-key cipher Huawei@1234
[Hub-ike-peer-hub] dpd type periodic
[Hub-ike-peer-hub] dpd idle-time 40
[Hub-ike-peer-hub] quit
```

Configure Spoke1.

```
[Spoke1] ike peer spoke1
[Spoke1-ike-peer-spoke1] ike-proposal 1
[Spoke1-ike-peer-spoke1] pre-shared-key cipher Huawei@1234
[Spoke1-ike-peer-spoke1] dpd type periodic
[Spoke1-ike-peer-spoke1] dpd idle-time 40
[Spoke1-ike-peer-spoke1] quit
```

Configure Spoke2.

```
[Spoke2] ike peer spoke2
[Spoke2-ike-peer-spoke2] ike-proposal 1
[Spoke2-ike-peer-spoke2] pre-shared-key cipher Huawei@1234
[Spoke2-ike-peer-spoke2] dpd type periodic
[Spoke2-ike-peer-spoke2] dpd idle-time 40
[Spoke2-ike-peer-spoke2] quit
```

Step 6 Create IPsec proposals.

Configure IPsec proposals on Hub, Spoke1, and Spoke2.

Configure Hub.

```
[Hub] ipsec proposal pro1
[Hub-ipsec-proposal-pro1] transform ah-esp
[Hub-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Hub-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Hub-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Hub-ipsec-proposal-pro1] quit
```

Configure Spoke1.

```
[Spoke1] ipsec proposal pro1
[Spoke1-ipsec-proposal-pro1] transform ah-esp
[Spoke1-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Spoke1-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke1-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Spoke1-ipsec-proposal-pro1] quit
```

Configure Spoke2.

```
[Spoke2] ipsec proposal pro1
[Spoke2-ipsec-proposal-pro1] transform ah-esp
[Spoke2-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Spoke2-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke2-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Spoke2-ipsec-proposal-pro1] quit
```

Step 7 Configure IPsec profiles.

Configure IPsec profiles on Hub, Spoke1, and Spoke2.

Configure Hub.

```
[Hub] ipsec profile profile1
[Hub-ipsec-profile-profile1] ike-peer hub
[Hub-ipsec-profile-profile1] proposal pro1
[Hub-ipsec-profile-profile1] quit
```

Configure Spoke1.

```
[Spoke1] ipsec profile profile1
[Spoke1-ipsec-profile-profile1] ike-peer spoke1
[Spoke1-ipsec-profile-profile1] proposal pro1
[Spoke1-ipsec-profile-profile1] quit
```

Configure Spoke2.

```
[Spoke2] ipsec profile profile1
[Spoke2-ipsec-profile-profile1] ike-peer spoke2
[Spoke2-ipsec-profile-profile1] proposal pro1
[Spoke2-ipsec-profile-profile1] quit
```

Step 8 Configure tunnel interfaces.

On Hub, configure a tunnel interface, configure OSPF, and apply the IPsec profile.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] ospf network-type p2mp
[Hub-Tunnel0/0/0] nhrp redirect
[Hub-Tunnel0/0/0] ipsec profile profile1
[Hub-Tunnel0/0/0] quit
```

On Spoke1, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and apply the IPsec profile.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] ipsec profile profile1
[Spoke1-Tunnel0/0/0] quit
```

On Spoke2, configure a tunnel interface, OSPF, and a static NHRP mapping entry of Hub, and apply the IPsec profile.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 172.16.1.1 202.1.1.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] ipsec profile profile1
[Spoke2-Tunnel0/0/0] quit
```

Step 9 Verify the DSVPN configuration.

After the preceding configurations are complete, check the NHRP mapping entries of Spoke1 and Spoke2.

Run the **display nhrp peer all** command on Spoke and Hub. Take Spoke1 as an example. The command output is as follows:

```
[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunnel0/0/0
Created time    : 00:10:58
Expire time     : --
Number of nhrp peers: 1
```

NOTE

If you run the **display nhrp peer all** command on Spoke1 and Spoke2, you can view only the static NHRP mapping entry of Hub.

Step 10 Verify the IPSec SA configuration.

Check the IPSec SAs generated on Hub, Spoke1, and Spoke2. Take Spoke1 as an example.

Run the **display ipsec sa** command on Spoke1. The command output is as follows:

```
[Spoke1] display ipsec sa
ipsec sa information:
=====
Interface: Tunnel0/0/0
=====

-----
IPSec profile name: "profile1"
Mode                : PROF-ISAKMP
-----

Connection ID       : 2
Encapsulation mode  : Tunnel
Tunnel local        : 202.1.2.10:500
Tunnel remote       : 202.1.1.10:500

[Outbound ESP SAs]
SPI: 2485560141 (0x9426a34d)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887426800/2652
Outpacket count     : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 107
UDP encapsulation used for NAT traversal: N

[Outbound AH SAs]
SPI: 3662509166 (0xda4d746e)
Proposal: SHA2-512-256
```

```

SA remaining key duration (bytes/sec): 1887436800/2652
Outpacket count      : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 107
UDP encapsulation used for NAT traversal: N

[Inbound AH SAs]
SPI: 833505824 (0x31ae4a20)
Proposal: SHA2-512-256
SA remaining key duration (bytes/sec): 1887436800/2652
Inpacket count      : 10
Inpacket decap count : 10
Inpacket drop count : 0
Max received sequence-number: 119
UDP encapsulation used for NAT traversal: N
Anti-replay : Enable
Anti-replay window size: 1024

[Inbound ESP SAs]
SPI: 2140030022 (0x7f8e4446)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887425168/2652
Inpacket count      : 10
Inpacket decap count : 10
Inpacket drop count : 0
Max received sequence-number: 119
UDP encapsulation used for NAT traversal: N
Anti-replay : Enable
Anti-replay window size: 1024
  
```

Step 11 Run the **ping** command to check the configuration result.

Ping 192.168.2.1 on Spoke1. You can see that Spoke1 and Spoke2 have learned dynamic NHRP mapping entries from each other.

Run the **ping -a 192.168.1.1 192.168.2.1** command on Spoke1. The command output is as follows:

```

[Spoke1] ping -a 192.168.1.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/2/3 ms
  
```

Run the **display nhrp peer all** command on Spoke. Take Spoke1 as an example. The command output is as follows:

```

[Spoke1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.1     32    202.1.1.10     172.16.1.1    hub       up
-----
Tunnel interface: Tunne10/0/0
Created time   : 00:46:35
Expire time    : --
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
192.168.2.1   32    202.1.3.10     172.16.1.3    remote-network up
  
```

```
-----
Tunnel interface: Tunnel0/0/0
Created time      : 00:00:28
Expire time      : 01:59:32
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.3      32   202.1.3.10     172.16.1.3    remote         up
-----
Tunnel interface: Tunnel0/0/0
Created time      : 00:00:28
Expire time      : 01:59:32
-----
Protocol-addr   Mask  NBMA-addr      NextHop-addr   Type           Flag
-----
172.16.1.2      32   202.1.2.10     172.16.1.2    local          up
-----
Tunnel interface: Tunnel0/0/0
Created time      : 00:00:28
Expire time      : 01:59:32

Number of nhrp peers: 4
```

Run the **display ipsec sa** command on Spoke. Take Spoke1 as an example. The command output is as follows:

```
[Spoke1] display ipsec sa
ipsec sa information:

=====
Interface: Tunnel0/0/0
=====

-----
IPSec profile name: "profile1"
Mode                : PROF-ISAKMP
-----

Connection ID       : 2
Encapsulation mode: Tunnel
Tunnel local        : 202.1.2.10:500
Tunnel remote       : 202.1.1.10:500

[Outbound ESP SAs]
SPI: 2485560141 (0x9426a34d)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887420488/2020
Outpacket count      : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 175
UDP encapsulation used for NAT traversal: N

[Outbound AH SAs]
SPI: 3662509166 (0xda4d746e)
Proposal: SHA2-512-256
SA remaining key duration (bytes/sec): 1887436800/2020
Outpacket count      : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 175
UDP encapsulation used for NAT traversal: N

[Inbound AH SAs]
SPI: 833505824 (0x31ae4a20)
Proposal: SHA2-512-256
SA remaining key duration (bytes/sec): 1887436800/2020
Inpacket count       : 10
Inpacket decap count : 10
Inpacket drop count  : 0
Max received sequence-number: 192
```

```
UDP encapsulation used for NAT traversal: N
Anti-replay : Enable
Anti-replay window size: 1024

[Inbound ESP SAs]
SPI: 2140030022 (0x7f8e4446)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887418092/2020
Inpacket count      : 10
Inpacket decap count : 10
Inpacket drop count : 0
Max received sequence-number: 192
UDP encapsulation used for NAT traversal: N
Anti-replay : Enable
Anti-replay window size: 1024

-----
IPSec profile name: "profile1"
Mode                : PROF-ISAKMP
-----

Connection ID       : 5
Encapsulation mode: Tunnel
Tunnel local        : 202.1.2.10:500
Tunnel remote       : 202.1.3.10:500

[Outbound ESP SAs]
SPI: 576349831 (0x225a6687)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887436368/3511
Outpacket count     : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 4
UDP encapsulation used for NAT traversal: N

[Outbound AH SAs]
SPI: 3363305474 (0xc877f802)
Proposal: SHA2-512-256
SA remaining key duration (bytes/sec): 1887436800/3511
Outpacket count     : 8
Outpacket encap count : 8
Outpacket drop count : 0
Max sent sequence-number: 4
UDP encapsulation used for NAT traversal: N

[Inbound AH SAs]
SPI: 3753703982 (0xdfbcfa2e)
Proposal: SHA2-512-256
SA remaining key duration (bytes/sec): 1887436800/3511
Inpacket count      : 10
Inpacket decap count : 10
Inpacket drop count : 0
Max received sequence-number: 4
UDP encapsulation used for NAT traversal: N
Anti-replay : Enable
Anti-replay window size: 1024

[Inbound ESP SAs]
SPI: 3361785078 (0xc860c4f6)
Proposal: ESP-ENCRYPT-AES-192 SHA2-512-256
SA remaining key duration (bytes/sec): 1887436368/3511
Inpacket count      : 10
Inpacket decap count : 10
Inpacket drop count : 0
Max received sequence-number: 4
UDP encapsulation used for NAT traversal: N
```

```
Anti-replay : Enable
Anti-replay window size: 1024
```

---End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
ipsec proposal prol
  transform ah-esp
  ah authentication-algorithm sha2-256
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-192
#
ike proposal 1
  encryption-algorithm
  aes-256
  dh
  group5
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf aes-xcbc-128
#
ike peer hub
  pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl0OU@DraF58J4=;%^%#
  ike-proposal 1
  dpd type periodic
  dpd idle-time 40
#
ipsec profile profile1
  ike-peer hub
  proposal prol
#
interface GigabitEthernet1/0/0
  ip address 202.1.1.10 255.255.255.0
#
interface LoopBack0
  ip address 192.168.0.1 255.255.255.0
#
interface Tunnel0/0/0
  ip address 172.16.1.1 255.255.255.0
  tunnel-protocol gre p2mp
  source GigabitEthernet1/0/0
  ospf network-type p2mp
  nhrp redirect
  nhrp entry multicast dynamic
  ipsec profile profile1
#
ospf 1 router-id 172.16.1.1
  area 0.0.0.0
    network 172.16.1.0 0.0.0.255
    network 192.168.0.0 0.0.0.255
#
ospf 2 router-id 202.1.1.10
  area 0.0.0.1
    network 202.1.1.0 0.0.0.255
#
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
```

```
#
ipsec proposal pro1
 transform ah-esp
 ah authentication-algorithm sha2-256
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-192
#
ike proposal 1
 encryption-algorithm
 aes-256
 dh
 group5
 authentication-algorithm sha2-256
 authentication-method pre-
 share
 integrity-algorithm hmac-
 sha2-256
 prf aes-xcbc-128
#
ike peer spokel
 pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iV100U@DraF58J4=;%^%#
 ike-proposal 1
 dpd type periodic
 dpd idle-time 40
#
ipsec profile profile1
 ike-peer spokel
 proposal pro1
#
interface GigabitEthernet1/0/0
 ip address 202.1.2.10 255.255.255.0
#
interface LoopBack0
 ip address 192.168.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 172.16.1.2 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type p2mp
 nhrp shortcut
 nhrp entry 172.16.1.1 202.1.1.10 register
 ipsec profile profile1
#
ospf 1 router-id 172.16.1.2
 area 0.0.0.0
 network 172.16.1.0 0.0.0.255
 network 192.168.1.0 0.0.0.255
#
ospf 2 router-id 202.1.2.10
 area 0.0.0.1
 network 202.1.2.0 0.0.0.255
#
return
```

● Spoke2 configuration file

```
#
sysname Spoke2
#
ipsec proposal pro1
 transform ah-esp
 ah authentication-algorithm sha2-256
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-192
#
ike proposal 1
 encryption-algorithm
 aes-256
 dh
 group5
```

```
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf aes-xcbc-128
#
ike peer spoke2
pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%^%#
ike-proposal 1
dpd type periodic
dpd idle-time 40
#
ipsec profile profile1
ike-peer spoke2
proposal pro1
#
interface GigabitEthernet1/0/0
ip address 202.1.3.10 255.255.255.0
#
interface LoopBack0
ip address 192.168.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 172.16.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type p2mp
nhp shortcut
nhp entry 172.16.1.1 202.1.1.10 register
ipsec profile profile1
#
ospf 1 router-id 172.16.1.3
area 0.0.0.0
network 172.16.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
ospf 2 router-id 202.1.3.10
area 0.0.0.1
network 202.1.3.0 0.0.0.255
#
return
```

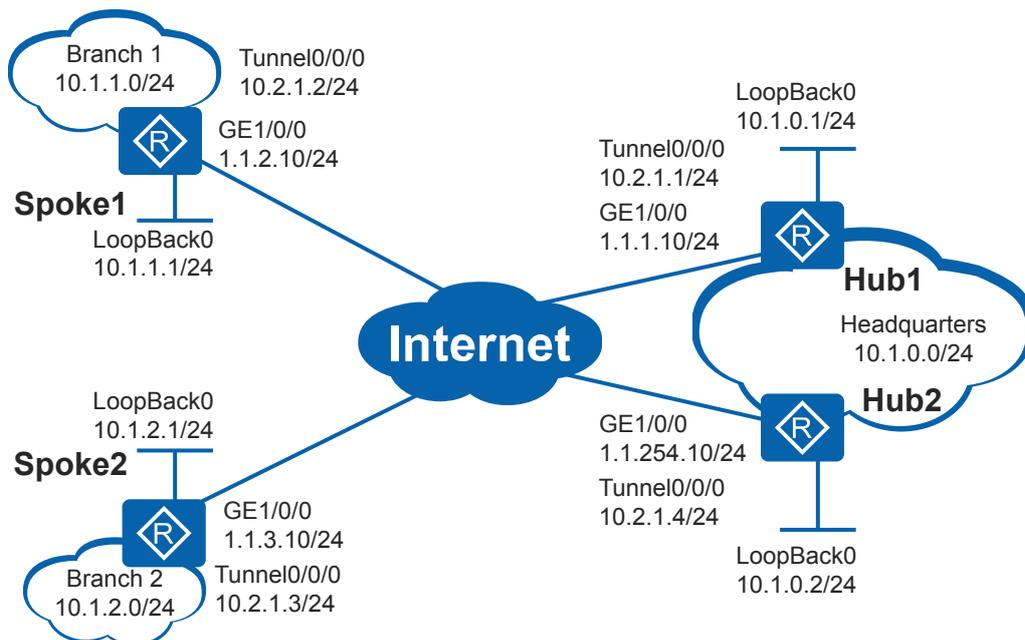
5.8.11 Example for Configuring a Dual-Hub DSVPN Protected by IPsec

Networking Requirements

In a large-size enterprise, two hubs (Hub1 and Hub2) in the headquarters communicate with multiple branches (Spoke1 and Spoke2 in this example) over the Internet. Spokes in branches use dynamic addresses to connect to the Internet.

The enterprise wants to protect traffic exchanged between the headquarters and branch and has the following requirements: Normally, the branch should communicate with the headquarters through Hub1. Traffic should be switched to Hub2 when Hub1 becomes faulty but back to Hub1 when Hub1 recovers.

Figure 5-24 Configuring a dual-hub DSVPN protected by IPsec



Configuration Roadmap

The configuration roadmap is as follows:

1. Branches use dynamic addresses to connect to the Internet; therefore, they do not know the public addresses of each other. Configure DSVPN to implement direct communication between branches.
2. Use the shortcut DSVPN because there are a large number of branches.
3. Subnets of the headquarters and branches frequently change. To simplify maintenance, configure OSPF based on the enterprise network plan to enable communication between the headquarters and branches.
4. To protect data transmitted between the headquarters and branch as well as between branches, configure IPsec for DSVPN.

Procedure

Step 1 Configure IP addresses for interfaces.

Configure IP addresses for the interfaces of the each Router. The configurations of Spoke1, Spoke2, and Hub2 are similar to that of Hub1, and are not mentioned here.

Configure an IP address for each interface on Hub1.

```
<Huawei> system-view
[Huawei] sysname Hub1
[Hub1] interface GigabitEthernet 1/0/0
[Hub1-GigabitEthernet1/0/0] ip address 1.1.1.10 255.255.255.0
[Hub1-GigabitEthernet1/0/0] quit
[Hub1] interface tunnel 0/0/0
[Hub1-Tunnel0/0/0] ip address 10.2.1.1 255.255.255.0
[Hub1-Tunnel0/0/0] quit
[Hub1] interface loopback 0
[Hub1-LoopBack0] ip address 10.1.0.1 255.255.255.0
[Hub1-LoopBack0] quit
```

Step 2 Configure routes between the Routers.

Configure OSPF on each Router to enable reachable routes over the Internet.

Configure OSPF on Hub1.

```
[Hub1] ospf 2 router-id 1.1.1.10
[Hub1-ospf-2] area 0.0.0.1
[Hub1-ospf-2-area-0.0.0.1] network 1.1.1.0 0.0.0.255
[Hub1-ospf-2-area-0.0.0.1] quit
[Hub1-ospf-2] quit
```

Configure OSPF on Hub2.

```
[Hub2] ospf 2 router-id 1.1.254.10
[Hub2-ospf-2] area 0.0.0.1
[Hub2-ospf-2-area-0.0.0.1] network 1.1.254.0 0.0.0.255
[Hub2-ospf-2-area-0.0.0.1] quit
[Hub2-ospf-2] quit
```

Configure OSPF on Spoke1.

```
[Spoke1] ospf 2 router-id 1.1.2.10
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 1.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure OSPF on Spoke2.

```
[Spoke2] ospf 2 router-id 1.1.3.10
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 1.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure basic OSPF functions.

Configure Hub1.

```
[Hub1] ospf 1 router-id 10.2.1.1
[Hub1-ospf-1] area 0.0.0.0
[Hub1-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Hub1-ospf-1-area-0.0.0.0] network 10.1.0.0 0.0.0.255
[Hub1-ospf-1-area-0.0.0.0] quit
[Hub1-ospf-1] quit
```

Configure Hub2.

```
[Hub2] ospf 1 router-id 10.2.1.4
[Hub2-ospf-1] area 0.0.0.0
[Hub2-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Hub2-ospf-1-area-0.0.0.0] network 10.1.0.0 0.0.0.255
[Hub2-ospf-1-area-0.0.0.0] quit
[Hub2-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 10.2.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 10.2.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
```

```
[Spoke2-ospf-1-area-0.0.0.0] network 10.1.2.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 4 Configure tunnel interfaces.

Set the OSPF network type to p2mp on the hubs and spokes. Enable NHRP redirect on Hub1 and Hub2. Configure static NHRP peer entries of Hub1 and Hub2 and enable NHRP shortcut on Spoke1 and Spoke2.

Configure a tunnel interface and OSPF attributes and enable NHRP redirect on Hub1.

```
[Hub1] interface tunnel 0/0/0
[Hub1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub1-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub1-Tunnel0/0/0] ospf network-type p2mp
[Hub1-Tunnel0/0/0] nhrp authentication cipher huawei@1
[Hub1-Tunnel0/0/0] gre key cipher 1999
[Hub1-Tunnel0/0/0] nhrp redirect
[Hub1-Tunnel0/0/0] quit
```

Configure a tunnel interface and OSPF attributes and enable NHRP redirect on Hub2.

```
[Hub2] interface tunnel 0/0/0
[Hub2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Hub2-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub2-Tunnel0/0/0] ospf network-type p2mp
[Hub2-Tunnel0/0/0] nhrp authentication cipher huawei@1
[Hub2-Tunnel0/0/0] nhrp redirect
[Hub2-Tunnel0/0/0] gre key cipher 1999
[Hub2-Tunnel0/0/0] quit
```

Configure tunnel interfaces, OSPF attributes, and static NHRP peer entries of Hub1 and Hub2, and enable NHRP shortcut on Spoke1.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 10.2.1.1 1.1.1.10 register
[Spoke1-Tunnel0/0/0] nhrp entry 10.2.1.4 1.1.254.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] nhrp authentication cipher huawei@1
[Spoke1-Tunnel0/0/0] nhrp shortcut
[Spoke1-Tunnel0/0/0] nhrp registration interval 300
[Spoke1-Tunnel0/0/0] gre key cipher 1999
[Spoke1-Tunnel0/0/0] quit
```

Configure tunnel interfaces, OSPF attributes, and static NHRP peer entries of Hub1 and Hub2, and enable NHRP shortcut on Spoke2.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source GigabitEthernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 10.2.1.1 1.1.1.10 register
[Spoke2-Tunnel0/0/0] nhrp entry 10.2.1.4 1.1.254.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] nhrp authentication cipher huawei@1
[Spoke2-Tunnel0/0/0] nhrp shortcut
[Spoke2-Tunnel0/0/0] nhrp registration interval 300
[Spoke2-Tunnel0/0/0] gre key cipher 1999
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Configure an IKE proposal.

Configure an IKE proposal on the hubs and spokes. Ensure that the authentication mode is the same on all the devices.

Configure Hub1.

```
[Hub1] ike proposal 1
[Hub1-ike-proposal-1] dh group5
[Hub1-ike-proposal-1] encryption-algorithm aes-256
[Hub1-ike-proposal-1] authentication-algorithm sha2-256
[Hub1-ike-proposal-1] prf aes-xcbc-128
[Hub1-ike-proposal-1] quit
```

Configure Hub2.

```
[Hub2] ike proposal 1
[Hub2-ike-proposal-1] dh group5
[Hub2-ike-proposal-1] encryption-algorithm aes-256
[Hub2-ike-proposal-1] authentication-algorithm sha2-256
[Hub2-ike-proposal-1] prf aes-xcbc-128
[Hub2-ike-proposal-1] quit
```

Configure Spoke1.

```
[Spoke1] ike proposal 1
[Spoke1-ike-proposal-1] dh group5
[Spoke1-ike-proposal-1] encryption-algorithm aes-256
[Spoke1-ike-proposal-1] authentication-algorithm sha2-256
[Spoke1-ike-proposal-1] prf aes-xcbc-128
[Spoke1-ike-proposal-1] quit
```

Configure Spoke2.

```
[Spoke2] ike proposal 1
[Spoke2-ike-proposal-1] dh group5
[Spoke2-ike-proposal-1] encryption-algorithm aes-256
[Spoke2-ike-proposal-1] authentication-algorithm sha2-256
[Spoke2-ike-proposal-1] prf aes-xcbc-128
[Spoke2-ike-proposal-1] quit
```

Step 6 Configure an IKE peer.

Configure an IKE peer for IKE negotiation on the hubs and spokes.

Configure Hub1.

```
[Hub1] ike peer hub1
[Hub1-ike-peer-hub1] undo version 2
[Hub1-ike-peer-hub1] ike-proposal 1
[Hub1-ike-peer-hub1] pre-shared-key cipher Huawei@1234
[Hub1-ike-peer-hub1] dpd type periodic
[Hub1-ike-peer-hub1] dpd idle-time 40
[Hub1-ike-peer-hub1] quit
```

Configure Hub2.

```
[Hub2] ike peer hub2
[Hub2-ike-peer-hub2] undo version 2
[Hub2-ike-peer-hub2] ike-proposal 1
[Hub2-ike-peer-hub2] pre-shared-key cipher Huawei@1234
[Hub2-ike-peer-hub2] dpd type periodic
[Hub2-ike-peer-hub2] dpd idle-time 40
[Hub2-ike-peer-hub2] quit
```

Configure Spoke1.

```
[Spoke1] ike peer spokel
[Spoke1-ike-peer-spokel] undo version 2
[Spoke1-ike-peer-spokel] ike-proposal 1
[Spoke1-ike-peer-spokel] pre-shared-key cipher Huawei@1234
[Spoke1-ike-peer-spokel] dpd type periodic
[Spoke1-ike-peer-spokel] dpd idle-time 40
[Spoke1-ike-peer-spokel] quit
```

Configure Spoke2.

```
[Spoke2] ike peer spoke2
[Spoke2-ike-peer-spoke2] undo version 2
[Spoke2-ike-peer-spoke2] ike-proposal 1
[Spoke2-ike-peer-spoke2] pre-shared-key cipher Huawei@1234
[Spoke2-ike-peer-spoke2] dpd type periodic
[Spoke2-ike-peer-spoke2] dpd idle-time 40
[Spoke2-ike-peer-spoke2] quit
```

Step 7 Create an IPsec proposal.

Create an IPsec proposal on the hubs and spokes.

Configure Hub1.

```
[Hub1] ipsec proposal prol
[Hub1-ipsec-proposal-pro1] transform ah-esp
[Hub1-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Hub1-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Hub1-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Hub1-ipsec-proposal-pro1] quit
```

Configure Hub2.

```
[Hub2] ipsec proposal prol
[Hub2-ipsec-proposal-pro1] transform ah-esp
[Hub2-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Hub2-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Hub2-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Hub2-ipsec-proposal-pro1] quit
```

Configure Spoke1.

```
[Spoke1] ipsec proposal prol
[Spoke1-ipsec-proposal-pro1] transform ah-esp
[Spoke1-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Spoke1-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke1-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Spoke1-ipsec-proposal-pro1] quit
```

Configure Spoke2.

```
[Spoke2] ipsec proposal prol
[Spoke2-ipsec-proposal-pro1] transform ah-esp
[Spoke2-ipsec-proposal-pro1] ah authentication-algorithm sha2-256
[Spoke2-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke2-ipsec-proposal-pro1] esp encryption-algorithm aes-192
[Spoke2-ipsec-proposal-pro1] quit
```

Step 8 Create an IPsec profile.

Create an IPsec profile on the hubs and spokes.

Configure Hub1.

```
[Hub1] ipsec profile profile1
[Hub1-ipsec-profile-profile1] ike-peer hub1
[Hub1-ipsec-profile-profile1] proposal prol
[Hub1-ipsec-profile-profile1] quit
```

Configure Hub2.

```
[Hub2] ipsec profile profile1
[Hub2-ipsec-profile-profile1] ike-peer hub2
[Hub2-ipsec-profile-profile1] proposal prol
[Hub2-ipsec-profile-profile1] quit
```

Configure Spoke1.

```
[Spoke1] ipsec profile profile1
[Spoke1-ipsec-profile-profile1] ike-peer spoke1
```

```
[Spoke1-ipsec-profile-profile1] proposal prol  
[Spoke1-ipsec-profile-profile1] quit
```

Configure Spoke2.

```
[Spoke2] ipsec profile profile1  
[Spoke2-ipsec-profile-profile1] ike-peer spoke2  
[Spoke2-ipsec-profile-profile1] proposal prol  
[Spoke2-ipsec-profile-profile1] quit
```

Step 9 Apply the IPsec profile to interfaces.

Configure Hub1.

```
[Hub1] interface tunnel 0/0/0  
[Hub1-Tunnel0/0/0] ipsec profile profile1  
[Hub1-Tunnel0/0/0] quit
```

Configure Hub2.

```
[Hub2] interface tunnel 0/0/0  
[Hub2-Tunnel0/0/0] ipsec profile profile1  
[Hub2-Tunnel0/0/0] quit
```

Configure Spoke1.

```
[Spoke1] interface tunnel 0/0/0  
[Spoke1-Tunnel0/0/0] ipsec profile profile1  
[Spoke1-Tunnel0/0/0] quit
```

Configure Spoke2.

```
[Spoke2] interface tunnel 0/0/0  
[Spoke2-Tunnel0/0/0] ipsec profile profile1  
[Spoke2-Tunnel0/0/0] quit
```

Step 10 Verify the configuration.

The headquarters and branch as well as branches can communicate with each other, and data flows between them are protected by IPsec.

1. Check whether IKE SAs are established.

Run the **display ike sa** command to check whether IKE SAs are established. The command output on Hub1 and Spoke1 is used as an example.

```
[Spoke1] display ike sa  
Conn-ID Peer VPN Flag(s)  
Phase  
-----  
v1:2 442 1.1.1.10 0 RD|ST  
v1:1 138 1.1.1.10 0 RD|ST  
v1:2 409 1.1.254.10 0 RD|ST  
v1:1 5 1.1.254.10 0 RD|ST  
Number of IKE SA : 4  
-----  
Flag Description:  
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT  
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP  
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

You can find that Spoke1 establishes IPsec tunnels with Hub1 and Hub2 successfully.

Run the **ping -a 10.1.1.1 10.1.2.1** command on Spoke1, and the command output is as follows.

```
[Spoke1] ping -a 10.1.1.1 10.1.2.1
PING 10.1.2.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 10.1.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.1.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms
[Spoke1] display ike sa
Conn-ID Peer          VPN  Flag(s)
Phase
-----
v1:2    442    1.1.1.10          0    RD|ST
v1:1    138    1.1.1.10          0    RD|ST
v1:2    342    1.1.3.10          0    RD|ST
v1:1    284    1.1.3.10          0    RD|ST
v1:2    409    1.1.254.10        0    RD|ST
v1:1     5     1.1.254.10        0    RD|ST

Number of IKE SA : 6
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

When branches communicate with each other, Spoke1 and Spoke2 establish an IPsec tunnel.

- When Hub1 fails, the headquarters and branch as well as branches can still communicate with each other.

Run the **ping -a 10.1.1.1 10.1.2.1** command on Spoke1, and the command output is as follows.

```
[Spoke1] ping -a 10.1.1.1 10.1.2.1
PING 10.1.2.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.2.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 10.1.2.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.1.2.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.1.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms
```

----End

Configuration Files

- Hub1 configuration file

```
#
sysname Hub1
```

```
#
ipsec proposal
prol
  transform ah-
  esp
  ah authentication-algorithm
  sha2-256
  esp authentication-algorithm
  sha2-256
  esp encryption-algorithm
  aes-192
#

ike proposal
1
  encryption-algorithm aes-256
  dh
  group5
  authentication-algorithm
  sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf aes-xcbc-128
#

ike peer hub1
undo version 2
pre-shared-key cipher %%#r]yCG7r(%0be2oGBu,[XG'[76vVusGq|D9KF,7K@%^%#
ike-proposal
1
  dpd type
  periodic
  dpd idle-time
  40
#

ipsec profile
profile1
  ike-peer
  hub1
  proposal
  prol
#

interface GigabitEthernet1/0/0
ip address 1.1.1.10 255.255.255.0
#

interface LoopBack0
ip address 10.1.0.1 255.255.255.0
#

interface Tunnel0/0/0
ip address 10.2.1.1 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
gre key cipher %%#q'~GF"30`<g3mxV46;`!_&1{>'e5ALQLkU6~+T>C%#%#
ospf network-type p2mp
ipsec profile profile1
nhrp authentication cipher %%#!Noa/<I+/WhpAwVfx`QI=vcV),t#@Ihg=PQeN]%C%#%#
nhrp redirect
nhrp entry multicast dynamic
#

ospf 1 router-id 10.2.1.1
area 0.0.0.0
  network 10.2.1.0 0.0.0.255
  network 10.1.0.0 0.0.0.255
#

ospf 2 router-id 1.1.1.10
area 0.0.0.1
  network 1.1.1.0 0.0.0.255
```

```
#
return
● Hub2 configuration file
#
sysname Hub2
#
ipsec proposal
prol
  transform ah-
  esp
    ah authentication-algorithm
    sha2-256
    esp authentication-algorithm
    sha2-256
    esp encryption-algorithm aes-192
#
ike proposal
1
  encryption-algorithm aes-256
  dh
  group5
  authentication-algorithm
  sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf aes-xcbc-128
#
ike peer hub2
  undo version 2
  pre-shared-key cipher %%#W8t$Ji82`Y-RX')iNvw9dZ3.K8bxvKioU4LNKx*7%^%#
  ike-proposal
  1
  dpd type
  periodic
  dpd idle-time
  40
#
ipsec profile
profile1
  ike-peer
  hub2
  proposal
  prol
#
interface GigabitEthernet1/0/0
  ip address 1.1.254.10 255.255.255.0
#
interface LoopBack0
  ip address 10.1.0.2 255.255.255.0
#
interface Tunnel0/0/0
  ip address 10.2.1.4 255.255.255.0
  tunnel-protocol gre
p2mp
  source GigabitEthernet1/0/0
  gre key cipher %%#[*8)P`Ra>LdAI7Hamn2t=W5D$M]kMjMEH:9^tr-%%#
  ospf network-type
p2mp
  ipsec profile
  profile1
  nhrp authentication cipher %%#T(U)=!7|/2^zbH",\BxIKTySV/5xQ*n+<U,dc!36%^%#
  nhrp redirect
  nhrp entry multicast dynamic
#
ospf 1 router-id 10.2.1.254
  area 0.0.0.0
  network 10.2.1.0 0.0.0.255
  network 10.1.0.0 0.0.0.255
#
```

```
ospf 2 router-id 1.1.254.10
 area 0.0.0.1
  network 1.1.254.0 0.0.0.255
#
return
```

● Spoke1 configuration file

```
#
sysname Spoke1
#
ipsec proposal
prol
 transform ah-
 esp
  ah authentication-algorithm
 sha2-256
  esp authentication-algorithm
 sha2-256
  esp encryption-algorithm
 aes-192
#
ike proposal
1
 encryption-algorithm aes-256
 dh
 group5
 authentication-algorithm
 sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf aes-xcbc-128
#
ike peer spoke1
undo version 2
pre-shared-key cipher %^%#yRiB!lV4gKvCG_LJ&QDF'FuTPhzX,)QVajSs&M_I%^%#
ike-proposal
1
 dpd type
 periodic
 dpd idle-time
 40
#
ipsec profile
profile1
 ike-peer
 spoke1
 proposal
 prol
#
interface GigabitEthernet1/0/0
 ip address 1.1.2.10 255.255.255.0
#
interface LoopBack0
 ip address 10.1.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 10.2.1.2
 255.255.255.0
 tunnel-protocol gre
 p2mp
 source GigabitEthernet1/0/0
 gre key cipher %^%#qi,=:z}BQCPT5D>A)20MCIEc6-SBY*d<|bE~>i;2%^%#
 ospf network-type
 p2mp
 ipsec profile
 profile1
 nhrp authentication cipher %^%#e1an+f[D*$J{NJ4ubMM$N1L1F2O6#O/u:-[EkSJ%^%#
```

```
nhrp
shortcut
nhrp registration interval
300
nhrp entry 10.2.1.1 1.1.1.10 register
nhrp entry 10.2.1.4 1.1.254.10 register
#
ospf 1 router-id 10.2.1.2
area 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.2.1.0 0.0.0.255
#
ospf 2 router-id 1.1.2.10
area 0.0.0.1
network 1.1.2.0 0.0.0.255
#
return
```

● Spoke2 configuration file

```
#
sysname Spoke2
#
ipsec proposal
pro1
transform ah-
esp
ah authentication-algorithm
sha2-256
esp authentication-algorithm
sha2-256
esp encryption-algorithm
aes-192
#
ike proposal
1
encryption-algorithm aes-256
dh
group5
authentication-algorithm
sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf aes-xcbc-128
#
ike peer spoke2
undo version 2
pre-shared-key cipher %%#yRiB!lV4gKvCG_LJ&QDF'FuTPhzX,)QVajSs&M_I%^#
ike-proposal
1
dpd type
periodic
dpd idle-time
40
#
ipsec profile
profile1
ike-peer
spoke2
proposal
pro1
#
interface GigabitEthernet1/0/0
ip address 1.1.3.10 255.255.255.0
#
interface LoopBack0
ip address 10.1.2.1 255.255.255.0
#
```

```
interface Tunnel0/0/0
 ip address 10.2.1.3
 255.255.255.0
 tunnel-protocol gre
 p2mp
 source GigabitEthernet1/0/0
 gre key cipher %^%#y0|R0B_>==#1"D) 42/nU!;A56Zx=oDj,707>#:4.%^%#
 ospf network-type
 p2mp
 ipsec profile
 profile1
 nhrp authentication cipher %^%#FosR<0omi.W{}Y7gp`XP|I-v" |]+7S>{'T/(vK00%^%#
 nhrp
 shortcut
 nhrp registration interval
 300
 nhrp entry 10.2.1.1 1.1.1.10 register
 nhrp entry 10.2.1.4 1.1.254.10 register
 #
 ospf 1 router-id 10.2.1.3
 area 0.0.0.0
 network 10.1.2.0 0.0.0.255
 network 10.2.1.0 0.0.0.255
 #
 ospf 2 router-id 1.1.3.10
 area 0.0.0.1
 network 1.1.3.0 0.0.0.255
 #
return
```

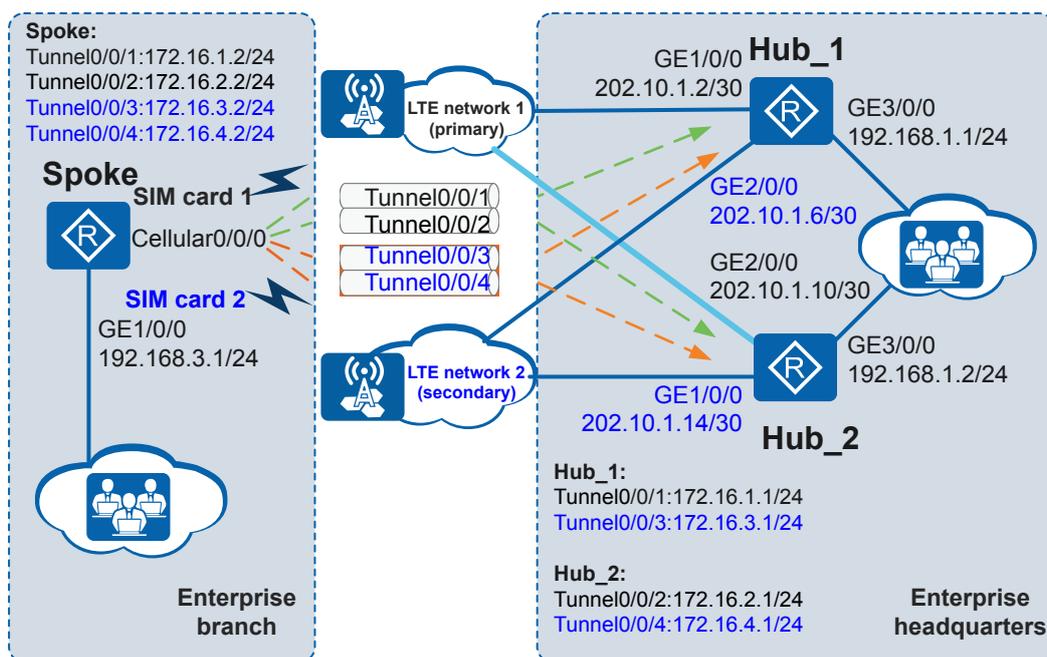
5.8.12 Example for Configuring a DSVPN Based on the LTE Dialup Status

Networking Requirements

As shown in [Figure 5-25](#), an enterprise headquarters (Hub_1 as the primary device and Hub_2 as the secondary device) and branch (Spoke) locate in different areas. The branch connects to the headquarters through an LTE network, that is, LTE network 1 shown in the figure. The enterprise requires that the branch communicate with the headquarters through a VPN and data transmitted between them be encrypted.

To ensure that the enterprise users can still connect to the headquarters even when the primary SIM card 1 or LTE network 1 is faulty, the enterprise leases the other LTE network, that is LTE network 2 shown in the figure, to set up a backup link (through the secondary SIM card 2) for temporary service transmission.

Figure 5-25 Configuring a DSVPN based on the LTE dialup status



Configuration Roadmap

The branch address is not fixed because it connects to the headquarters through an LTE network; therefore, the branch and headquarters must be connected through a VPN.

To ensure reliable data transmission, two SIM cards in redundancy mode need to be configured in the branch and they connect to different LTE networks. A tunnel can be established between the headquarters and branch based on the association between the LTE dialup status and DSVPN to ensure uninterrupted data transmission.

The configuration roadmap is as follows:

1. Configure a cellular interface and APN profile, so that the branch can connect to the LTE network.
2. Use the non-shortcut DSVPN scenario because the enterprise has only few branches. Use the RIP protocol to advertise private network routes between the headquarters and branch and associate NHRP peer information with the APN profile. When the APN profile is in use, the associated NHRP peer information takes effect; therefore, a tunnel can be established between the headquarters and branch.
3. Configure the NQA function to implement switching between the primary and secondary SIM cards.
4. Install a primary and a secondary SIM card on the cellular interface to ensure reliable data transmission.
5. Bind IPsec policies to the cellular interface on the branch device and the public network interfaces on the headquarters devices, so that data transmitted between them can be encrypted.

Procedure

Step 1 Configure IP addresses for interfaces.

Configure an IP address for each interface on Hub_1 and Hub_2 according to [Figure 5-25](#).

Configure an IP address for each interface on Hub_1.

```
<Huawei> system-view
[Huawei] sysname Hub_1
[Hub_1] interface GigabitEthernet 1/0/0
[Hub_1-GigabitEthernet1/0/0] ip address 202.10.1.2 255.255.255.252
[Hub_1-GigabitEthernet1/0/0] quit
[Hub_1] interface gigabitethernet 2/0/0
[Hub_1-GigabitEthernet2/0/0] ip address 202.10.1.6 255.255.255.252
[Hub_1-GigabitEthernet2/0/0] quit
[Hub_1] interface gigabitethernet 3/0/0
[Hub_1-GigabitEthernet3/0/0] ip address 192.168.1.1 255.255.255.0
[Hub_1-GigabitEthernet3/0/0] quit
[Hub_1] interface tunnel 0/0/1
[Hub_1-Tunnel0/0/1] ip address 172.16.1.1 255.255.255.0
[Hub_1-Tunnel0/0/1] quit
[Hub_1] interface tunnel 0/0/3
[Hub_1-Tunnel0/0/3] ip address 172.16.3.1 255.255.255.0
[Hub_1-Tunnel0/0/3] quit
```

The configurations of Hub_2 and the Spoke are similar to the configuration of Hub_1, and are not mentioned here.

Step 2 Configure a cellular interface and APN profile.

Configure the Spoke.

```
[Spoke] dialer-rule
[Spoke-dialer-rule] dialer-rule 1 ip permit
[Spoke-dialer-rule] quit
[Spoke] interface cellular 0/0/0
[Spoke-Cellular0/0/0] ip address negotiate
[Spoke-Cellular0/0/0] dialer enable-circular
[Spoke-Cellular0/0/0] dialer-group 1
[Spoke-Cellular0/0/0] dialer timer autodial 15
[Spoke-Cellular0/0/0] dialer timer probe-interval 15
[Spoke-Cellular0/0/0] dialer number *99# autodial
[Spoke-Cellular0/0/0] mode lte auto
[Spoke-Cellular0/0/0] quit
[Spoke] apn profile ltenet
[Spoke-apn-profile-ltenet] sim-id 1
[Spoke-apn-profile-ltenet] apn LTENET1
[Spoke-apn-profile-ltenet] quit
[Spoke] apn profile ltewap
[Spoke-apn-profile-ltewap] sim-id 2
[Spoke-apn-profile-ltewap] apn LTENET2
[Spoke-apn-profile-ltewap] quit
```

Step 3 Configure reachable public network routes between the devices.

Configure static routes on each device to ensure that the public network routes between the devices are reachable.

Configure Hub_1.

```
[Hub_1] ip route-static 0.0.0.0 0 202.10.1.1
[Hub_1] ip route-static 0.0.0.0 0 202.10.1.5
```

Configure Hub_2.

```
[Hub_2] ip route-static 0.0.0.0 0 202.10.1.9
[Hub_2] ip route-static 0.0.0.0 0 202.10.1.13
```

Configure the Spoke.

```
[Spoke] ip route-static 0.0.0.0 0 cellular 0/0/0
```

Step 4 Configure the DSVPN function.

Configure tunnel interfaces on the Hubs and Spoke and associate NHRP peer information with the APN profile. Configure the RIP protocol to advertise private network routes and configure the Spoke to add different metric values to the routes when different tunnel interfaces send or receive RIP packets to implement communication between the headquarters and branch.

Configure Hub_1.

```
[Hub_1] interface tunnel 0/0/1
[Hub_1-Tunnel0/0/1] tunnel-protocol gre p2mp
[Hub_1-Tunnel0/0/1] source GigabitEthernet 1/0/0
[Hub_1-Tunnel0/0/1] nhrp registration no-unique
[Hub_1-Tunnel0/0/1] nhrp entry multicast dynamic
[Hub_1-Tunnel0/0/1] gre key 111
[Hub_1-Tunnel0/0/1] nhrp authentication cipher Huawei@1
[Hub_1-Tunnel0/0/1] nhrp entry holdtime seconds 60
[Hub_1-Tunnel0/0/1] quit
[Hub_1] interface tunnel 0/0/3
[Hub_1-Tunnel0/0/3] tunnel-protocol gre p2mp
[Hub_1-Tunnel0/0/3] source gigabitethernet 2/0/0
[Hub_1-Tunnel0/0/3] nhrp registration no-unique
[Hub_1-Tunnel0/0/3] nhrp entry multicast dynamic
[Hub_1-Tunnel0/0/3] gre key 333
[Hub_1-Tunnel0/0/3] nhrp authentication cipher Huawei@3
[Hub_1-Tunnel0/0/3] nhrp entry holdtime seconds 60
[Hub_1-Tunnel0/0/3] quit
[Hub_1] rip 1
[Hub_1-rip-1] version 2
[Hub_1-rip-1] undo summary
[Hub_1-rip-1] network 172.16.0.0
[Hub_1-rip-1] network 192.168.1.0
[Hub_1-rip-1] quit
```

Configure Hub_2.

```
[Hub_2] interface tunnel 0/0/2
[Hub_2-Tunnel0/0/2] tunnel-protocol gre p2mp
[Hub_2-Tunnel0/0/2] source gigabitethernet 2/0/0
[Hub_2-Tunnel0/0/2] nhrp registration no-unique
[Hub_2-Tunnel0/0/2] nhrp entry multicast dynamic
[Hub_2-Tunnel0/0/2] gre key 222
[Hub_2-Tunnel0/0/2] nhrp authentication cipher Huawei@2
[Hub_2-Tunnel0/0/2] nhrp entry holdtime seconds 60
[Hub_2-Tunnel0/0/2] quit
[Hub_2] interface tunnel 0/0/4
[Hub_2-Tunnel0/0/4] tunnel-protocol gre p2mp
[Hub_2-Tunnel0/0/4] source GigabitEthernet 1/0/0
[Hub_2-Tunnel0/0/4] nhrp registration no-unique
[Hub_2-Tunnel0/0/4] nhrp entry multicast dynamic
[Hub_2-Tunnel0/0/4] gre key 444
[Hub_2-Tunnel0/0/4] nhrp authentication cipher Huawei@4
[Hub_2-Tunnel0/0/4] nhrp entry holdtime seconds 60
[Hub_2-Tunnel0/0/4] quit
[Hub_2] rip 1
[Hub_2-rip-1] version 2
[Hub_2-rip-1] undo summary
[Hub_2-rip-1] network 172.16.0.0
[Hub_2-rip-1] network 192.168.1.0
[Hub_2-rip-1] quit
```

Associate NHRP peer information with the APN profile on the Spoke and configure the Spoke to add different metric values to the routes when different tunnel interfaces send or receive RIP packets.

```
[Spoke] rip 1
[Spoke-rip-1] version 2
[Spoke-rip-1] network 172.16.0.0
[Spoke-rip-1] network 192.168.3.0
[Spoke-rip-1] quit
[Spoke] interface tunnel 0/0/1
```

```
[Spoke-Tunnel0/0/1] tunnel-protocol gre p2mp
[Spoke-Tunnel0/0/1] source cellular 0/0/0
[Spoke-Tunnel0/0/1] gre key 111
[Spoke-Tunnel0/0/1] nhrp authentication cipher Huawei@1
[Spoke-Tunnel0/0/1] nhrp registration interval 20
[Spoke-Tunnel0/0/1] nhrp entry 172.16.1.1 202.10.1.2 register track apn ltenet
[Spoke-Tunnel0/0/1] rip metricin 1
[Spoke-Tunnel0/0/1] quit
[Spoke] interface tunnel 0/0/2
[Spoke-Tunnel0/0/2] tunnel-protocol gre p2mp
[Spoke-Tunnel0/0/2] source cellular 0/0/0
[Spoke-Tunnel0/0/2] gre key 222
[Spoke-Tunnel0/0/2] nhrp authentication cipher Huawei@2
[Spoke-Tunnel0/0/2] nhrp registration interval 20
[Spoke-Tunnel0/0/2] nhrp entry 172.16.2.1 202.10.1.10 register track apn ltenet
[Spoke-Tunnel0/0/2] rip metricin 7
[Spoke-Tunnel0/0/2] rip metricout 7
[Spoke-Tunnel0/0/2] quit
[Spoke] interface tunnel 0/0/3
[Spoke-Tunnel0/0/3] tunnel-protocol gre p2mp
[Spoke-Tunnel0/0/3] source cellular 0/0/0
[Spoke-Tunnel0/0/3] gre key 333
[Spoke-Tunnel0/0/3] nhrp authentication cipher Huawei@3
[Spoke-Tunnel0/0/3] nhrp registration interval 20
[Spoke-Tunnel0/0/3] nhrp entry 172.16.3.1 202.10.1.6 register track apn ltewap
[Spoke-Tunnel0/0/3] rip metricin 4
[Spoke-Tunnel0/0/3] rip metricout 4
[Spoke-Tunnel0/0/3] quit
[Spoke] interface tunnel 0/0/4
[Spoke-Tunnel0/0/4] tunnel-protocol gre p2mp
[Spoke-Tunnel0/0/4] source cellular 0/0/0
[Spoke-Tunnel0/0/4] gre key 444
[Spoke-Tunnel0/0/4] nhrp authentication cipher Huawei@4
[Spoke-Tunnel0/0/4] nhrp registration interval 20
[Spoke-Tunnel0/0/4] nhrp entry 172.16.4.1 202.10.1.14 register track apn ltewap
[Spoke-Tunnel0/0/4] rip metricin 10
[Spoke-Tunnel0/0/4] rip metricout 10
[Spoke-Tunnel0/0/4] quit
```

Step 5 Configure the NQA function.

Determine whether to perform a primary/secondary SIM card switching based on the NQA detection results on tunnel interfaces and the LTE dialup status.

Configure the Spoke.

```
[Spoke] nqa test-instance admin Tunnel0/0/1
[Spoke-nqa-admin-Tunnel0/0/1] test-type icmp
[Spoke-nqa-admin-Tunnel0/0/1] destination-address ipv4 172.16.1.1
[Spoke-nqa-admin-Tunnel0/0/1] source-address ipv4 172.16.1.2
[Spoke-nqa-admin-Tunnel0/0/1] frequency 15
[Spoke-nqa-admin-Tunnel0/0/1] source-interface tunnel 0/0/1
[Spoke-nqa-admin-Tunnel0/0/1] start now
[Spoke-nqa-admin-Tunnel0/0/1] quit
[Spoke] nqa test-instance admin Tunnel0/0/2
[Spoke-nqa-admin-Tunnel0/0/2] test-type icmp
[Spoke-nqa-admin-Tunnel0/0/2] destination-address ipv4 172.16.2.1
[Spoke-nqa-admin-Tunnel0/0/2] source-address ipv4 172.16.2.2
[Spoke-nqa-admin-Tunnel0/0/2] frequency 15
[Spoke-nqa-admin-Tunnel0/0/2] source-interface tunnel 0/0/2
[Spoke-nqa-admin-Tunnel0/0/2] start now
[Spoke-nqa-admin-Tunnel0/0/2] quit
[Spoke] nqa test-instance admin Tunnel0/0/3
[Spoke-nqa-admin-Tunnel0/0/3] test-type icmp
[Spoke-nqa-admin-Tunnel0/0/3] destination-address ipv4 172.16.3.1
[Spoke-nqa-admin-Tunnel0/0/3] source-address ipv4 172.16.3.2
[Spoke-nqa-admin-Tunnel0/0/3] frequency 15
[Spoke-nqa-admin-Tunnel0/0/3] source-interface tunnel 0/0/3
[Spoke-nqa-admin-Tunnel0/0/3] start now
```

```
[Spoke-nqa-admin-Tunnel0/0/3] quit
[Spoke] nqa test-instance admin Tunnel0/0/4
[Spoke-nqa-admin-Tunnel0/0/4] test-type icmp
[Spoke-nqa-admin-Tunnel0/0/4] destination-address ipv4 172.16.4.1
[Spoke-nqa-admin-Tunnel0/0/4] source-address ipv4 172.16.4.2
[Spoke-nqa-admin-Tunnel0/0/4] frequency 15
[Spoke-nqa-admin-Tunnel0/0/4] source-interface tunnel 0/0/4
[Spoke-nqa-admin-Tunnel0/0/4] start now
[Spoke-nqa-admin-Tunnel0/0/4] quit
```

Step 6 Install a primary and a secondary SIM card on the Spoke.

Configure the Spoke.

```
[Spoke] interface cellular 0/0/0
[Spoke-Cellular0/0/0] apn-profile ltenet priority 200 track nqa admin Tunnel0/0/1
admin Tunnel0/0/2
[Spoke-Cellular0/0/0] apn-profile ltewap priority 150 track nqa admin Tunnel0/0/3
admin Tunnel0/0/4
[Spoke-Cellular0/0/0] shutdown
[Spoke-Cellular0/0/0] undo shutdown
[Spoke-Cellular0/0/0] quit
```

Step 7 Configure the IPSec function to protect data transmitted between the headquarters and branch.

Configure Hub_1.

```
[Hub_1] acl number 3001
[Hub_1-acl-adv-3001] rule 5 permit ip source 202.10.1.2 0
[Hub_1-acl-adv-3001] quit
[Hub_1] acl number 3003
[Hub_1-acl-adv-3003] rule 5 permit ip source 202.10.1.6 0
[Hub_1-acl-adv-3003] quit
[Hub_1] ipsec proposal 1
[Hub_1-ipsec-proposal-1] quit
[Hub_1] ipsec proposal 3
[Hub_1-ipsec-proposal-3] quit
[Hub_1] ike peer 1 v1
[Hub_1-ike-peer-1] pre-shared-key cipher Huawei@1234
[Hub_1-ike-peer-1] quit
[Hub_1] ike peer 3 v1
[Hub_1-ike-peer-3] pre-shared-key cipher Huawei@1234
[Hub_1-ike-peer-3] quit
[Hub_1] ipsec policy-template use1 10
[Hub_1-ipsec-policy-templet-use1-10] ike-peer 1
[Hub_1-ipsec-policy-templet-use1-10] proposal 1
[Hub_1-ipsec-policy-templet-use1-10] security acl 3001
[Hub_1-ipsec-policy-templet-use1-10] quit
[Hub_1] ipsec policy policy1 10 isakmp template use1
[Hub_1] ipsec policy-template use3 10
[Hub_1-ipsec-policy-templet-use3-10] ike-peer 3
[Hub_1-ipsec-policy-templet-use3-10] proposal 3
[Hub_1-ipsec-policy-templet-use3-10] security acl 3003
[Hub_1-ipsec-policy-templet-use3-10] quit
[Hub_1] ipsec policy policy3 10 isakmp template use3
[Hub_1] interface GigabitEthernet 1/0/0
[Hub_1-GigabitEthernet1/0/0] ipsec policy policy1
[Hub_1-GigabitEthernet1/0/0] quit
[Hub_1] interface gigabitethernet 2/0/0
[Hub_1-GigabitEthernet2/0/0] ipsec policy policy3
[Hub_1-GigabitEthernet2/0/0] quit
```

Configure Hub_2.

```
[Hub_2] acl number 3002
[Hub_2-acl-adv-3002] rule 5 permit ip source 202.10.1.10 0
[Hub_2-acl-adv-3002] quit
[Hub_2] acl number 3004
[Hub_2-acl-adv-3004] rule 5 permit ip source 202.10.1.14 0
[Hub_2-acl-adv-3004] quit
```

```
[Hub_2] ipsec proposal 2
[Hub_2-ipsec-proposal-2] quit
[Hub_2] ipsec proposal 4
[Hub_2-ipsec-proposal-4] quit
[Hub_2] ike peer 2 v1
[Hub_2-ike-peer-2] pre-shared-key cipher Huawei@1234
[Hub_2-ike-peer-2] quit
[Hub_2] ike peer 4 v1
[Hub_2-ike-peer-4] pre-shared-key cipher Huawei@1234
[Hub_2-ike-peer-4] quit
[Hub_2] ipsec policy-template use2 10
[Hub_2-ipsec-policy-templet-use2-10] ike-peer 2
[Hub_2-ipsec-policy-templet-use2-10] proposal 2
[Hub_2-ipsec-policy-templet-use2-10] security acl 3002
[Hub_2-ipsec-policy-templet-use2-10] quit
[Hub_2] ipsec policy policy2 10 isakmp template use2
[Hub_2] ipsec policy-template use4 10
[Hub_2-ipsec-policy-templet-use4-10] ike-peer 4
[Hub_2-ipsec-policy-templet-use4-10] proposal 4
[Hub_2-ipsec-policy-templet-use4-10] security acl 3004
[Hub_2-ipsec-policy-templet-use4-10] quit
[Hub_2] ipsec policy policy4 10 isakmp template use4
[Hub_2] interface GigabitEthernet 1/0/0
[Hub_2-GigabitEthernet1/0/0] ipsec policy policy4
[Hub_2-GigabitEthernet1/0/0] quit
[Hub_2] interface gigabitethernet 2/0/0
[Hub_2-GigabitEthernet2/0/0] ipsec policy policy2
[Hub_2-GigabitEthernet2/0/0] quit
```

Configure the Spoke.

```
[Spoke] acl number 3001
[Spoke-acl-adv-3001] rule 5 permit ip destination 202.10.1.2 0
[Spoke-acl-adv-3001] quit
[Spoke] acl number 3002
[Spoke-acl-adv-3002] rule 5 permit ip destination 202.10.1.10 0
[Spoke-acl-adv-3002] quit
[Spoke] acl number 3003
[Spoke-acl-adv-3003] rule 5 permit ip destination 202.10.1.6 0
[Spoke-acl-adv-3003] quit
[Spoke] acl number 3004
[Spoke-acl-adv-3004] rule 5 permit ip destination 202.10.1.14 0
[Spoke-acl-adv-3004] quit
[Spoke] ipsec proposal 1
[Spoke-ipsec-proposal-1] quit
[Spoke] ipsec proposal 2
[Spoke-ipsec-proposal-2] quit
[Spoke] ipsec proposal 3
[Spoke-ipsec-proposal-3] quit
[Spoke] ipsec proposal 4
[Spoke-ipsec-proposal-4] quit
[Spoke] ike peer 1 v1
[Spoke-ike-peer-1] pre-shared-key cipher Huawei@1234
[Spoke-ike-peer-1] remote-address 202.10.1.2
[Spoke-ike-peer-1] quit
[Spoke] ike peer 2 v1
[Spoke-ike-peer-2] pre-shared-key cipher Huawei@1234
[Spoke-ike-peer-2] remote-address 202.10.1.10
[Spoke-ike-peer-2] quit
[Spoke] ike peer 3 v1
[Spoke-ike-peer-3] pre-shared-key cipher Huawei@1234
[Spoke-ike-peer-3] remote-address 202.10.1.6
[Spoke-ike-peer-3] quit
[Spoke] ike peer 4 v1
[Spoke-ike-peer-4] pre-shared-key cipher Huawei@1234
[Spoke-ike-peer-4] remote-address 202.10.1.14
[Spoke-ike-peer-4] quit
[Spoke] ipsec policy policy1 10 isakmp
[Spoke-ipsec-policy-isakmp-policy1-10] ike-peer 1
[Spoke-ipsec-policy-isakmp-policy1-10] proposal 1
```

```
[Spoke-ipsec-policy-isakmp-policy1-10] security acl 3001
[Spoke-ipsec-policy-isakmp-policy1-10] quit
[Spoke] ipsec policy policy1 20 isakmp
[Spoke-ipsec-policy-isakmp-policy1-20] ike-peer 2
[Spoke-ipsec-policy-isakmp-policy1-20] proposal 2
[Spoke-ipsec-policy-isakmp-policy1-20] security acl 3002
[Spoke-ipsec-policy-isakmp-policy1-20] quit
[Spoke] ipsec policy policy1 30 isakmp
[Spoke-ipsec-policy-isakmp-policy1-30] ike-peer 3
[Spoke-ipsec-policy-isakmp-policy1-30] proposal 3
[Spoke-ipsec-policy-isakmp-policy1-30] security acl 3003
[Spoke-ipsec-policy-isakmp-policy1-30] quit
[Spoke] ipsec policy policy1 40 isakmp
[Spoke-ipsec-policy-isakmp-policy1-40] ike-peer 4
[Spoke-ipsec-policy-isakmp-policy1-40] proposal 4
[Spoke-ipsec-policy-isakmp-policy1-40] security acl 3004
[Spoke-ipsec-policy-isakmp-policy1-40] quit
[Spoke] interface cellular 0/0/0
[Spoke-Cellular0/0/0] ipsec policy policy1
[Spoke-Cellular0/0/0] quit
```

Step 8 Verify the configuration.

After the configuration is complete, run the **display nhrp peer all** command on Hub_1 and Hub_2 to check the registration information of the Spoke. The display on Hub_1 is used as an example:

```
[Hub_1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.1.2     32    202.10.10.10   172.16.1.2    registered up|unique
-----
Tunnel interface: Tunnel0/0/1
Created time    : 00:02:59
Expire time     : 01:57:01
```

The branch can ping the headquarters successfully and data transmitted between them is encrypted.

Run the **display ipsec sa** command on the Spoke. You can see that the Spoke has set up an IPSec tunnel with Hub_1.

Shut down GE1/0/0 on Hub_1 and GE2/0/0 on Hub_2 to simulate a fault on LTE network 1.

```
[Hub_1] interface GigabitEthernet 1/0/0
[Hub_1-GigabitEthernet1/0/0] shutdown
[Hub_1-GigabitEthernet1/0/0] quit
[Hub_2] interface gigabitethernet 2/0/0
[Hub_2-GigabitEthernet2/0/0] shutdown
[Hub_2-GigabitEthernet2/0/0] quit
```

Run the **display nhrp peer all** command on Hub_1 and Hub_2. You can see that the Spoke registers to the headquarters through LTE network 2. The display on Hub_1 is used as an example:

```
[Hub_1] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.16.3.2     32    202.11.11.11   172.16.3.2    registered up|unique
-----
Tunnel interface: Tunnel0/0/3
Created time    : 00:02:59
Expire time     : 01:57:01
```

The branch can ping the headquarters successfully and data transmitted between them is encrypted.

```
[Spoke] ping -a 192.168.3.1 192.168.1.1
PING 192.168.1.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.1.1: bytes=56 Sequence=1 ttl=254 time=3 ms
  Reply from 192.168.1.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 192.168.1.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 192.168.1.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 192.168.1.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 192.168.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/2/3 ms
```

----End

Configuration Files

- Hub_1 configuration file

```
#
sysname Hub_1
#
acl number
3001
 rule 5 permit ip source 202.10.1.2
0
acl number
3003
 rule 5 permit ip source 202.10.1.6 0
#
ipsec proposal
1
ipsec proposal
3
#
ike peer 1
v1
 pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl0OU@DraF58J4=;%^%#
ike peer 3
v1
 pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl0OU@DraF58J4=;%^%#
#
ipsec policy-template use1
10
 security acl
3001
 ike-peer
1
 proposal
1
ipsec policy-template use3
10
 security acl
3003
 ike-peer
3
 proposal
3
#
ipsec policy policy1 10 isakmp template
use1
ipsec policy policy3 10 isakmp template
use3
#
interface GigabitEthernet1/0/0
```

```
ip address 202.10.1.2
255.255.255.252
ipsec policy policy1
#
interface GigabitEthernet2/0/0
ip address 202.10.1.6
255.255.255.252
ipsec policy policy3
#
interface GigabitEthernet3/0/0
ip address 192.168.1.1 255.255.255.0
#
interface
Tunnel0/0/1
ip address 172.16.1.1
255.255.255.0
tunnel-protocol gre
p2mp
source
GigabitEthernet1/0/0

gre key cipher %^%#3isY%^1X6F&N'Us)3x+\m@F0A2(SQ&=2|;K8abO%^%#
nhrp authentication cipher %^%#1"<9Jp7D_'(SE-N.oVH5B5wZ=WO^KC1OL|-UOIQ$%^%#
nhrp registration no-
unique
nhrp entry multicast
dynamic
nhrp entry holdtime seconds
60
#

interface
Tunnel0/0/3
ip address 172.16.3.1
255.255.255.0
tunnel-protocol gre
p2mp
source
GigabitEthernet2/0/0

gre key cipher %^%#=SXc*PbQgMQ|6<1H|8_W!PU!XFrjE7}LVC(ycs38%^%#
nhrp authentication cipher %^%#EjU:.Y}].8YZ8JK07')Qw\rTXJ|;LFAFFIH:C]W=%^%#
nhrp registration no-
unique
nhrp entry multicast
dynamic
nhrp entry holdtime seconds
60
#

rip
1
undo
summary
version
2
network
172.16.0.0
network
192.168.1.0
#
ip route-static 0.0.0.0 0.0.0.0
202.10.1.1
ip route-static 0.0.0.0 0.0.0.0 202.10.1.5
#
return
```

- Hub_2 configuration file

```
#
sysname Hub_2
```

```
#
acl number
3002
 rule 5 permit ip source 202.10.1.10
0
acl number
3004
 rule 5 permit ip source 202.10.1.14 0
#
ipsec proposal
2
ipsec proposal
4
#
ike peer 2
v1
pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%^%#
ike peer 4
v1
pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%^%#
#
ipsec policy-template use2
10
 security acl
3002
 ike-peer
2
 proposal
2
ipsec policy-template use4
10
 security acl
3004
 ike-peer
4
 proposal
4
#
ipsec policy policy2 10 isakmp template
use2
ipsec policy policy4 10 isakmp template
use4
#
interface GigabitEthernet1/0/0
 ip address 202.10.1.14
255.255.255.252
 ipsec policy policy4
#
interface GigabitEthernet2/0/0
 ip address 202.10.1.10
255.255.255.252
 ipsec policy policy2
#
interface GigabitEthernet3/0/0
 ip address 192.168.1.2 255.255.255.0
#
interface
Tunnel0/0/2
 ip address 172.16.2.1
255.255.255.0
 tunnel-protocol gre
p2mp
 source GigabitEthernet2/0/0
 gre key cipher %^%#9gxVF{"ZQT;-D<%Gm2I10Qd5 (uV!2> (3#q2%V3R#%^
%#
 nhrp authentication cipher %^%#g9*MEwPqQOCw:@Jt2WS9:,LNDn[|8If>@9&!2zQQ%^
```

```
%#
 nhrp registration no-
unique
 nhrp entry multicast
dynamic
 nhrp entry holdtime seconds
60
#

interface
 Tunnel0/0/4
 ip address 172.16.4.1
255.255.255.0
 tunnel-protocol gre
p2mp
 source GigabitEthernet1/0/0
 gre key cipher %^%#Y4YfQCCO%Of+{(KpezQ9b!nWTt:6I9wR)o#:Kr,!%^%#
 nhrp authentication cipher %^%#BChE#]PR%Z' [<-&:Eq/GM@z=L%^%#BChE#]PR%Z' [<-
&:Eq/GM@z=L
 nhrp registration no-
unique
 nhrp entry multicast
dynamic
 nhrp entry holdtime seconds
60
#
rip
1
 undo
summary
version
2
 network
172.16.0.0
 network
192.168.1.0
#
ip route-static 0.0.0.0 0.0.0.0
202.10.1.9
ip route-static 0.0.0.0 0.0.0.0 202.10.1.13
#
return
```

● Spoke configuration file

```
#
sysname Spoke
#
acl number
3001
 rule 5 permit ip destination 202.10.1.2
0
acl number
3002
 rule 5 permit ip destination 202.10.1.10
0
acl number
3003
 rule 5 permit ip destination 202.10.1.6
0
acl number
3004
 rule 5 permit ip destination 202.10.1.14
0
#
ipsec proposal
1
ipsec proposal
2
ipsec proposal
3
```

```
ipsec proposal
4
#
ike peer 1 v1
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%%#
remote-address 202.10.1.2
ike peer 2 v1
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%%#
remote-address 202.10.1.10
ike peer 3
v1
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%%#
remote-address 202.10.1.6
ike peer 4
v1
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iVl00U@DraF58J4=;%%#
remote-address 202.10.1.14
#
ipsec policy policy1 10
isakmp
security acl
3001
ike-peer
1
proposal
1
ipsec policy policy1 20
isakmp
security acl
3002
ike-peer
2
proposal
2
ipsec policy policy1 30
isakmp
security acl
3003
ike-peer
3
proposal
3
ipsec policy policy1 40
isakmp
security acl
3004
ike-peer
4
proposal
4
#
interface GigabitEthernet1/0/0
ip address 192.168.3.1 255.255.255.0
#
interface
Cellular0/0/0
dialer enable-
circular
dialer-group
1
dialer timer autodial
15
dialer timer probe-interval
15
dialer number *99#
autodial
apn-profile ltenet priority 200 track nqa admin Tunnel0/0/1 admin Tunnel0/0/2
apn-profile ltewap priority 150 track nqa admin Tunnel0/0/3 admin Tunnel0/0/4
```

```
ip address negotiate
ipsec policy policy1
#
interface
Tunnel0/0/1
 ip address 172.16.1.2
255.255.255.0
 rip metricin
1
 tunnel-protocol gre
p2mp
 source
Cellular0/0/0
 gre key cipher %^%#3isY%^1X6F&N'Us)3x+\m@F0A2(SQ&=2|;K8abO%^%#
 nhrp authentication cipher %^%#1"<9Jp7D_'(SE-N.oVH5B5wZ=WO^KClOL|-UOIQ$%^%#
 nhrp registration interval
20
 nhrp entry 172.16.1.1 202.10.1.2 register track apn
ltenet
#

interface
Tunnel0/0/2
 ip address 172.16.2.2
255.255.255.0
 rip metricin
7
 rip metricout
7
 tunnel-protocol gre
p2mp
 source
Cellular0/0/0
 gre key cipher %^%#9gxVF("ZQT;-D<%Gm2I1OQd5(uV!2>(3#g2%V3R#%^%#
 nhrp authentication cipher %^%#g9*MEwPqQOCw:@Jt2WS9:,LNDn[|8If>@9&!2zQQ$%^%#
 nhrp registration interval
20
 nhrp entry 172.16.2.1 202.10.1.10 register track apn
ltenet
#

interface
Tunnel0/0/3
 ip address 172.16.3.2
255.255.255.0
 rip metricin
4
 rip metricout
4
 tunnel-protocol gre
p2mp
 source
Cellular0/0/0
 gre key cipher %^%#=SXc*EbQgMQ|6<1H|8_W!PU!XFrjE7}LVC(ycs38%^%#
 nhrp authentication cipher %^%#EjU:.Y}] .8YZ8JK07')Qw\rTXJ|;LFAFfIH:C]W=%^
%#
 nhrp registration interval
20
 nhrp entry 172.16.3.1 202.10.1.6 register track apn
ltewap
#

interface
Tunnel0/0/4
 ip address 172.16.4.2
255.255.255.0
 rip metricin
10
 rip metricout
```

```
10
 tunnel-protocol gre
 p2mp
 source
 Cellular0/0/0
 gre key cipher %^%#Y4YfQCCO%Of+{(KpezQ9b!nWTt:6I9wR) o#:Kr,!%^%#
 nhrp authentication cipher %^%#BChE#]PR%Z' [<-&:Eq/GM@z=L%^%#BChE#]PR%Z' [<-
 &:Eq/GM@z=L
 nhrp registration interval
20
 nhrp entry 172.16.4.1 202.10.1.14 register track apn
 ltewap
#

dialer-
rule
 dialer-rule 1 ip
 permit
#

apn profile
 ltenet
 apn LTENET1
 sim-id 1
apn profile
 ltewap
 apn LTENET2
 sim-id 2
#

rip
1
 version
2
 network
172.16.0.0
 network
192.168.3.0
#
ip route-static 0.0.0.0 0.0.0.0 Cellular0/0/0
#
nqa test-instance admin
 Tunnel0/0/1
 test-type
 icmp
 destination-address ipv4
172.16.1.1
 source-address ipv4
172.16.1.2
 frequency
15
 source-interface
 Tunnel0/0/1
 start
now
nqa test-instance admin
 Tunnel0/0/2
 test-type
 icmp
 destination-address ipv4
172.16.2.1
 source-address ipv4
172.16.2.2
 frequency
15
 source-interface
 Tunnel0/0/2
 start
now
```

```
nqa test-instance admin
Tunnel0/0/3
 test-type
 icmp
 destination-address ipv4
 172.16.3.1
 source-address ipv4
 172.16.3.2
 frequency
 15
 source-interface
 Tunnel0/0/3
 start
 now
nqa test-instance admin
Tunnel0/0/4
 test-type
 icmp
 destination-address ipv4
 172.16.4.1
 source-address ipv4
 172.16.4.2
 frequency
 15
 source-interface
 Tunnel0/0/4
 start
 now
#
return
```

5.9 Troubleshooting DSVPN

5.9.1 Spoke Fails to Register with a Hub

Fault Description

After the **display nhrp peer** command is executed on the Hub, no NHRP mapping entry that records the mapping between the tunnel address of the Spoke and the public network address is displayed.

Procedure

Step 1 Check that the Spoke has reachable routes to the remote Spoke and the Hub.

Run the **display ip routing-table** command to check whether there are routes to the remote end.

- If there is no reachable route between the Spoke and Hub, check the configurations of routes on the Spoke and Hub. For the configurations of routes, see the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IP Routing*.
- If there are reachable routes between the Spoke and Hub, go to step 2.

Step 2 Check that configurations of the Spoke and Hub are correct.

Run the **display nhrp peer** command on the Spoke and Hub to check NHRP mapping entries.

If the Hub does not have dynamic NHRP mapping entries of the Spoke, run the **display this** command on mGRE tunnel interfaces of the Spoke and Hub to check whether the

configurations on both ends are consistent. The following table lists the fields in the command output that you need to check the follow-up operations.

Item	Check Standard and Operation
nhrp authentication	Check whether NHRP authentication string configurations of the Spoke and Hub are the same. If they are different, run the nhrp authentication command to modify the configurations.
nhrp entry	Check whether the static NHRP mapping entries on the Spoke contain the interface information of the Hub. If not, run the nhrp entry command to modify the configurations.

----End

5.9.2 Subnets Between Spokes Cannot Communicate Directly in Non-Shortcut Mode

Fault Description

After the non-shortcut mode is configured, subnets between Spokes cannot communicate.

Procedure

- Step 1** Check whether subnet routes are available between Spokes, and between Spokes and the Hub, and whether the next-hop addresses of subnet routes are the tunnel addresses of the peer devices.

Run the **display ip routing-table** command on the local Spoke to check whether routes to the remote Spoke exist in the local IP routing table. Run the **display ip routing-table** command on the Hub to check whether subnet routes to Spokes exist in the local IP routing table.

- If no subnet route is available between Spokes, and between Spokes and the Hub, configure subnet routes. For the configurations of routes, see the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IP Routing*.
- If subnet routes are available between Spokes, and between Spokes and the Hub, but the next hop to the destination subnet is not the tunnel address of the remote device, configure routing information to set the next hop to the destination subnet to the tunnel address of remote device. For details, see [Configuring Routes](#).
- If subnet routes are available between Spokes, and between Spokes and the Hub, and the next hop to the destination subnet is the tunnel address of remote device, go to step 2.

- Step 2** Check whether NHRP mapping entries of a local Spoke have been generated on the Hub and the remote Spoke.

Run the **display nhrp peer** command on the Hub and Spoke to check NHRP mapping entries.

If no NHRP mapping entry of the Spoke is generated on the Hub, rectify the fault according to [5.9.1 Spoke Fails to Register with a Hub](#).

----End

5.9.3 Subnets Between Spokes Cannot Communicate Directly in Shortcut Mode

Fault Description

After the shortcut mode is configured, subnets between Spokes cannot communicate.

Procedure

Step 1 Check that subnet routes are available between Spokes, and between Spokes and the Hub.

Run the **display ip routing-table** command on the local Spoke to check whether routes to the remote Spoke exist in the local IP routing table. Run the **display ip routing-table** command on the Hub to check whether subnet routes to Spokes exist in the local IP routing table.

- If no subnet route is available between Spokes, and between Spokes and the Hub, configure subnet routes. For the configurations of routes, see the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IP Routing*.
- If subnet routes are available between Spokes, and between Spokes and the Hub, go to step 2.

Step 2 Check whether the next hop to the destination subnet is the tunnel address of the Hub.

Run the **display ip routing-table** command on the local Spoke to check whether routes to the remote Spoke exist.

- If the next hop to the destination subnet is not the tunnel address of the Hub, configure routing information to set the next hop to the destination subnet to the tunnel address of the Hub. For details, see [Configuring Routes](#).
- If the next hop to the destination subnet is the tunnel address of the Hub, go to step 3.

Step 3 Check whether NHRP mapping entries of a local Spoke have been generated on the Hub and the remote Spoke.

Run the **display nhrp peer** command on the Hub and Spoke to check NHRP mapping entries.

If no NHRP mapping entry of the Spoke is generated on the Hub, rectify the fault according to [5.9.1 Spoke Fails to Register with a Hub](#).

----End

5.9.4 Backup Hub Only Forwards Data After the Master Hub Fails

Fault Description

In dual-Hub DSVPN scenario, the backup Hub only forwards data after the master Hub fails. No tunnel can be established between the Spokes.

Procedure

Step 1 Check whether the public addresses configured on the master and backup Hubs are on the same network segment.

Run the **display this** command on the mGRE interfaces of the master and backup Hubs to check whether the IP addresses of the Hubs are on the same network segment.

- If so, change the IP address of one Hub to an IP address on a different network segment.
- If not, go to step 2.

Step 2 Check whether routes to the master Hub are available on the Spokes.

Run the **display ip routing-table** command on the Spokes to check whether routes to the Hub exist.

If the IP routing table contains routes to the master Hub, deletes the routes. For details, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IP Routing*.

----End

5.10 References for DSVPN

The following table lists the references for DSVPN.

Document	Description
RFC2332	Next Hop Resolution Protocol

6 IPsec Configuration

About This Chapter

- [6.1 Overview of IPsec](#)
- [6.2 Understanding IPsec](#)
- [6.3 Application Scenarios for IPsec](#)
- [6.4 Summary of IPsec Configuration Tasks](#)
- [6.5 Licensing Requirements and Limitations for IPsec](#)
- [6.6 Default Settings for IPsec](#)
- [6.7 Using an ACL to Establish an IPsec Tunnel](#)
- [6.8 Using a Virtual Tunnel Interface to Establish an IPsec Tunnel](#)
- [6.9 Establishing an IPsec Tunnel Using an Efficient VPN Policy](#)
- [6.10 Configuring IKE](#)
- [6.11 Maintaining IPsec](#)
- [6.12 Configuration Examples for IPsec](#)
- [6.13 Troubleshooting IPsec](#)
- [6.14 FAQ About IPsec](#)
This section describes the FAQ about IPsec.
- [6.15 References for IPsec](#)

6.1 Overview of IPsec

Definition

Internet Protocol Security (IPsec), defined by the Internet Engineering Task Force (IETF), is a series of open network security protocols and services provided on an IP network. **Figure 6-1** shows the IPsec protocol framework.

Figure 6-1 IPsec protocol framework

Security protocols	ESP			AH		
Encryption	DES	3DES	AES			
Authentication	MD5	SHA1	SHA2	MD5	SHA1	SHA2
Key exchange	IKE (ISAKMP, DH)					

IPsec protects IP packets using two **security protocols**: Authentication Header (AH) and Encapsulating Security Payload (ESP).

- AH provides data origin authentication, data integrity check, and anti-replay, but does not provide encryption.
- ESP provides encryption, data origin authentication, data integrity check, and anti-replay.

Security functions provided by the AH and ESP protocols depend on **authentication** and **encryption** algorithms.

- Both AH and ESP can provide data origin authentication and data integrity check using authentication algorithms Message Digest 5 (MD5), Secure Hash Algorithm 1 (SHA1), Secure Hash Algorithm 2 (SHA2)-256, SHA2-384, and SHA2-512.
- ESP can also encrypt IP packets using symmetric encryption algorithms, including Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES).

NOTE

- The MD5 and SHA1 authentication algorithms have security risks. The SHA2 algorithm is recommended.
- The DES and 3DES encryption algorithms have security risks. The AES algorithm is recommended.

The keys used in IPsec encryption and authentication algorithms can be manually configured or dynamically negotiated through the Internet Key Exchange (IKE) protocol. IKE works in the Internet Security Association and Key Management Protocol (ISAKMP) framework. It uses the Diffie-Hellman (DH) algorithm to securely deliver keys and authenticate identities over an insecure network, ensuring data transmission security. IKE improves key security and simplifies IPsec management.

Purpose

On the Internet, most data is transmitted in plain text, causing security risks. For example, bank accounts and passwords face risks of eavesdropping or tampering, user identities may be counterfeited, or bank networks may be attacked. IPsec can protect IP packets transmitted over an insecure network to reduce the risk of information leaks.

Benefits

Taking advantage of encryption and authentication, IPsec ensures secure service data transmission over the Internet in terms of:

- Data origin authentication: The receiver checks validity of the sender.
- Data encryption: The sender encrypts data packets and transmits them in cipher text on the Internet. The receiver decrypts or directly forwards the received data packets.
- Data integrity check: The receiver validates received data to check whether the data has been tampered with.
- Anti-replay: The receiver rejects old or duplicate packets to prevent attacks that malicious users initiate by re-sending obtained packets.

6.2 Understanding IPsec

6.2.1 Basic Concepts of IPsec

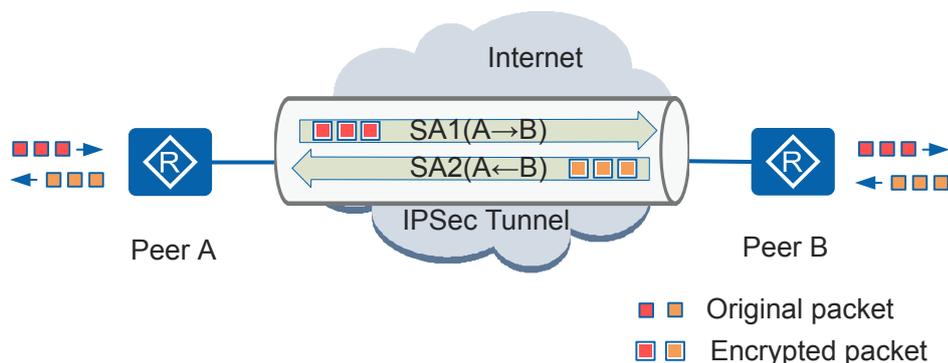
A **security association (SA)** needs to be established between IPsec peers (two IPsec endpoints) before IPsec can implement secure data transmission. An SA defines a set of parameters for data transmission between two IPsec peers, including the **security protocol**, characteristics of data flows to be protected, data **encapsulation mode**, **encryption** algorithm, **authentication** algorithm, **Key Exchange, IKE**, and SA lifetime.

6.2.1.1 Security Association

An SA is identified by three parameters: security parameter index (SPI), destination IP address, and security protocol ID (AH or ESP). The SPI is a 32-bit value generated for uniquely identifying an SA, and is transmitted in an AH or ESP header. The SPI needs to be specified when an SA is manually configured. When an SA is generated during IKE negotiation, an SPI is generated randomly.

Because SAs are unidirectional, at least two SAs are required to protect incoming and outgoing data flows. In **Figure 6-2**, two SAs need to be established if an IPsec tunnel needs to be established between IPsec peers A and B. SA1 defines the protection mode for data sent from Peer A to Peer B, and SA2 defines the protection mode for data sent from Peer B to Peer A.

Figure 6-2 IPsec SAs



How many SAs are required also depends on the security protocol used. If you use either AH or ESP to protect traffic between two peers, two SAs are required to protect incoming and outgoing flows. If you use both AH and ESP to protect traffic between two peers, four SAs are required, two for each protocol.

An IPsec SA is established in manual or IKE auto-negotiation mode. The two modes differ in the following:

- Key generation mode
In manual mode, all the parameters used to establish an SA, including the encryption key and authentication key, need to be manually configured and updated, leading to high key management costs on large and medium-sized networks. In IKE auto-negotiation mode, the encryption key and authentication key are generated using the DH algorithm and can be dynamically updated, reducing key management costs and improving security.
- SA lifetime
An SA established manually exists permanently. How long an SA established in IKE auto-negotiation mode can exist depend on the lifetime parameters configured on two peers.

Based on the differences, the manual mode applies to small networks with a small number of IPsec peers, where as IKE auto-negotiation mode is recommended on large and medium-sized networks.

6.2.1.2 Security Protocol

IPsec uses two security protocols, Authentication Header (AH) and Encapsulating Security Payload (ESP), to transmit and encapsulate data. The security protocols provide security services such as authentication and encryption.

AH

AH is an IP-based transport-layer protocol with protocol number 51. According to the AH protocol, an AH header is appended to the standard IP header in each packet, as shown in [6.2.1.3 Encapsulation Mode](#). The sender performs hash calculation on packets and the authentication key. When the packets carrying the calculation result arrive at the receiver, the receiver also performs hash calculation and compares the calculation result with the received calculation result. Any changes to the data during transmission will make the calculation

result invalid. AH provides data origin authentication and data integrity check in this way. An integrity check is performed on an entire IP packet.

Figure 6-3 shows the AH header format.

Figure 6-3 AH header format

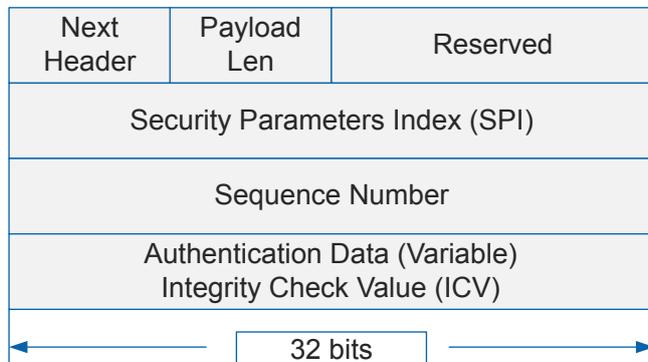


Table 6-1 describes the fields in an AH header.

Table 6-1 AH header fields

Field	Length	Description
Next Header	8 bits	This field identifies the type of the payload following the AH header. In transport mode , the Next Header field is the number of the protected upper-layer protocol (TCP or UDP) or ESP. In tunnel mode , the Next Header field is the number of the IP or ESP protocol. NOTE When both AH and ESP are used, the next header following an AH header is an ESP header.
Payload Length	8 bits	The value of this field is the AH packet length in 32-bit words minus 2. The default value is 4.
Reserved	16 bits	This field is reserved and defaults to 0.
SPI	32 bits	This field uniquely identifies an IPsec security association (SA).
Sequence Number	32 bits	This field is a counter that monotonically increases from 1. It uniquely identifies a packet to prevent replay attacks.

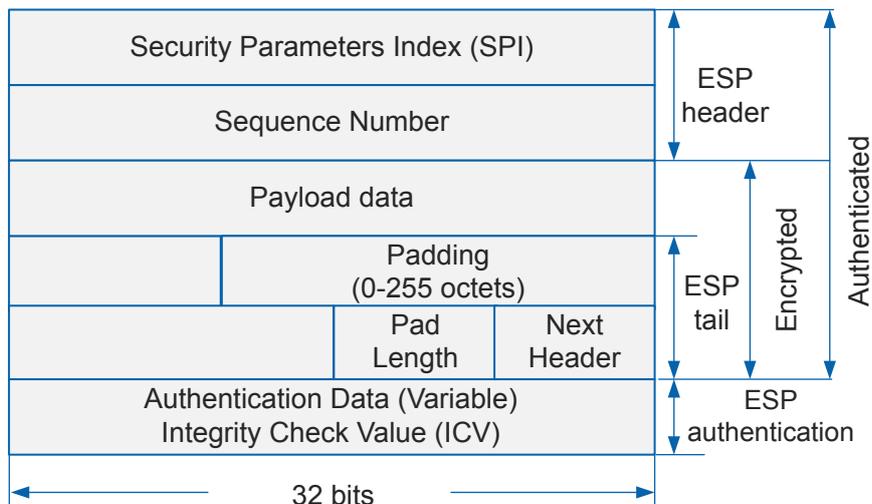
Field	Length	Description
Authentication Data	Integral multiple of 32 bits. It is 96 bits in common cases.	This field contains the Integrity Check Value (ICV) and is used by the receiver for data integrity check. Authentication algorithms include MD5, SHA1, and SHA2.

ESP

ESP is an IP-based transport-layer protocol, with the protocol number 50. According to the ESP protocol, an ESP header is appended to the standard IP header in each packet, and an ESP tail and ESP Auth data are appended to each packet, as shown in [6.2.1.3 Encapsulation Mode](#). Unlike AH, ESP encrypts the valid payload and then encapsulates it to a packet to ensure data confidentiality. However, ESP does not protect an IP header.

[Figure 6-4](#) shows the ESP header format.

Figure 6-4 ESP header format



[Table 6-2](#) describes the fields in an ESP header.

Table 6-2 ESP header fields

Field	Length	Description
SPI	32 bits	This field uniquely identifies an IPsec SA.

Field	Length	Description
Sequence Number	32 bits	This field is a counter that monotonically increases from 1. It uniquely identifies a packet to prevent replay attacks.
Payload Data	-	This field contains variable-length data identified by the Next Header field.
Padding	-	This field extends the size of an ESP header. The Padding field length depends on the payload data length and algorithm. If the plaintext length of the packet to be encrypted does not comply with the encryption algorithm, the Padding field is used.
Pad Length	8 bits	This field specifies the length of the Padding field. The value 0 indicates no padding.
Next Header	8 bits	This field identifies the type of the payload following the ESP header. In transport mode, the Next Header field is the number of the protected upper-layer protocol (TCP or UDP). In tunnel mode, the Next Header field is the IP protocol number.
Authentication Data	Integral multiple of 32 bits. It is 96 bits in common cases.	This field contains the Integrity Check Value (ICV) and is used by the receiver for data integrity check. Authentication algorithms which are the same as those of AH. The authentication function of ESP is optional. If data check is enabled, an ICV value is appended to encrypted data.

Comparisons Between AH and ESP

Table 6-3 compares AH and ESP.

Table 6-3 Comparisons between AH and ESP

Security Feature	AH	ESP
Protocol number	51	50
Data integrity check	Supported (checking the entire IP packet)	Supported (not checking the IP header)
Data origin authentication	Supported	Supported
Data encryption	Not supported	Supported
Anti-replay	Supported	Supported
IPsec NAT-T (NAT traversal)	Not supported	Supported

According to [Table 6-3](#), AH does not provide data encryption and ESP does not check an IP header. Therefore, use both AH and ESP when high security is required.

6.2.1.3 Encapsulation Mode

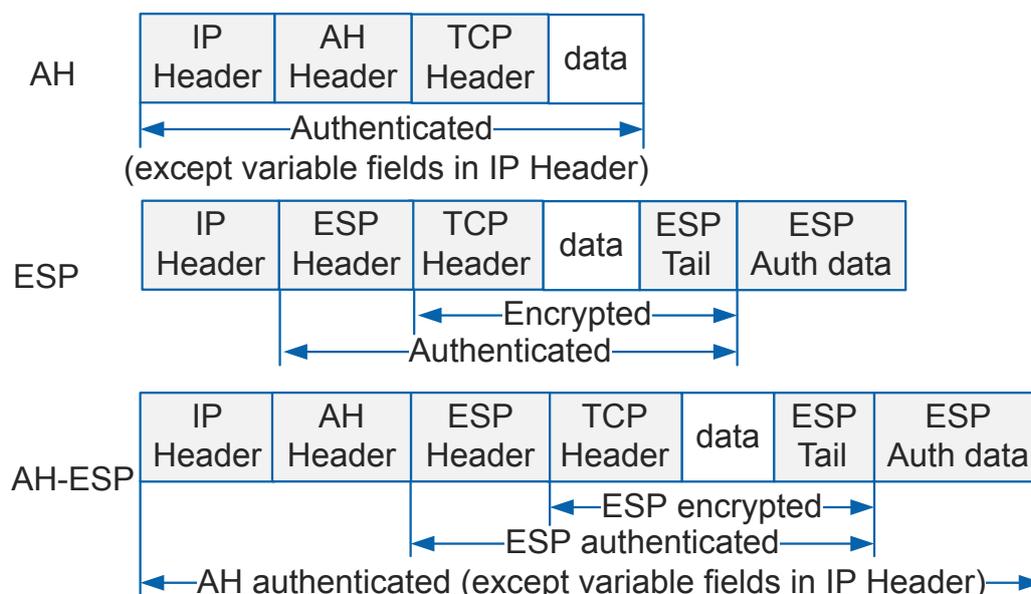
Encapsulation is a process of adding AH or ESP fields to original IP packets for packet authentication and encryption. This process is implemented in transport or tunnel mode.

Transport Mode

In transport mode, an AH or ESP header is added between an IP header and a transport-layer protocol header to protect the TCP, UDP, or ICMP payload. The transport mode does not change the IP header, so the source and destination addresses of an IPsec tunnel must be the same as those in the IP header. This encapsulation mode applies only to communication between two hosts or between a host and a VPN gateway.

[Figure 6-5](#) shows an example of TCP packet encapsulation in transport mode.

Figure 6-5 Packet encapsulation in transport mode



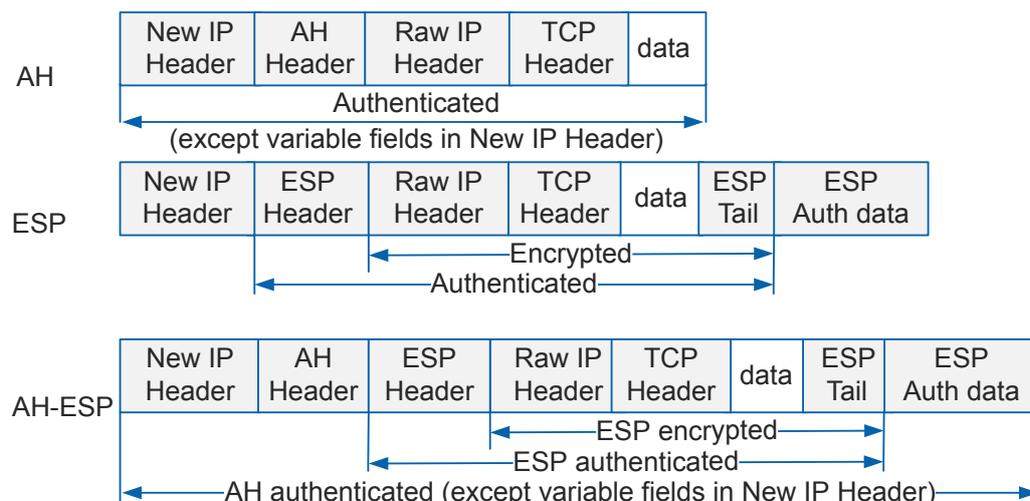
In transport mode, AH checks the integrity of the entire IP packet. ESP checks the integrity of the ESP header, transport layer protocol header, data, and ESP tail, excluding the IP header. Therefore, ESP cannot protect the IP header. ESP encrypts the transport-layer protocol header, data, and ESP tail.

Tunnel Mode

In tunnel mode, an AH or ESP header is added outside the raw IP header, and a new IP header added outside the AH or ESP header to protect the IP header and payload. The tunnel mode applies to communication between two VPN gateways or between a host and a VPN gateway.

[Figure 6-6](#) shows an example of TCP packet encapsulation in tunnel mode.

Figure 6-6 Packet encapsulation in tunnel mode



In tunnel mode, AH checks the integrity of the entire IP packet including the new IP header. ESP checks the integrity of the ESP header, raw IP header, transport layer protocol header, data, and ESP tail, excluding the new IP header. Therefore, ESP cannot protect the new IP header. ESP encrypts the raw IP header, transport-layer protocol header, data, and ESP tail.

Comparisons Between the Transport Mode and Tunnel Mode

The two encapsulation modes differ in the following:

- The tunnel mode is more secure because original IP packets can be completely authenticated and encrypted in tunnel mode. This mode hides the IP address, protocol type, and port number in an original IP packet.
- The tunnel mode generates an additional IP header, occupying more bandwidth than the transport mode.

When both AH and ESP are used to protect traffic, they must use the same encapsulation mode.

6.2.1.4 Encryption

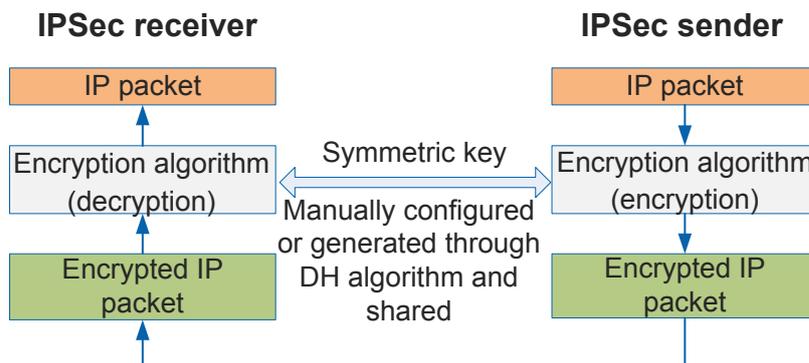
Encryption is a process of converting plaintext data into ciphertext data using an algorithm. The receiver can decrypt ciphertext data only when it has the correct key. The encryption mechanism ensures data confidentiality and prevents data from being eavesdropped during transmission. IPsec involves data encryption and protocol message encryption.

Data Encryption

IPsec uses symmetric encryption algorithms to encrypt and decrypt data. Symmetric encryption algorithms require that the sender and receiver use the same key to encrypt and decrypt data.

Figure 6-7 shows the process of using a symmetric encryption algorithm to encrypt and decrypt data.

Figure 6-7 Data encryption and decryption



The symmetric key can be manually configured or generated through **IKE** auto-negotiation.

Common symmetric encryption algorithms include:

- Data Encryption Standard (DES)
DES was developed by the National Institute of Standards and Technology (NIST). It uses a 56-bit key to encrypt a 64-bit plaintext block.
- Triple Data Encryption Standard (3DES)
3DES is an enhancement to DES and uses three different 56-bit keys (168 bits in total) to encrypt a plaintext block.
Compared with DES, 3DES is slower but more secure.
- Advanced Encryption Standard (AES)
AES is designed to replace 3DES and is faster and more secure than 3DES. AES supports three types of keys: AES-128, ES-192, and AES-256, which have key lengths of 128 bits, 192 bits, and 256 bits, respectively.
The encryption algorithm with a longer key is more secure but slower. In general, AES-128 can meet security requirements.

Protocol Message Encryption

Protocol message encryption occurs in IKE negotiation. Symmetric encryption algorithms, such as DES, 3DES, and AES, are used to encrypt protocol messages. The symmetric key used for protocol message encryption is generated through **IKE** auto-negotiation.

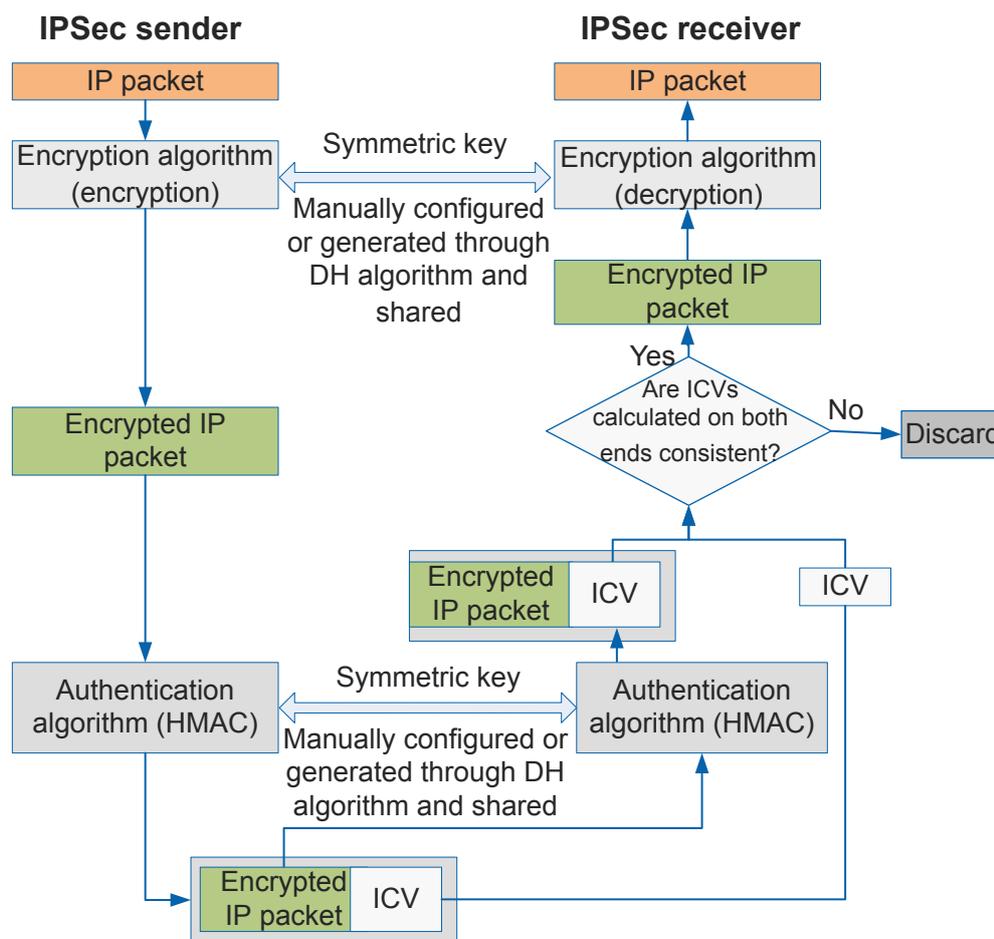
6.2.1.5 Authentication

Authentication is an operation that a receiver performs to verify the identity of the sender (source origin authentication) and whether data has been tampered with during transmission data integrity check. IPsec ensures data reliability using source origin authentication and data integrity check together.

Although encrypted data can only be decrypted using the original encryption key, the decrypted data cannot be proved to be the original data. Additionally, data encryption and decryption consume many CPU resources, and malicious users may send spoofing packets to consume CPU resources. Keyed-Hash Message Authentication Code (HMAC) compares digital signatures to check data integrity and authenticity. This process consumes only a few CPU resources and is very efficient. Therefore, IPsec uses HMAC for authentication.

Encryption and authentication are often used together on the IPsec sender. The sender encrypts IP packets, generates a digital signature during HMAC authentication, and then sends both the encrypted IP packets and digital signature to the receiver. The digital signature is included in the Integrity Check Value (ICV) field in an AH or ESP header. For details, see [6.2.1.2 Security Protocol](#). The receiver compares the received digital signature with the locally generated one to check data integrity and authenticity in the received IP packets. The receiver discards the packets that fail the authentication and decrypts those that pass the authentication. [Figure 6-8](#) shows the process of encryption and HMAC authentication.

Figure 6-8 Encryption and HMAC authentication



Like an **encryption** key, a symmetric authentication key can be manually configured or generated through **IKE** auto-negotiation.

Common authentication algorithms include:

- MD5
 - MD5 is defined in RFC 1321. It generates a 128-bit signature based on a message of any length.
 - MD5 is faster but less secure than Secure Hash Algorithm (SHA).
- SHA1

SHA was developed by the National Institute of Standards and Technology (NIST). SHA1 is a revision to SHA and was published in 1994. Defined in RFC 2404, SHA1 converts a message of a length less than 2^{64} bits into a 160-bit message digest.

SHA1 is slower but more secure than MD5. SHA1 generates a long signature to prevent key cracking, and discovers the shared key efficiently.

- SHA2

SHA2 is an enhancement to SHA1. It has a larger key length and is much more secure than SHA1. SHA2 includes SHA2-256, SHA2-384, and SHA2-512, with 256-bit, 384-bit, and 512-bit key lengths, respectively.

The authentication algorithm with a longer key is more secure but slower. In general, SHA2-256 can meet security requirements.

- AES-XCBC-MAC-96 is a type of AES-based message authentication algorithm and is described in RFC 3566.

The algorithms have their own strengths and weaknesses. MD5 is faster than SHA1, but less secure. SHA2 has a longer key than SHA1, which means that SHA2 is more difficult to defeat, and therefore more secure.

6.2.1.6 Key Exchange

How to securely share a key is an important issue during symmetric key encryption and authentication. Two ways are available to address this issue:

- Out-of-band key sharing

The encryption key and authentication key are manually configured on the sender and receiver. The two parties ensure key consistency in out-of-band mode (through phones or mails for example). This mode has poor scalability and multiplies the workload in configuring the key in point-to-multipoint networking. In addition, this mode is difficult to implement because the keys need to be changed periodically to improve network security.

- Using a secure key distribution protocol

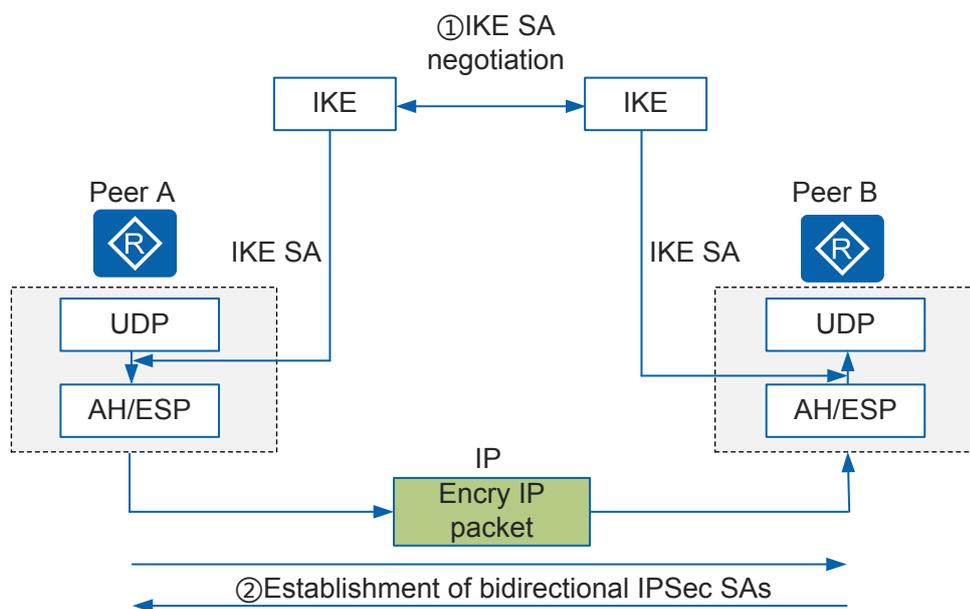
The key is generated through **IKE** auto-negotiation. The Internet Key Exchange (IKE) protocol uses DH (Diffie-Hellman) algorithm to implement secure key distribution over an insecure network. This mode is easy to configure and has high scalability, especially on a large dynamic network. Two communicating parties exchange key materials to calculate the shared key. Even through a third party obtains all the exchanged data used to calculate the shared key, it cannot calculate the shared key.

6.2.1.7 IKE

Internet Key Exchange (IKE) is a User Datagram Protocol (UDP)-based application-layer protocol built on the Internet Security Association and Key Management Protocol (ISAKMP) framework. It implements automatic key negotiation and IPsec **6.2.1.1 Security Association** setup, to simplify IPsec use and management, and facilitate IPsec configuration and maintenance.

Figure 6-9 shows the relationship between IKE and IPsec. The two peers establish an IKE SA for identity authentication and key information exchange. Protected by the IKE SA, the peers negotiate a pair of IPsec SAs using the Authentication Header (AH) or Encapsulating Security Payload (ESP) protocol and other parameters configured. Subsequently, data is encrypted and transmitted between the peers in an IPsec tunnel.

Figure 6-9 Relationship between IKE and IPsec



IKE Security Mechanisms

IKE defines a series of self-protection mechanisms that can securely authenticate identities, distribute keys, and establish IPsec SAs on an insecure network:

- Identity authentication

Two peers authenticate the identity (IP address or name) of each other, using Pre-shared key (PSK) authentication, RSA signature authentication.

- Pre-shared key authentication: An authentication key is used to generate a key. The two peers compute the hash value of packets using a shared key and check whether they obtain the same hash value. If they obtain the same hash value, the authentication succeeds. Otherwise, the authentication fails.
- RSA signature authentication: The two peers use a certificate issued by a certificate authority (CA) to verify validity of a digital certificate. Each peer has a public key (transmitted over the network) and a private key (possessed by itself). The sender computes a hash value for original packets, and then encrypts the hash value using its private key to generate a digital signature. The receiver decrypts the digital signature using the public key received from the sender, and then computes a hash value. If the computed hash value is the same as that decrypted from the digital signature, the authentication succeeds. Otherwise, the authentication fails.

When a peer has multiple peers, PSK authentication requires that the same pre-shared key be configured on all peers. This authentication method can be easily implemented on small-scale networks but has low security. RSA signature authentication provides high security but requires digital certificates issued by a CA. This authentication method is applicable to large-scale networks.

IKE supports the following authentication algorithms: AES-XCBC-MAC-96, MD5, SHA1, SHA2-256, SHA2-384, and SHA2-512.

- Identity protection

After a key is generated, identity data is encrypted to ensure secure transmission.

IKE supports the following encryption algorithms: DES, 3DES, AES-128, AES-192, and AES-256.

- **DH**

Diffie-Hellman (DH) is a public key exchange mechanism that generates key materials and uses ISAKMP messages to exchange key materials between the initiator and responder. Then the devices at both ends calculate the same symmetric key to generate the encryption key and authentication key. The two devices do not exchange the real key in any cases. DH key exchange is the core part of IKE.

The MD5, SHA1, DES, 3DES and AES algorithms can use the DH algorithm to enable sharing of a symmetric key between two parties.

DH uses key groups to define the key length. A longer key indicates a stronger key.

Table 6-4 DH key group

Key Group	Key Length
1	768 bits
2	1024 bits
5	1536 bits
14	2048 bits
15	3072 bits
16	4096 bits
19	256 bits ECP (Elliptic Curve Groups modulo a Prime)
20	384 bits ECP
21	521 bits ECP

- **PFS**

Perfect Forward Secrecy (PFS) is a security feature that protects security of other keys in case a key is deciphered, because these keys are independent of one another. The key used in an IPsec SA is derived from the key used in an IKE SA. An IKE SA generates one or more pairs of IPsec SAs through negotiation. After obtaining the IKE key, an attacker may collect enough information to calculate the key used in the IPsec SA. To ensure security of the key, PFS performs an additional DH key exchange.

 **NOTE**

- The MD5 and SHA1 authentication algorithms are insecure. The more secure AES-XCBC-MAC-96, SHA2-256, SHA2-384, or SHA2-512 algorithm is recommended.
- The DES and 3DES encryption algorithms are insecure. The more secure AES algorithm is recommended.

Comparison Between IKEv1 and IKEv2

IKE has two versions: IKEv1 and IKEv2. IKEv2 has the following advantages over IKEv1:

- Simplifies SA negotiation and improves negotiation efficiency.
IKEv1 goes through two phases to negotiate the key and establish SAs for IPsec. In phase 1, two IKE peers negotiate to establish a secure channel, IKE SA. In phase 2, IKE peers establish a pair of IPsec SAs using the secure channel established in phase 1. IKEv2 generates the key and establishes SAs for IPsec in just one negotiation, simplifying the negotiation process. For details about IKEv1 negotiation and IKEv2 negotiation, see [6.2.2.2 Establishing an SA Through IKEv1 Negotiation](#) and [6.2.2.3 Establishing an SA Through IKEv2 Negotiation](#).
- Fixes many cryptographic vulnerabilities, enhancing security.

6.2.2 IPsec Fundamentals

IPsec establishes bidirectional **security associations** between IPsec peers to form a secure IPsec tunnel, imports data flows to be protected to the tunnel, and then uses **security protocols** to encrypt and authenticate the data passing through the tunnel to securely transmit the data over the Internet.

IPsec SAs can be established manually or through IKEv1 or IKEv2 auto-negotiation. In manual mode, all the parameters used to establish an SA need to be manually configured and maintained, and an SA is established when parameters on two IPsec peers match and IPsec SA negotiation succeeds. In IKE auto-negotiation mode, only IKE negotiation information needs to be configured, and an SA is established and maintained through IKE auto-negotiation. The IKE auto-negotiation mode is recommended because it is easy to configure and maintain and has a secure key. This document describes only the process of establishing an SA through IKE auto-negotiation.

6.2.2.1 Defining IPsec Protected Data Flows

IPsec supports the following method to define data flows to be protected:

- Using an ACL
On an IPsec tunnel established in manual mode or IKE negotiation mode, data flows that will be protected by IPsec can be defined by an ACL. The packets matching permit clauses in the ACL are protected, and those matching the deny clauses are not protected. The ACL can define packet attributes such as the IP address, port number, and protocol type, which provide flexibility in defining IPsec policies.
- Using an IPsec profile
In this method, an IPsec tunnel is established through IPsec virtual tunnel interfaces, and these interfaces protect all the packets routed to them. An IPsec virtual tunnel interface is a type of Layer 3 logical interface.
This method has the following advantages:
 - Simplifies configuration.
You only need to import data flows to be protected to an IPsec tunnel interface, and do not need to use an ACL to define the characteristics of traffic to be encrypted and decrypted.
 - Supports more types of traffic.
An IPsec tunnel interface protects traffic of dynamic routing protocols and multicast traffic through GRE over IPsec.

6.2.2.2 Establishing an SA Through IKEv1 Negotiation

IKEv1 negotiation goes through two phases: In phase 1, two IPsec peers negotiate and establish a secure tunnel (an IKE SA). In phase 2, two IPsec peers establish a pair of IPsec SAs for secure data transmission through the secure tunnel established in phase 1.

IKEv1 Negotiation Phase 1

Phase 1 needs to establish an IKE SA. After an IKE SA is established, all the Internet Security Association and Key Management Protocol (ISAKMP) messages transmitted between the two IPsec peers will be encrypted and authenticated. The secure tunnel established in phase 1 enables IPsec peers to communicate securely in phase 2. An IKE SA is bidirectional, so only one IKE SA needs to be established between two IPsec peers.

Phase 1 supports two negotiation modes: main mode and aggressive mode.

The main mode uses six ISAKMP messages to implement three bidirectional exchanges, as shown in [Figure 6-10](#). The three exchanges are as follows:

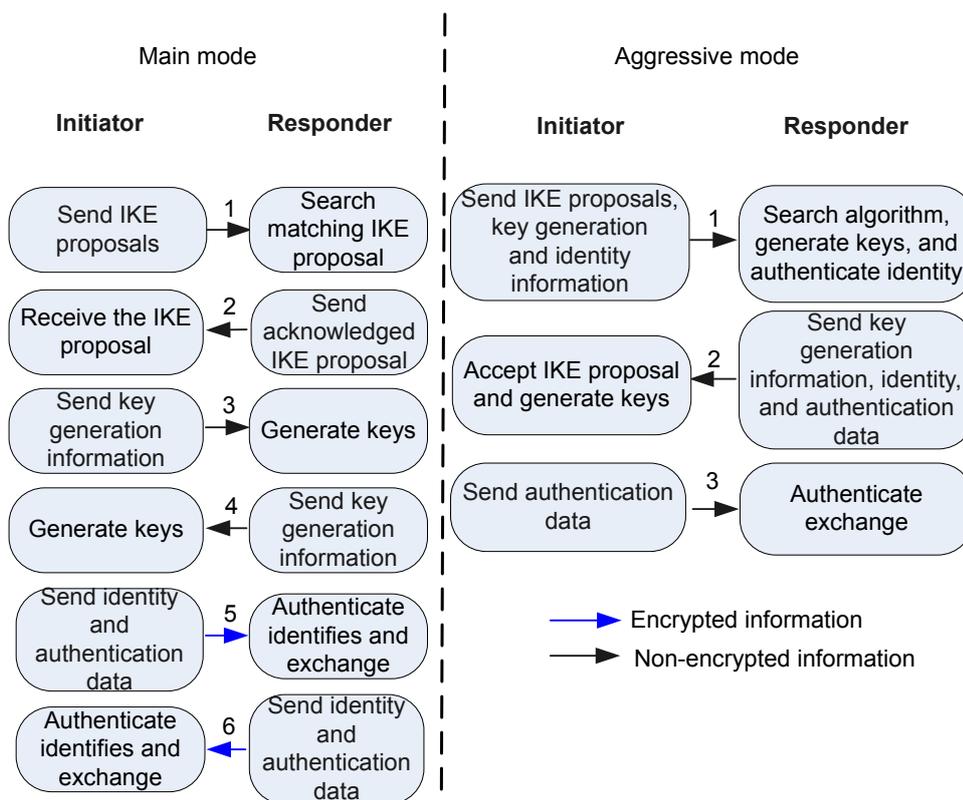
1. Messages (1) and (2) are used for policy exchange.
The initiator sends one or multiple IKE proposals to the responder. The responder searches for the first matching IKE proposal and then sends the matching proposal to the initiator. IKE proposals of the initiator and responder match if they have the same encryption algorithm, authentication algorithm, authentication method, and Diffie-Hellman group identifier.
2. Messages (3) and (4) are used for key information exchange.
The initiator and responder exchange the Diffie-Hellman public value and nonce value to generate the IKE SA authentication and encryption key.
3. Messages (5) and (6) are used for identity and authentication information exchange.
The initiator and responder use the generated key to authenticate each other and the information exchanged in main mode.

NOTE

- An IKE proposal is a set of algorithms used to secure IKE negotiation, including encryption algorithm, authentication algorithm, Diffie-Hellman group, and authentication method.
- A nonce value is a random number that guarantees IKE SA liveness and protects against replay attacks.

The aggressive mode uses only three messages. Messages (1) and (2) are used to negotiate IKE proposal and exchange the Diffie-Hellman public value, mandatory auxiliary information, and identity information. Message (2) also contains the identity information sent by the responder to the initiator for authentication. Message (3) is used by the responder to authenticate the initiator. [Figure 6-10](#) shows IKEv1 negotiation phase 1.

Figure 6-10 IKEv1 negotiation phase 1



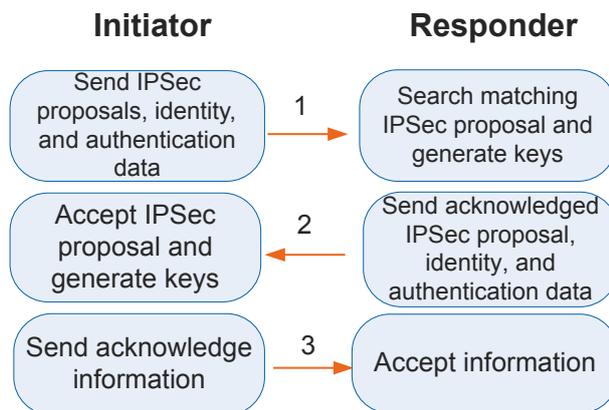
Compared with the main mode, the aggressive mode reduces the number of exchanged messages and speeds up the negotiation, but it does not encrypt identity information. Although the aggressive mode does not provide identity protection, it applies to the following scenarios:

- If the IP address of the initiator is variable or unknown, and both the initiator and responder need to use a pre-shared key to establish an IKE SA, only the aggressive mode can be used.
- If the initiator already knows the IPsec policy used by the responder, it is faster to establish an IKE SA in aggressive mode.

IKEv1 Negotiation Phase 2

Phase 2 establishes IPsec SAs used to securely transmit data and generates the key for data transmission. The quick mode is used in phase 2. This mode uses the key generated in phase 1 to verify the integrity of ISAKMP messages and identities of the initiator and responder, and to encrypt ISAKMP messages, ensuring exchange security. [Figure 6-11](#) shows IKEv1 negotiation phase 2.

Figure 6-11 IKEv1 negotiation phase 2



Phase 2 establishes two IPsec SAs through three ISAKMP messages:

1. The initiator sends local security parameters and identity authentication information to the responder.

Security parameters include protected data flows and parameters to be negotiated such as IPsec proposal. Identity authentication information includes the key generated in phase 1 and key materials generated in phase 2, and can be used to authenticate the peer again.

NOTE

An IPsec proposal is a set of protocols and algorithms used for negotiation, including security protocol, encryption algorithm, and authentication algorithm.

2. The responder sends confirmed security parameters and identity authentication information, and generates a new key.

The encryption key and authentication key used for IPsec SA data transmission are generated based on the key generated in phase 1 and parameters such as SPI and protocols, to ensure that each IPsec SA has a unique key.

If **Perfect Forward Secrecy (PFS)** needs to be enabled, the shared key calculated through DH algorithm needs to be used again to generate the encryption key and authentication key. During parameter negotiation, a DH key group needs to be negotiated for PFS.

3. The initiator sends confirmed information to communicate with the responder. IKEv1 negotiation then ends.

6.2.2.3 Establishing an SA Through IKEv2 Negotiation

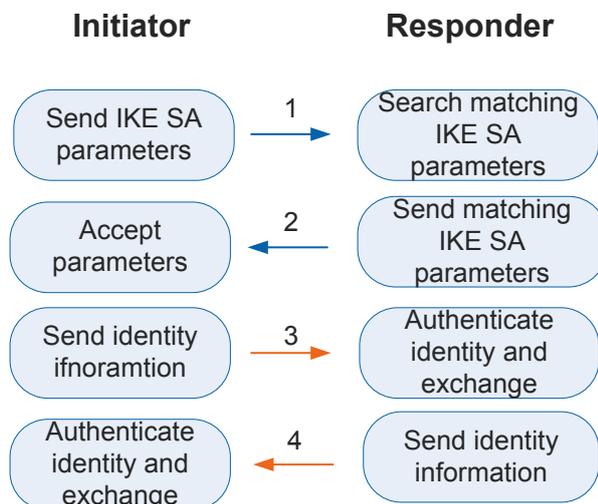
The process of establishing an SA through IKEv2 negotiation is much simpler than that through IKEv1 negotiation. To establish a pair of IPsec SAs, IKEv1 requires two phases: main mode (or aggressive mode) + quick mode. At least nine messages are exchanged in main mode + quick mode, and at least six messages are exchanged in aggressive mode + quick mode. In normal cases, IKEv2 can establish a pair of IPsec SAs only through four messages in two exchanges. One additional **Create_Child_SA Exchange** can be used to establish another pair of IPsec SAs if required, during which only two messages are exchanged.

IKEv2 defines three exchanges: Initial Exchanges, Create_Child_SA Exchange, and Informational Exchange.

Initial Exchanges

In normal cases, IKEv2 can establish the first pair of IPsec SAs through Initial Exchanges. Mapping phase 1 in IKEv1 negotiation, Initial Exchanges involves four messages in two exchanges, as shown in [Figure 6-12](#).

Figure 6-12 Initial Exchanges process



Messages (1) and (2) are used in exchange 1 (called `IKE_SA_INIT`) to negotiate IKE SA parameters in plain text, including the encryption key and authentication key, random number, and DH key. After `IKE_SA_INIT` is complete, a shared key material is generated, from which all IPsec SA keys are derived.

Messages (3) and (4) are used in exchange 2 (called `IKE_AUTH`) to authenticate identities of the two parties and the first two messages, and to negotiate IPsec SA parameters in encryption mode. IKEv2 supports authentication modes including the RSA signature, pre-shared key, and Extensible Authentication Protocol (EAP). EAP authentication is implemented in IKE as an additional `IKE_AUTH` exchange. The initiator does not set the authentication payload in message (3) to indicate that EAP authentication is required.

Create_Child_SA Exchange

When an IKE SA requires multiple pairs of IPsec SAs, `Create_Child_SA` Exchange is performed to negotiate these IPsec SAs. In addition, `Create_Child_SA` Exchange can be performed for IKE SA re-negotiation.

`Create_Child_SA` Exchange involves two messages in one exchange and maps [phase 2 in IKEv1 negotiation](#). The initiator can be the initiator or responder in Initial Exchanges. `Create_Child_SA` Exchange can be performed only after Initial Exchanges are complete. Exchange messages in `Create_Child_SA` Exchange are protected by keys negotiated in Initial Exchanges.

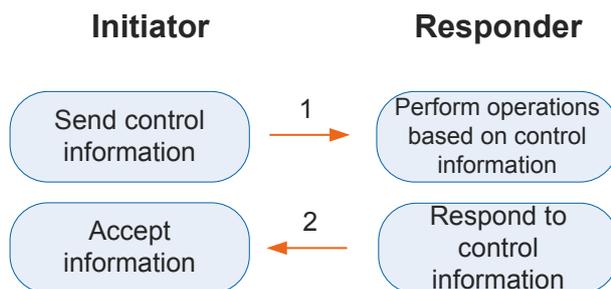
Similar to IKEv1, if PFS is enabled, `Create_Child_SA` Exchange requires an additional DH exchange to generate a new key material. All keys of child SAs are derived from this key material.

Informational Exchange

IKEv2 peers perform Informational Exchange to exchange control information, including error information and notifications, as shown in [Figure 6-13](#).

Informational Exchange must be performed with IKE SA protection, that is, performed after Initial Exchanges are complete. Control information may belong to an IKE SA or a child SA. Therefore, Informational Exchange must be protected by the IKE SA or the IKE SA that generates the child SA accordingly.

Figure 6-13 Informational Exchange process



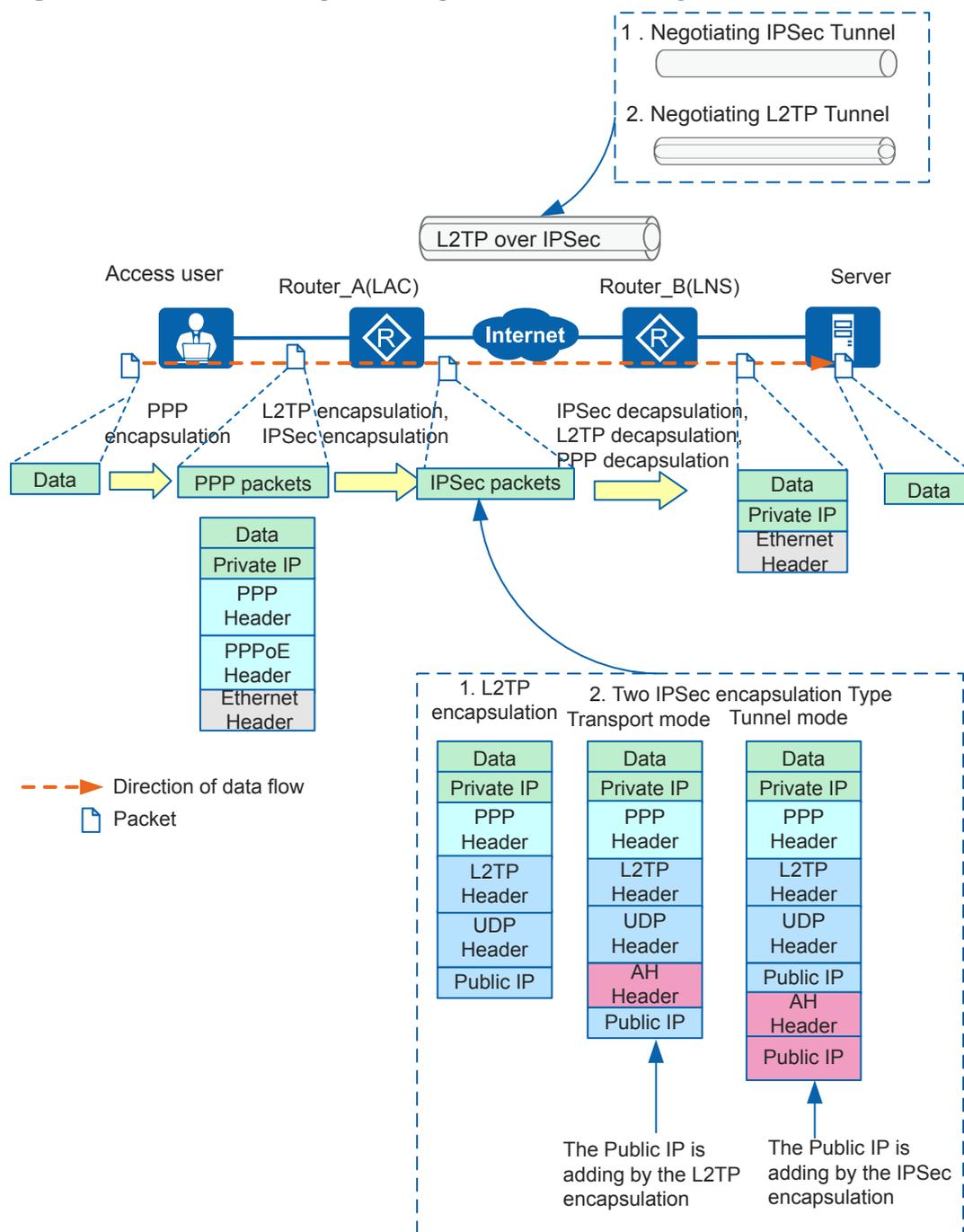
6.2.3 IPsec Enhancements

6.2.3.1 L2TP over IPsec

L2TP over IPsec encapsulates packets using L2TP and then IPsec. It uses L2TP to implement user authentication and address allocation and IPsec to ensure secure communication. L2TP over IPsec ensures that branches or traveling employees connect to the headquarters.

[Figure 6-14](#) illustrates how L2TP over IPsec allows branches to connect to the headquarters.

Figure 6-14 L2TP over IPsec packet encapsulation and tunnel negotiation



Packets are encapsulated by L2TP, and then by IPsec. In the IP header added during IPsec encapsulation, the source IP address is the IP address of the interface to which the IPsec policy is applied, and the destination IP address is the IP address of the peer interface to which the IPsec policy on the remote peer is applied.

IPsec protects the data flows from the source to the destination of the L2TP tunnel. In the new IP header added during L2TP encapsulation, the source IP address is the address of the L2TP source interface, and the destination IP address is the address of the L2TP destination interface. When a branch connects to the headquarters, the source address of the L2TP tunnel

is the IP address of the outbound interface on the LAC, and the designation address is the IP address of the inbound interface on the LNS.

A public IP address is added to the header in L2TP encapsulation, and one more public IP address is added in tunnel mode. As a result, the packets are larger and more packets will be fragmented in tunnel mode. Therefore, the transport mode of L2TP over IPsec is recommended.

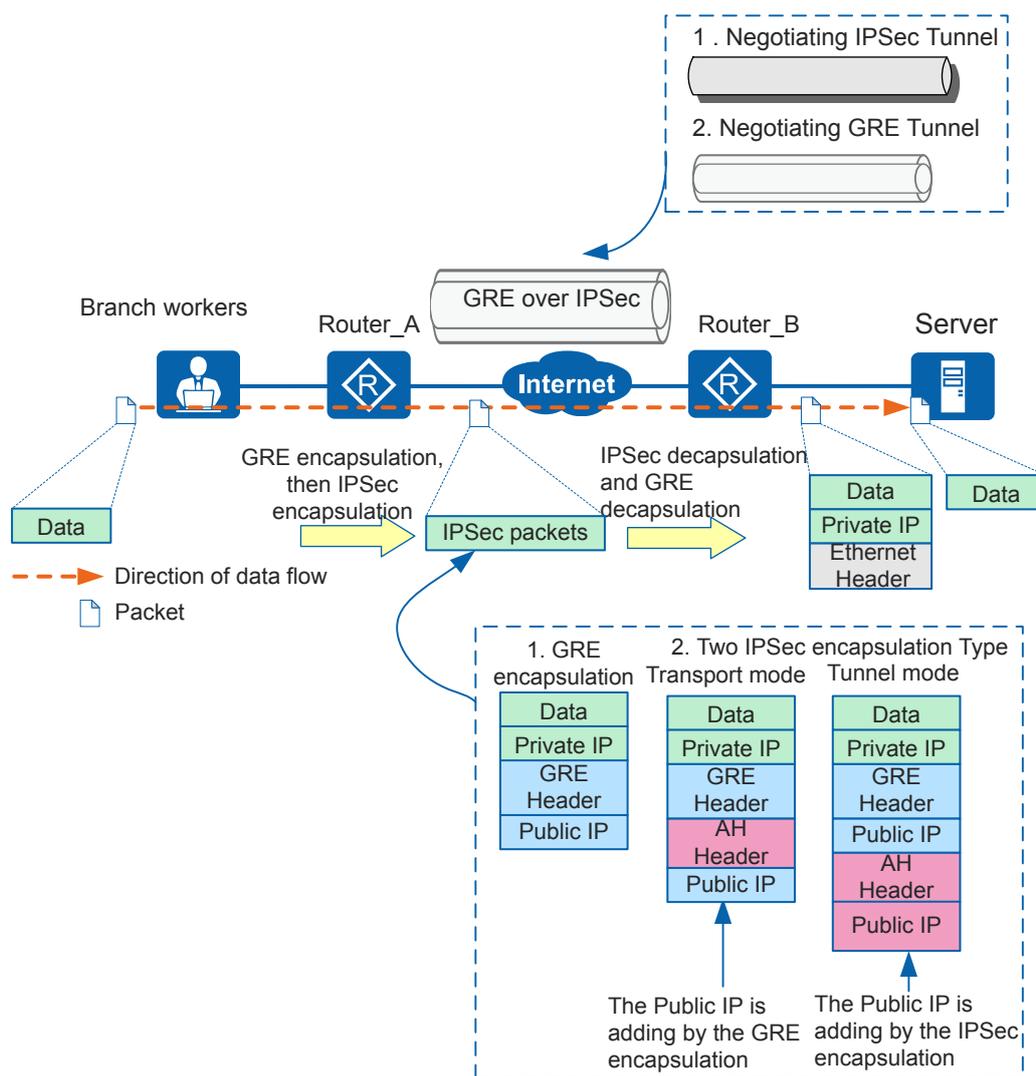
The L2TP over IPsec negotiation sequence and packet encapsulation process are the same for employees on the move and employees at branch offices. The difference is that, L2TP and IPsec encapsulation is performed on clients when employees on the move connect to the headquarters. The L2TP source address is the private address assigned to the client. The address can be any address in the address pool configured on the LNS. The destination address of the L2TP tunnel is the address of the inbound interface on the LNS.

6.2.3.2 GRE over IPsec

Integrating the advantages of both GRE and IPsec, GRE over IPsec uses GRE to encapsulate multicast, broadcast, and non-IP packets into common IP packets, and uses IPsec to provide secure communication for encapsulated IP packets. Therefore, broadcast and multicast services such as video conference or messages of dynamic routing protocols, can be securely transmitted between the headquarters and branch.

GRE over IPsec encapsulates packets using GRE, and then IPsec. The encapsulation can be implemented in tunnel mode and transport mode. The tunnel mode uses an extra IPsec header, which increases packet size and makes packets more likely to be fragmented. Therefore, the transport mode is recommended.

Figure 6-15 Packet encapsulation and tunnel negotiation in GRE over IPsec



In the IP header added during IPsec encapsulation, the source IP address is the IP address of the interface to which the IPsec policy is applied, and the destination IP address is the IP address of the peer interface to which the IPsec policy on the remote peer is applied.

IPsec protects the data flows from the GRE source address to the GRE destination address. In the IP header added during GRE encapsulation, the source address is the source address of the GRE tunnel, and the destination address is the destination address of the GRE tunnel.

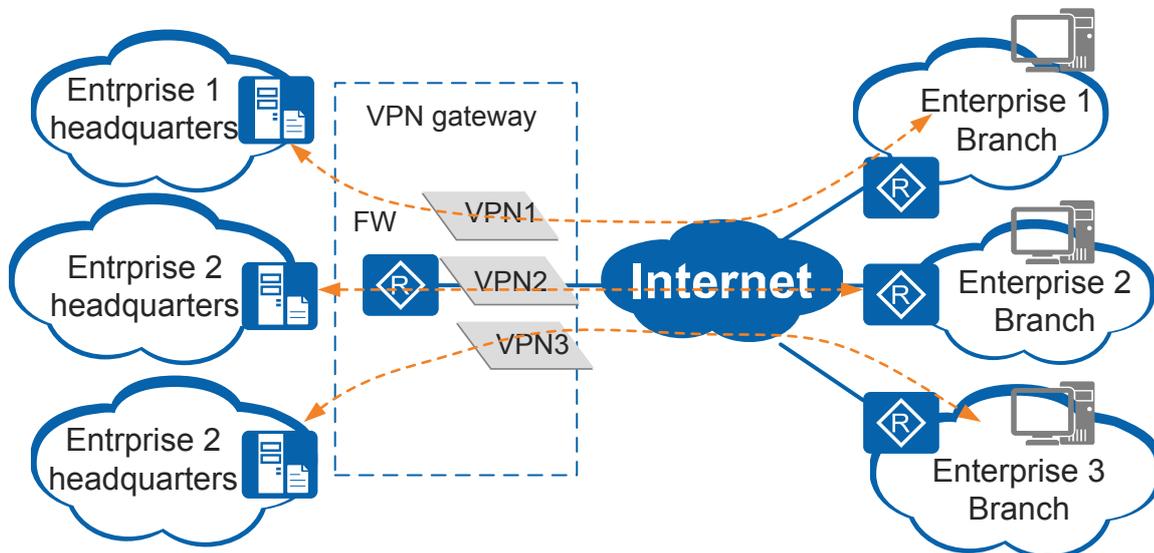
6.2.3.3 IPsec Multi-instance

IPsec multi-instance is used to provide the firewall lease service to isolate networks of small enterprises.

As shown in **Figure 6-16**, branches of three small enterprises share a VPN gateway. The three enterprise networks must be isolated. IP addresses of each enterprise are planned independently, and therefore IP addresses on different private networks may overlap. The IPsec multi-instance function can be configured on the VPN gateway to bind IPsec tunnels of

the three enterprises to different VPN instances. This ensures that packets with the same destination IP addresses can be correctly forwarded.

Figure 6-16 Typical IPsec multi-instance network



6.2.3.4 Efficient VPN

On an enterprise network with many branches, IPsec must be configured on headquarters and branch gateways. These IPsec configurations are complex and difficult to maintain. IPsec Efficient VPN can solve these problems with its high security, reliability, and flexibility. It has become the first choice for enterprises to establish VPNs.

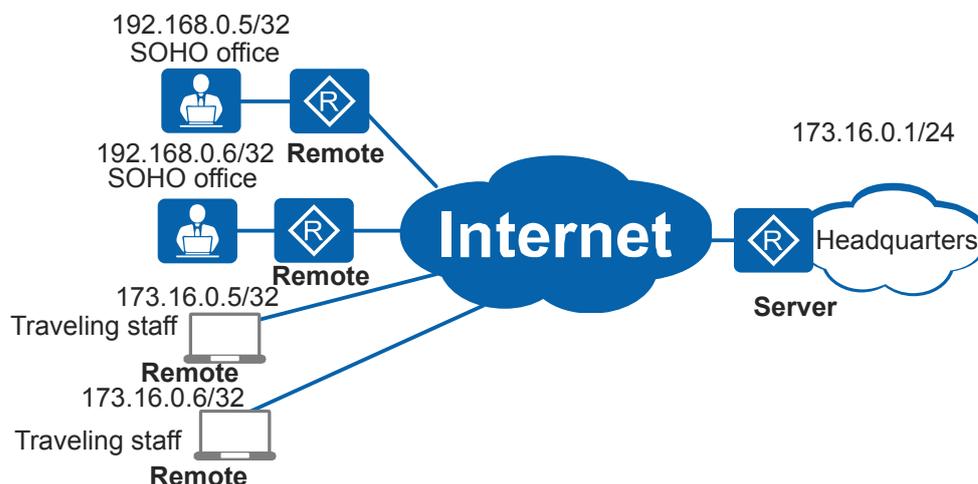
Efficient VPN uses the client/server model. It concentrates IPsec and other configurations on the Efficient VPN server (headquarters gateway). When basic parameters for establishing an SA are configured on the remote devices (branch gateways), the remote devices initiate a negotiation and establish an IPsec tunnel with the server. After IPsec tunnels are established, the Efficient VPN server allocates other IPsec attributes and network resources to the remote devices. Efficient VPN simplifies configurations and maintenance of IPsec and network resources for the branches. In addition, Efficient VPN supports automatic upgrades on remote devices.

Operation Modes

- Client mode
 - a. When a remote device requests an IP address from the Efficient VPN server, a loopback interface is dynamically created on the remote device and the IP address obtained from the server is assigned to the loopback interface.
 - b. The remote device automatically enables NAT to translate its original IP address into the obtained IP address, and then uses this IP address to establish an IPsec tunnel with the headquarters.
 - c. The remote device automatically enables NAT to translate its original IP address into the obtained IP address, and then uses this IP address to establish an IPsec tunnel with the headquarters.

The client mode applies to scenarios where traveling staff or small-scale branches connect to the headquarters network through private networks, as shown in [Figure 6-17](#). In client mode, devices connected to the Efficient VPN server or remote devices can use the same IP address. However, the number of devices allowed depends on the number of IP addresses assigned by the Efficient VPN server.

Figure 6-17 Client mode



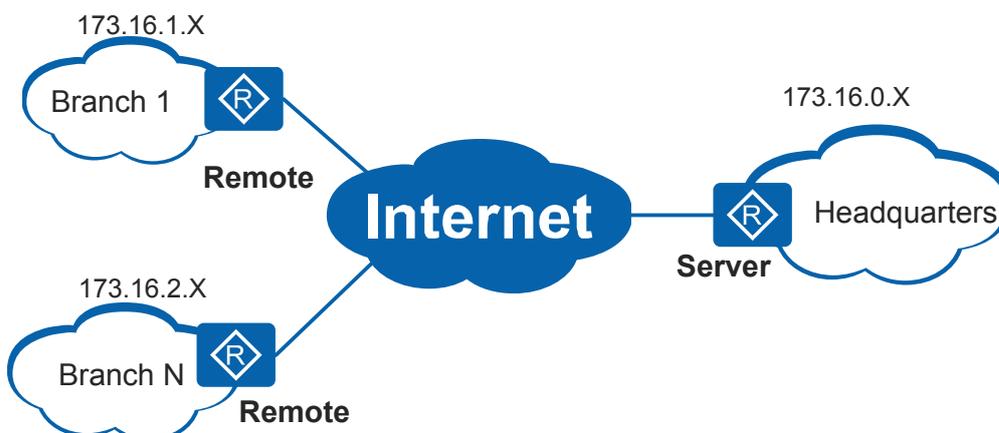
NOTE

Traveling staff use software to establish a virtual network adapter on a PC. The virtual network adapter then uses parameters such as addresses sent by the Efficient VPN server.

- **Network mode**

In network mode, a remote device does not apply to the Efficient VPN server for an IP address. Therefore, NAT is not automatically enabled in network mode. [Figure 6-18](#) shows the network mode.

Figure 6-18 Network mode



The network mode applies to scenarios where IP addresses of the headquarters and branches are planned uniformly. Ensure that IP addresses do not conflict.

- Network-plus mode
Compared with the network mode, the remote device applies to the Efficient VPN server for an IP address in network-plus mode. IP addresses of branches and headquarters are configured beforehand. A remote device applies to the Efficient VPN server for an IP address. The Efficient VPN server uses the IP address to perform ping, STelnet, or other management and maintenance operations on the remote device. NAT is not automatically enabled on the remote device.
- Network-auto-cfg mode
Compared with the network-plus mode, the remote device applies to the Efficient VPN server for an IP address pool in network-auto-cfg mode. The IP address pool is used for allocating addresses to users.

The Efficient VPN server also delivers the following resources in addition to parameters for establishing an IPsec tunnel:

- Network resources including the DNS domain name, DNS server IP addresses, and WINS server IP addresses
The Efficient VPN server delivers the preceding resources so that branches can access them on the Efficient VPN server.
- ACL resources
The Efficient VPN server delivers headquarters network information defined in an ACL to the remote device. The ACL defines the headquarters subnets that branches can access. Traffic not destined for the subnets specified in the ACL is directly forwarded to the Internet. Such traffic does not pass through the IPsec tunnel.

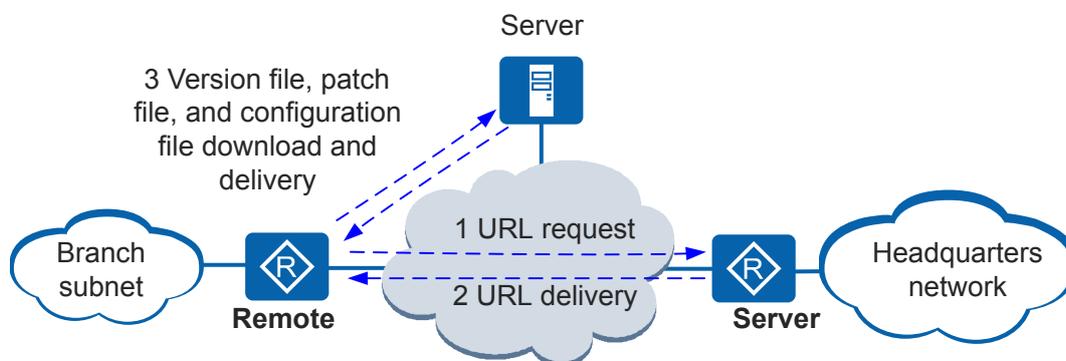
 **NOTE**

In the Network-auto-cfg mode, delivering of parameters defined in the ACL is not supported.

Automatic Upgrade of Efficient VPN Remote Devices

The server defines the uniform resource locator (URL) used to upgrade remote devices. A remote device automatically downloads the version file, patch file, and configuration file according to the URL configuration file to complete an upgrade. Automatic upgrade facilitates network deployment and maintenance. [Figure 6-19](#) shows the procedure for automatically upgrading the remote device.

Figure 6-19 Automatic upgrade of remote devices



1. A remote device with basic IPsec Efficient VPN configuration connects to the headquarters.
2. The remote device applies to the server for the address and version number of the URL configuration file.
3. The remote device obtains the address and version number of the URL configuration file and downloads the URL configuration file from the specified server.
4. The remote device downloads the corresponding version file, patch file, and configuration file according to the URL configuration file.
5. The remote device performs the upgrade according to the version file, patch file, and configuration file.

6.2.4 IPsec Reliability

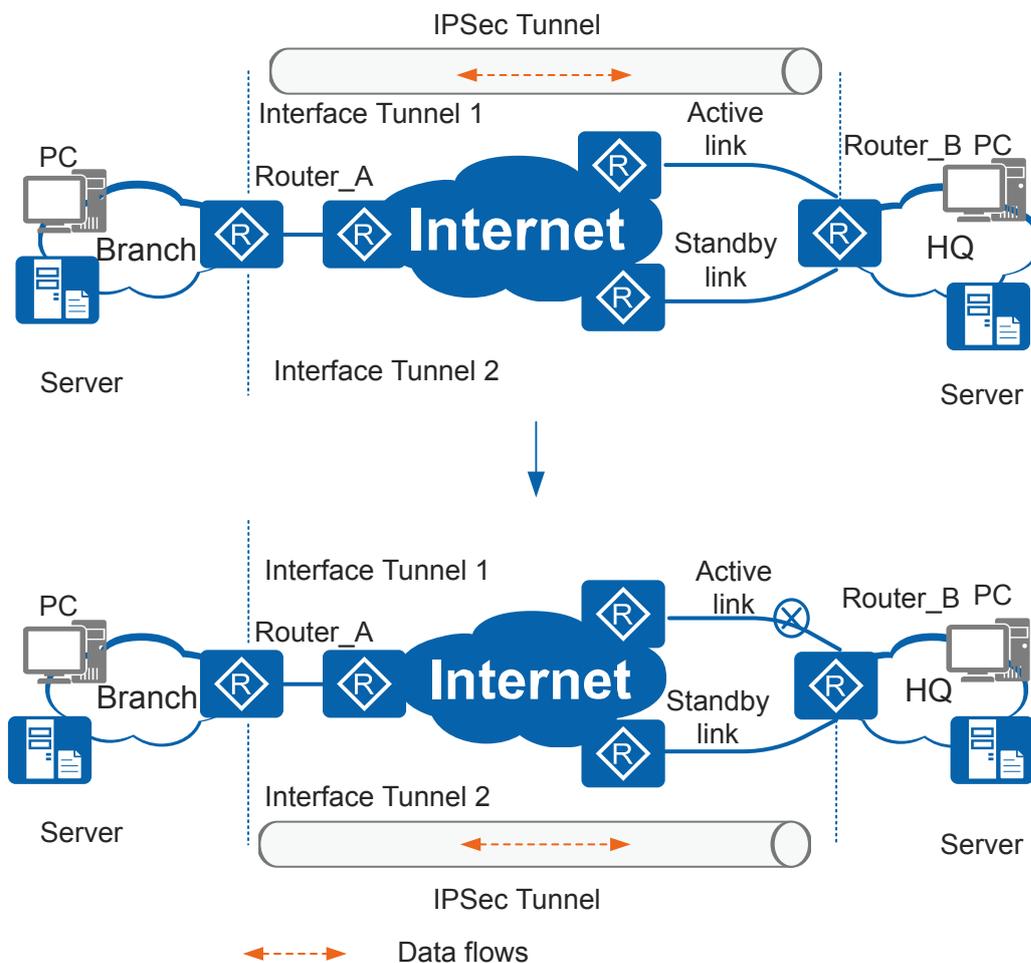
6.2.4.1 Link Redundancy

To improve network reliability, an enterprise connects a branch network to the headquarters network through two or more links. When a link fails, services are immediately switched to another link. The device provides two redundancy modes: active/standby IPsec links and IPsec multi-link.

Active and Standby IPsec Links

In [Figure 6-20](#), Router_A connects to Router_B through active/standby links. Two tunnel interfaces are created on Router_A and they borrow the IP address of the same physical interface. Different IPsec policies are applied to the two tunnel interfaces to create active and standby IPsec tunnels. Different IPsec policies are applied to the two physical interfaces on Router_B. When the active link fails, traffic is switched to the standby link. A new IPsec tunnel is established on the standby link, and the old IPsec tunnel is deleted.

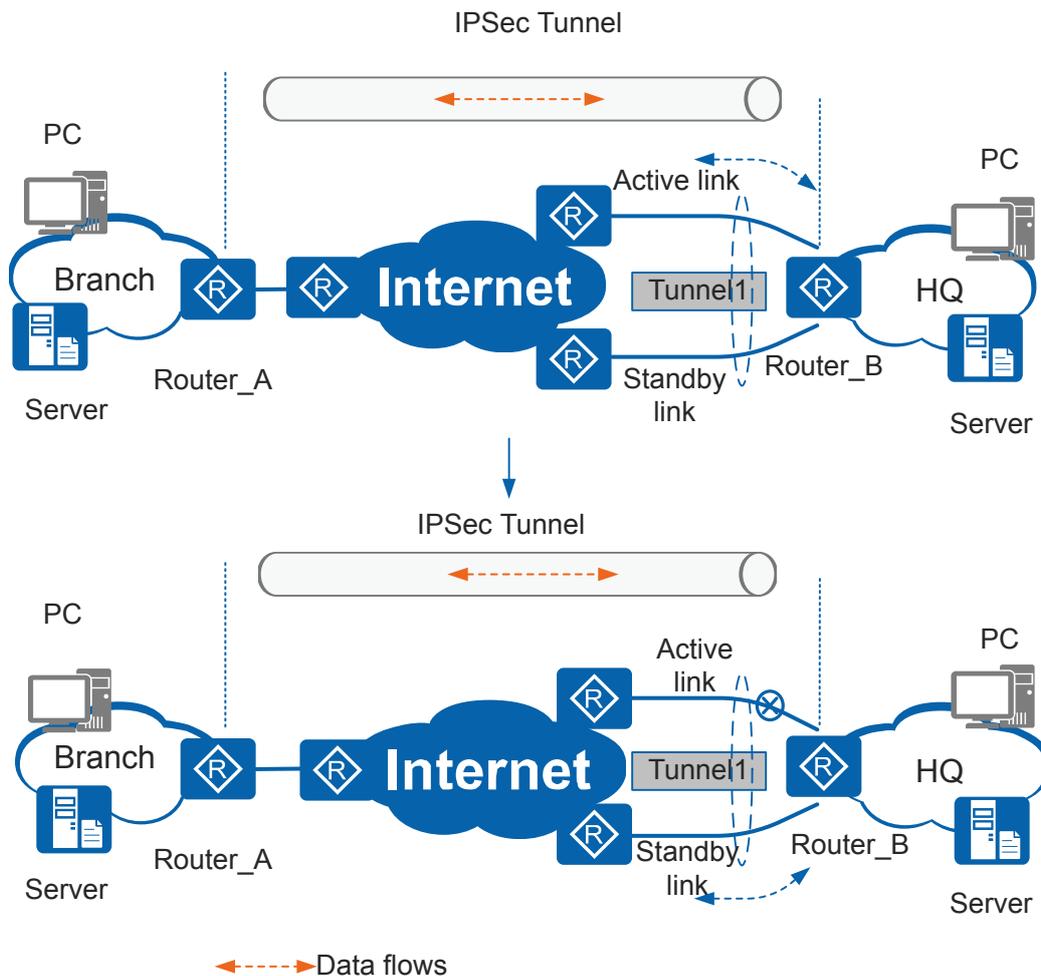
Figure 6-20 Active and standby IPsec links



IPsec Multi-link

In [Figure 6-21](#), Router_A connects to Router_B through active/standby links. An IPsec tunnel is established between the physical interface of Router_A and tunnel interface of Router_B. Traffic is processed by IPsec on the tunnel interface and sent out by a physical interface according to the routing table. When the active link fails, the route is unreachable and traffic is switched to the standby link. Re-negotiation is not required for the IPsec tunnel, so traffic can be rapidly switched.

Figure 6-21 Using the tunnel interface to implement link redundancy



A tunnel interface can implement multi-link redundancy. This mode is more simple and switches traffic faster than the active/standby links.

NOTICE

In the scenario where an IPsec gateway is connected to different ISP networks or the same ISP network but the active and standby links are connected to different access routers of the same ISP network across LANs or areas, if the active link becomes faulty, the device on the standby link may discard the IPsec packets whose source address belongs to a different ISP network or access router. Therefore, before configuring link redundancy, check whether active/standby link switching is allowed in the actual network environment.

6.3 Application Scenarios for IPsec

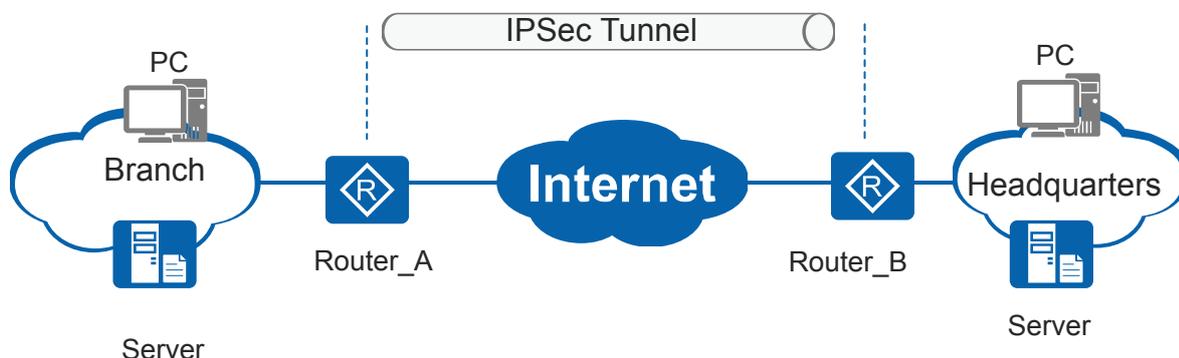
6.3.1 Using IPsec VPN to Implement Secure Interconnection Between LANs

The headquarters and branches of an enterprise are interconnected in various ways.

Site-to-Site VPN — IPsec

A site-to-site VPN, also called LAN-to-LAN VPN or gateway-to-gateway VPN, is used to set up an IPsec tunnel between two gateways, implementing secure access of LANs. [Figure 6-22](#) shows a typical site-to-site IPsec VPN network.

Figure 6-22 Typical site-to-site IPsec VPN network



This network requires that the two gateways on both ends of the tunnel have fixed IP addresses or fixed domain names, and both parties be able to initiate a connection.

NOTE

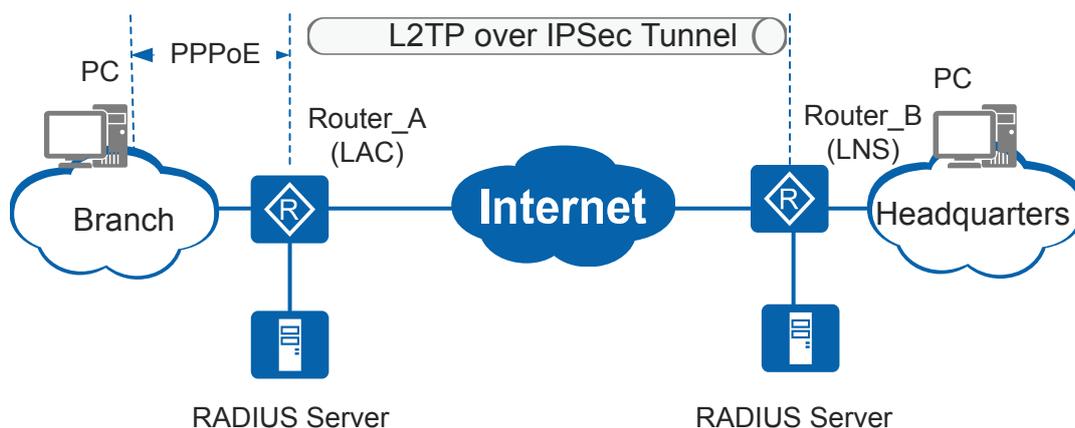
- A Router can serve as both an IPsec gateway and a NAT gateway.
- When a NAT device exists between two IPsec gateways, Routers support IPsec NAT traversal.

Site-to-Site VPN — L2TP over IPsec

L2TP over IPsec encapsulates packets using L2TP before transmitting them using IPsec. L2TP and IPsec are used together to allow branches to securely access VPNs by dialing the L2TP access concentrator (LAC). Branches use L2TP to dial the LAC and obtain private IP addresses on the headquarters network. IPsec is used to ensure communication security during this process.

[Figure 6-23](#) shows a network for the branch to access the headquarters through an L2TP over IPsec tunnel. The outbound interfaces of the LAC (FW_A) and L2TP network server (LNS) (FW_B) have fixed IP addresses. A user in the branch dials FW_A through PPPoE. FW_A then initiates a tunnel setup request to FW_B over the Internet. An L2TP over IPsec is set up between FW_A and FW_B. Then FW_A authenticates the user, and FW_B can also authenticate the user again if the user is successfully authenticated by FW_A. After the user is successfully authenticated by FW_B, FW_B assigns a private IP address to the user.

Figure 6-23 Branch accessing the headquarters through an L2TP over IPsec tunnel

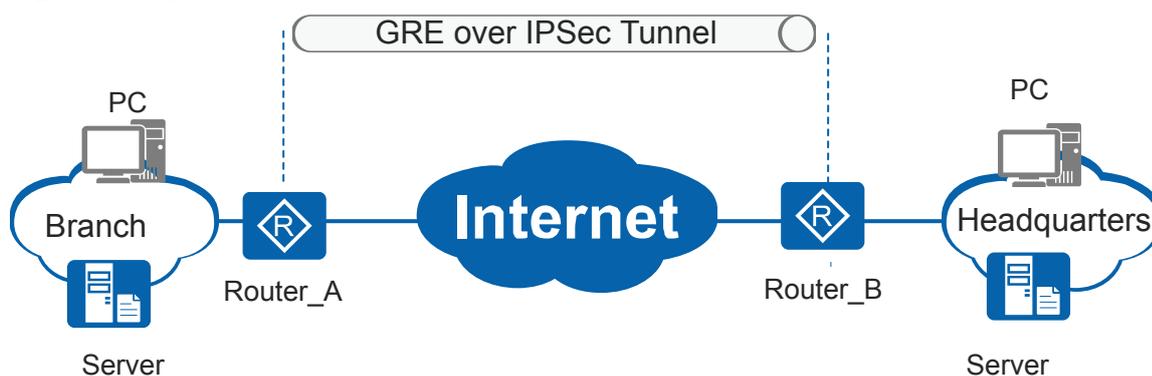


Site-to-Site VPN — GRE over IPsec

Generic Routing Encapsulation (GRE) is a generic tunneling protocol that encapsulates multicast, broadcast, and non-IP packets. GRE, however, provides only simple password authentication but not data encryption, and therefore cannot ensure data transmission security. IPsec provides high data transmission security but cannot encapsulate multicast, broadcast, or non-IP packets. Leveraging advantages of GRE and IPsec, GRE over IPsec encapsulates multicast, broadcast, and non-IP packets into common IP packets. For example, to hold a video conference between the branch and headquarters, use GRE over IPsec to transmit service traffic on an IPsec VPN.

Figure 6-24 shows a typical GRE over IPsec VPN network.

Figure 6-24 Typical GRE over IPsec VPN network

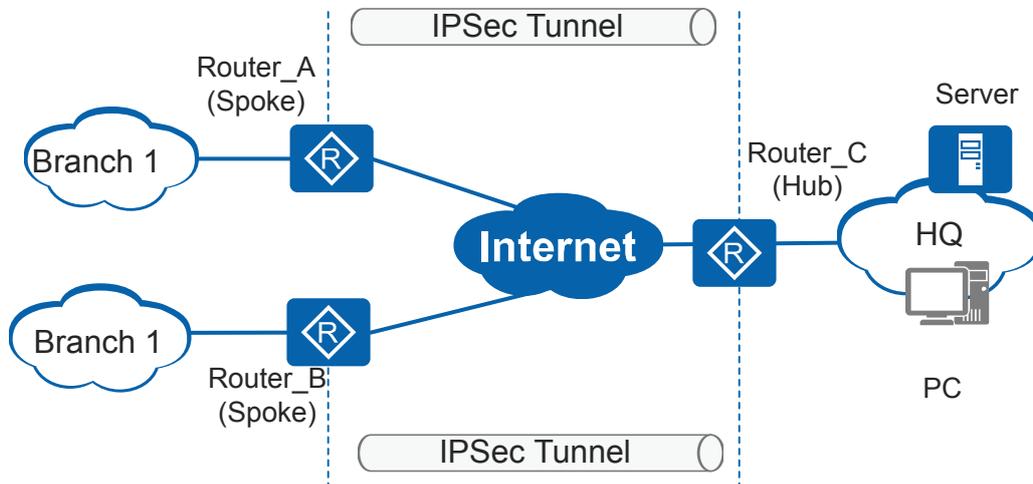


GRE over IPsec supports the transport and tunnel encapsulation modes. Compared to the transport mode, the tunnel mode has an IPsec header, which makes the packet longer and easier to be fragmented. Therefore, GRE over IPsec in transport mode is recommended.

Hub-Spoke VPN

In most cases, the headquarters of an enterprise are connected to multiple branches through IPsec VPN tunnels. **Figure 6-25** shows a typical network.

Figure 6-25 Typical hub-spoke IPsec VPN network



The headquarters has a fixed public IP address or fixed domain name. Branches support static or dynamic public IP addresses and private IP addresses. Data traffic is transmitted in the following scenarios:

- Branches do not need to communicate with each other.
Deploy IPsec VPN between the headquarters and branches.

- Branches need to communicate with each other.

If branches access the Internet using dynamic public IP addresses, traditional IPsec VPN will lead to a communication failure between branches. Communication data between branches has to be forwarded through the headquarters. This consumes the CPU and memory resources of the hub (FW_C). In addition, the headquarters must encapsulate and decapsulate traffic between branches, causing additional network delay.

To resolve this problem, deploy Dynamic Smart VPN (DSVPN) to set up VPN tunnels between branches using dynamic IP addresses. However, multipoint GRE (mGRE) tunnels do not have the encryption function and cannot ensure communication security. To achieve communication security, bind DSVPN with the IPsec security framework, that is, deploy DSVPN over IPsec. For details about DSVPN over IPsec, see [5.2.4 DSVPN Protected by IPsec](#).

6.3.2 Using IPsec VPN to Provide Secure Remote Access for Mobile Users

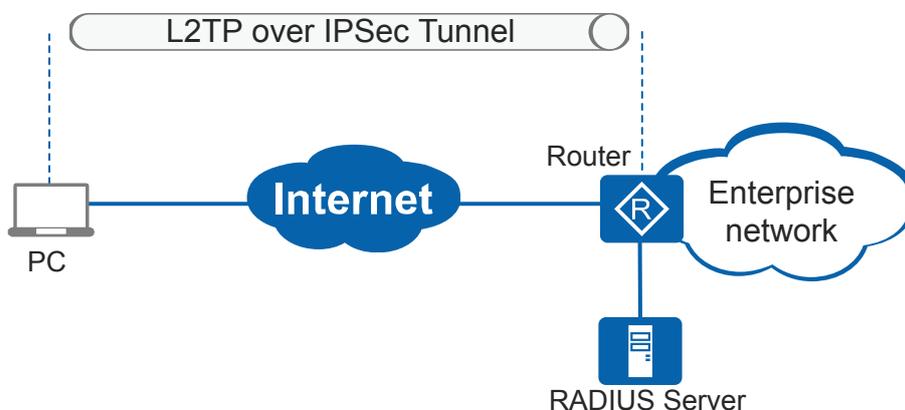
In public places, such as hotels and airports, traveling staff or partners connect to the core network through the insecure access network or public network such as the Internet to access internal resources of the core network. This process is called remote access. In remote access, traveling staff or partners access the core network through the insecure network; therefore, security is a major concern in remote access. IPsec VPN can be deployed to establish an

IPsec tunnel between a user terminal and the gateway of the core network. IPsec ensures secure and reliable data transmission.

As shown in [Figure 6-26](#), mobile users (such as traveling staff) use built-in VPN dial-up software of Windows or other dial-up software to dial to access the enterprise network. L2TP provides the user authentication function, but no encryption function. To ensure security, deploy L2TP over IPsec and set up an L2TP over IPsec tunnel between the PC and enterprise gateway Router. Packets are encapsulated using L2TP and encrypted using IPsec before being transmitted, ensuring communication security.

Access users are authenticated locally or remotely by the authentication server (RADIUS server, for example) in the headquarters. After authentication is successful, Router assigns private IP addresses within the headquarters network to users (PCs or mobile terminals).

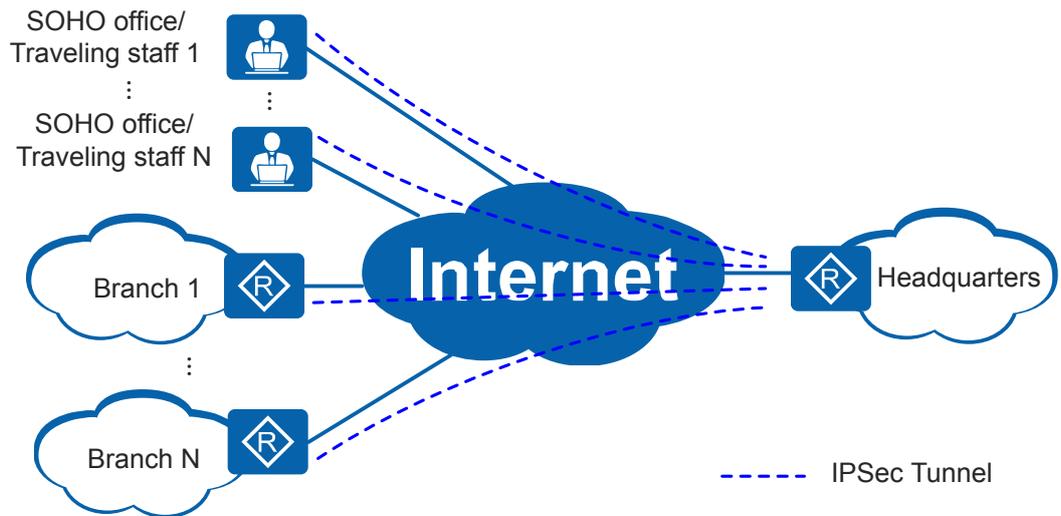
Figure 6-26 Remote access of mobile users using L2TP over IPsec



6.3.3 Secure LAN Interconnection Through Efficient VPN

In [Figure 6-27](#), when many branches and traveling staff need to communicate with the headquarters over IPsec tunnels, many similar or duplicate IPsec configurations and other network resource configurations need to be performed on the gateways of branches and headquarters. Efficient VPN provides a method to simplify the configurations. It allows you to perform complex configurations on the headquarters gateway and simple configurations on each branch gateway, freeing you from complex IPsec VPN configuration and maintenance.

Figure 6-27 Efficient VPN networking



6.4 Summary of IPsec Configuration Tasks

Two IPsec peers establish inbound and outbound security associations (SAs) to form a secure IPsec tunnel through which data packets can be transmitted securely on the Internet.

Table 6-5 lists IPsec configuration tasks.

Table 6-5 IPsec configuration tasks

Scenario	Description	Task
Using an ACL to establish an IPsec tunnel	<p>An ACL defines data flows to be protected. You need to configure an IPsec policy and apply the IPsec policy to an interface to protect IPsec packets. You can use an ACL to establish an IPsec tunnel in manual mode or IKE negotiation mode.</p> <p>SAs can be established in either of the following modes:</p> <ul style="list-style-type: none"> ● Manual mode: All information required by SAs must be manually configured. ● IKE negotiation mode: IPsec peers use IKE to negotiate keys and dynamically create and maintain SAs. <p>The manual mode applies to small-sized networks or scenarios where a few IPsec peers exist. The IKE negotiation mode applies to medium- and large-sized networks.</p>	6.7 Using an ACL to Establish an IPsec Tunnel
Using a tunnel interface to establish an IPsec tunnel	<p>An IPsec tunnel is established using a tunnel interface based on routes. In this mode, routes determine the data flows to be protected.</p> <p>You need to configure an IPsec profile and apply the IPsec profile to the IPsec tunnel interface to protect IPsec packets. All the packets routed to the IPsec tunnel interface are protected by IPsec.</p>	6.8 Using a Virtual Tunnel Interface to Establish an IPsec Tunnel

Scenario	Description	Task
Using the Efficient VPN policy to establish an IPsec tunnel	<p>Efficient VPN uses the client/server model. It concentrates IPsec and other configurations on the Efficient VPN server (headquarters gateway). When basic parameters for establishing an SA are configured on the remote devices (branch gateways), the remote devices initiate a negotiation and establish an IPsec tunnel with the server. After IPsec tunnels are established, the Efficient VPN server allocates other IPsec attributes and network resources to the remote devices. Efficient VPN simplifies configurations and maintenance of IPsec and network resources of branches.</p> <p>In addition, Efficient VPN supports automatic upgrades of remote devices.</p>	6.9 Establishing an IPsec Tunnel Using an Efficient VPN Policy

In manual mode, an ACL is used to establish an IPsec tunnel. In other modes, SAs are generated through IKE negotiation to establish an IPsec tunnel and an IKE peer needs to be configured and referenced.

6.5 Licensing Requirements and Limitations for IPsec

Involved Network Elements

None

Licensing Requirements

When using the Efficient VPN policy to establish an IPsec tunnel, note the following points:

- If a branch server needs to provide services for external users through NAT, the **nat static** command must be used on the remote device.
- When a remote device requests an IP address from the Efficient VPN server, a loopback interface is dynamically created on the remote device. Other services cannot be configured on the loopback interface.

For Efficient VPN-capable devices, their licensing requirements for the Efficient VPN function are as follows:

- AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S series: Efficient VPN is a basic feature of the device and is not under license control.
- AR2200-S&AR3200-S series: By default, Efficient VPN function is disabled on a new device. To use the Efficient VPN function, apply for and purchase the following license from the Huawei local office.
 - AR2200-S series: AR2200 Value-Added Security Package
 - AR3200-S series: AR3200 Value-Added Security Package

Impact on Performance

- The DH group value has impacts on IKE negotiation performance (such as the tunnel creation rate). A higher DH group value has greater impacts on IKE negotiation performance (for example, the tunnel creation rate greatly decreases).
- When the number of IPsec tunnels is larger than 50% of the maximum limit, high CPU usage alarms may be generated in a short period of time after the **undo ipsec policy** or **undo ipsec profile** command is run. After all the SAs are cleared, the CPU usage restores to the normal range.

Restrictions on the Use of IPsec

- The security protocol, authentication algorithm, encryption algorithm, and packet encapsulation mode on both tunnel endpoints must be the same when you configure a security proposal. Otherwise, tunnel negotiation will fail. If the PFS algorithm is configured, ensure that the two ends use the same PFS algorithm. Otherwise, tunnel negotiation will fail.
- In L2TP over IPsec scenarios, the function that the responder accepts the security proposal of the initiator is usually used together with L2TP. Separate use of this function will reduce network security, and is therefore not preferred.
- To reference an ACL in an IPsec policy, ensure that rules must be configured in this ACL view and the number of rules configured in this ACL view does not exceed 256. Otherwise, this ACL cannot be referenced in this IPsec policy.
- When configuring IPsec to-be-encrypted data flows, configure refined ACL rules based on services to prevent unnecessary data flows from entering the encryption tunnel due to loose ACL rules, causing service interruption.
- Dynamically modifying ACL configuration is not recommended. This configuration mode may lead to tunnel negotiation failures. To modify ACL configuration, you are advised to cancel the IPsec security policy group specified for the ACL on the corresponding interface, modify the ACL configuration, and re-apply the IPsec security policy group to the interface.
- Setting the MTU to a value smaller than 256 bytes is not recommended for the interface to which an IPsec security policy group applies. As IP packets become longer after IPsec processing, a small MTU makes the interface divide a large IP packet into multiple fragments. The peer device may not properly receive or process such fragmented packets.
- When a NAT device is deployed between IPsec peers, NAT traversal must be enabled and the security protocol must be ESP.
- In AH encapsulation mode, the DF flag bit of the inner packet is inherited to the outer packet, and the Router combines it with the DF flag bit of the outer layer to calculate the checksum of the packet. If the peer end of the tunnel removes the DF flag bit from the outer packet and then calculates the checksum, the checksum on both ends of the tunnel

is inconsistent. As a result, the interconnection fails. To prevent this, run the **ipsec df-bit clear** command to ensure that the checksum on both ends of the tunnel is consistent.

- When the IPsec protocol on both the AR and its connected other device uses the SHA-2 algorithm, an IPsec tunnel can be established but traffic cannot be transmitted if the SHA-2 encryption and decryption modes on the two devices are different. If so, you are advised to run the **ipsec authentication sha2 compatible enable** command on the AR to set the SHA-2 encryption and decryption modes to be the same as those on the other device.
- It is not recommended that IPsec be deployed on both physical interfaces and tunnel interfaces. If IPsec is deployed on both physical interfaces and tunnel interfaces, the device functioning as the negotiation responder first attempts to perform tunnel negotiation through IPsec of a tunnel interface. If the device does not match IPsec access requirements of the tunnel interface, the device attempts to perform tunnel negotiation through IPsec of a physical interface.
- In transport mode, the flow information after IPsec negotiation must be consistent with the IPsec tunnel address, a 32-bit host address.

Restrictions on the Use with NAT

If NAT is configured on an interface where an IPsec policy group applies, the IPsec configuration may not take effect because the device performs NAT first.

- If the interface implements IPsec but not NAT, the action of the ACL rule referenced by NAT needs to be set to deny, and the destination IP address of the rule needs to be set to that of the ACL rule referenced by the IPsec policy.
- If the interface implements NAT but not IPsec, the destination IP address of the ACL rule referenced by the IPsec policy cannot be a NATed IP address.
- If the interface implements both NAT and IPsec, the destination IP address of the ACL rule referenced by the IPsec policy must be a NATed IP address.

6.6 Default Settings for IPsec

Table 6-6 Default settings for IPsec

Parameter	Default Setting
Local host name used in IKE negotiation	Local device name
Interval for sending NAT keepalive packets	20s
IKE proposal	An IKE proposal with the lowest priority. For details, see 6.10.1 Configuring an IKE Proposal .
IPsec proposal	no IPsec proposal is configured. For detailed parameters in the created IPsec proposal, see 6.7.2 Configuring an IPsec Proposal .
SA trigger mode	Auto
IKE SA hard lifetime	86400s

Parameter	Default Setting
Global IPsec SA hard lifetime	<ul style="list-style-type: none"> ● Time-based: 3600s ● Traffic-based: 1843200 Kbytes (1800 Mbytes)
ACL check for decrypted packets	Disabled
Global IPsec anti-replay	Enabled
Global IPsec anti-replay window size	1024
Packet fragmentation mode	Fragmentation after encryption
NAT traversal	Enabled

6.7 Using an ACL to Establish an IPsec Tunnel

Pre-configuration Tasks

On an IPsec tunnel established in manual or IKE negotiation mode, an ACL defines data flows to be protected. The packets that match the permit clauses in the ACL are protected, and the packets that match the deny clauses are not protected. The ACL can define packet attributes such as the IP address, port number, and protocol type, which help you flexibly define IPsec policies.

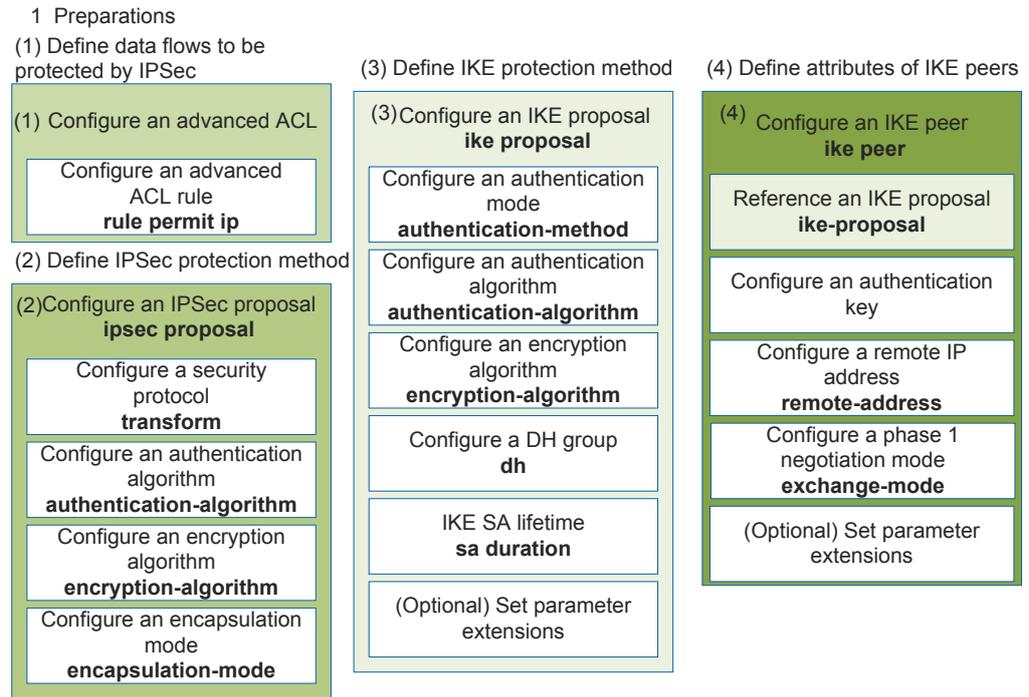
Before establishing an IPsec tunnel using an ACL, complete the following tasks:

- Configure a reachable route between source and destination interfaces.
- (Optional) If L2TP over IPsec needs to be configured, perform the following configurations:
 - Configure LAC** on the branch gateway. If a client on the branch network dials to connect to the headquarters network through the LAC, configure NAS-initiated VPN LAC. If the LAC connects to the headquarters network through automatic dial-up, configure LAC auto-dial.
 - Configure LNS** on the headquarters gateway.
- (Optional) If ACL-based GRE over IPsec needs to be configured, perform the following configurations:
 - Create a tunnel interface and set the type of the interface to GRE.
 - Configure source and destination IP addresses, and interface IP addresses. The source IP address is the IP address of the outbound interface on the gateway, and the destination IP address is the IP address of the outbound interface on the remote gateway.
 - Add tunnel interfaces to a zone.

Configuration Process

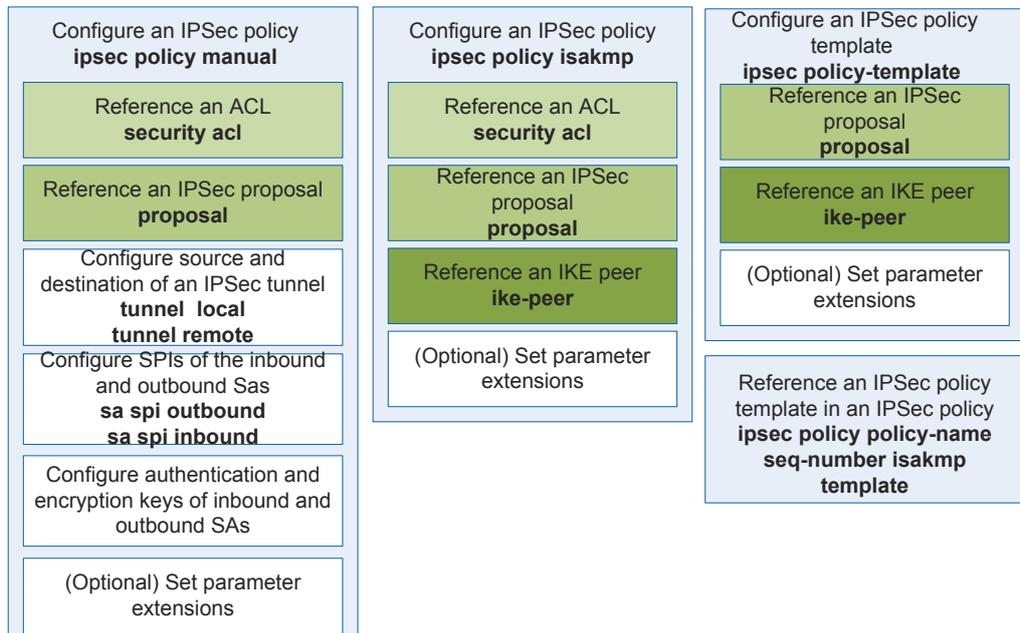
Figure 6-28 shows the configuration process (IKEv1 is used).

Figure 6-28 Using an ACL to establish an IPsec tunnel

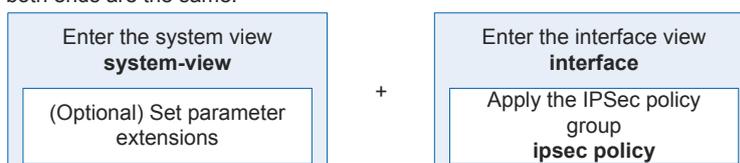


2 Configure an IPsec policy. Use a specified protection method for specified data flows.

- Manually created IPsec policy
- IPsec policy established in IKE negotiation mode
- IPsec policy established using an IPsec policy template



3 Apply an IPsec policy group to an interface. An IPsec tunnel is established when IPsec parameters at both ends are the same.



6.7.1 Defining Data Flows to Be Protected

Context

IPsec can protect one or more data flows, and the ACL specifies data flows to be protected by IPsec. Therefore, you need to create an ACL and apply the ACL to an IPsec policy. An IPsec policy can reference only one ACL. Note the following points:

- If data flows have different security requirements, create different ACLs and IPsec policies.
- If data flows have the same security requirements, configure multiple rules in an ACL.

ACL Keyword Usage

Each ACL rule is a deny or permit clause. In IPsec applications, a permit clause identifies a data flow protected by IPsec, and a deny clause identifies a data flow that is not protected by IPsec. An ACL can contain multiple rules. A packet is processed according to the first rule that it matches.

- In the outbound direction of an SA
If a packet matches a permit clause, IPsec encapsulates and sends the packet. If a packet matches a deny clause or does not match a permit clause, IPsec directly forwards the packet. A matched permit clause indicates that a data flow needs to be protected and a pair of SAs is created.
- In the inbound direction of an SA
The packet protected by IPsec is decrypted and the packet not protected by IPsec is forwarded.

NOTE

If [6.7.8 \(Optional\) Configuring IPsec Check](#) is performed, the device re-checks whether the IP header of the decrypted IPsec packet is in the range defined by the ACL. If the decrypted IPsec packet matches the permit clause, the device continues to process the IPsec packet. If the decrypted IPsec packet does not match the permit clause, the device discards the IPsec packet.

Precautions

- The protocols defined in the ACLs on both ends of the IPsec tunnel must be the same. For example, if the protocol on one end is IP, the protocol must also be IP on the other end.
- When ACL rules at both ends of an IPsec tunnel mirror each other, SAs can be set up successfully no matter which party initiates negotiation. If ACL rules at both ends of an IPsec tunnel do not mirror each other, SAs can be set up successfully only when the range specified by ACL rules on the initiator is included in the range specified by ACL rules on the responder. It is recommended that ACL rules at both ends of an IPsec tunnel mirror each other. That is, the source and destination addresses of an ACL rule at one end are the destination and source addresses of an ACL rule at the other end. The IKEv1 and IKEv2 configurations are as follows:

If IPsec policies in ISAKMP mode are configured at both ends, ACL rules at both ends of an IPsec tunnel must mirror each other. If an IPsec policy in ISAKMP mode is configured at one end and an IPsec policy using an IPsec policy template is configured at the other end, the range specified by ACL rules in the IPsec policy in ISAKMP mode can be included in the range specified by ACL rules in the IPsec policy using an IPsec policy template. The devices use overlapping ACL rules as the negotiation result.

- Avoid overlapped address segments in ACL rules. Rules with overlapped address segments may affect each other, causing data flow mismatch.
- The ACL referenced in an IPsec policy group cannot contain rules of the same ID.
- ACL rules referenced in all IPsec policies of an IPsec policy group cannot overlap. In the following example, ACL 3001 and ACL 3002 overlap.

```
acl number 3001
 rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
acl number 3002
 rule 5 permit ip source 10.1.0.0 0.0.255.255 destination 10.1.0.0 0.0.255.255
```

- When the responder uses an IPsec policy template, note the following points:
If data flows to be protected are not specified, the responder accepts the range of data flows to be protected on the initiator. If data flows to be protected are specified, the ACL on the responder must mirror the ACL on the initiator or the range specified by the ACL on the responder must cover the range specified by the ACL on the initiator.
After an IPsec tunnel has been established, if both permit and deny actions are configured in an ACL rule in the IPsec policy template view, the deny action does not take effect.
- If NAT is configured on an interface to which an IPsec policy is applied, IPsec may not take effect because NAT is performed first. You can use the following methods:
 - Configure the destination IP address that matches the deny clause in an ACL referenced by NAT as the destination IP address in an ACL rule referenced by IPsec. In this case, data flows protected by IPsec are not translated by NAT.
 - Configure the ACL rule referenced by NAT to match the IP address translated by NAT.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Run `acl [number] acl-number [match-order { config | auto }]`

An advanced ACL is created and the advanced ACL view is displayed. *acl-number* ranges from 3000 to 3999.

Step 3 Run the following commands as required.

- Run the `rule [rule-id] { deny | permit } ip [destination { destination-address destination-wildcard | any } | source { source-address source-wildcard | any } | dscp dscp] *` command to configure a rule to match the IP protocol.
- Run the `rule [rule-id] { deny | permit } tcp [destination { destination-address destination-wildcard | any } | destination-port eq port | source { source-address source-wildcard | any } | source-port eq port | dscp dscp] *` command to configure a rule to match the TCP protocol.
- Run the `rule [rule-id] { deny | permit } udp [destination { destination-address destination-wildcard | any } | destination-port eq port | source { source-address source-wildcard | any } | source-port eq port | dscp dscp] *` command to configure a rule to match the UDP protocol.
- Run the `rule [rule-id] { deny | permit } gre [destination { destination-address destination-wildcard | any } | source { source-address source-wildcard | any } | dscp`

dscp | **precedence** *precedence* | **tos** *tos* | **time-range** *time-name* | **logging**] * command to configure a rule to match the GRE protocol.

- Run the **rule** [*rule-id*] { **deny** | **permit** } **gre** [**destination** { *destination-address* | *destination-wildcard* | **any** } | **source** { *source-address* | *source-wildcard* | **any** }] [**dscp** *dscp* | [**tos** *tos* | **precedence** *precedence*] *] | **time-range** *time-name* | **logging**] * command to configure a rule to match the GRE protocol.

----End

Configuration Guidelines

The configurations of rules vary in different scenarios. For details, see the following examples:

Site-to-Site IPsec VPN

A site-to-site IPsec tunnel is set up between gateway A and gateway B. Gateway A protects subnet 10.1.1.0/24 and gateway B protects subnet 192.168.196.0/24.

Configurations on gateway A:

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
192.168.196.0 0.0.0.255
```

Configurations on gateway B:

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
```

Ensure that the subnets on both ends use the same wildcard.

Hub-Spoke IPsec VPN

Hub-Spoke IPsec tunnels are set up between the headquarters and branches. The headquarters resides at subnet 192.168.196.0/24; branch A resides at subnet 10.1.1.0/24; branch B resides at subnet 10.1.2.0/24.

- To allow the communication between branches and the headquarters but forbid the communication between branches, configure the ACL for the branch network in the same way as in the site-to-site IPsec VPN. Note that the destination address of the ACL at the headquarters must include all branch subnets.

The ACL at the headquarters is configured as follows:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255
destination 10.1.1.0 0.0.0.255
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255
destination 10.1.2.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

- To allow the communication between branches and the headquarters, and between branches through the headquarters, set the source address of the ACL at the headquarters to all subnets of the headquarters and branches. Set the destination address to all branch subnets. The source addresses of the ACLs at the branch offices remain, but the destination addresses must be the subnets of the headquarters and all other branches.

The ACL at the headquarters is configured as follows:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255
destination 10.1.1.0 0.0.0.255
```

```
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255
destination 10.1.2.0 0.0.0.255
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[Huawei-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

The ACL at branch A is configured as follows:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
192.168.196.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

The ACL at branch B is configured as follows:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Huawei-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255 destination
192.168.196.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

IPsec Gateway with NAT Configured

- If endpoint A uses NAT only for the Internet access, not for IPsec traffic, you must reject the IPsec traffic from NAT.

Endpoint A protects network 10.1.1.0/24 and endpoint B protects network 192.168.196.0/24. The ACL and NAT configurations on endpoint A are as follows:

Define the data flow to be protected.

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
192.168.196.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

Exclude the networks connected by the IPsec tunnel from the ACL referenced in the NAT policy.

```
[Huawei] acl 3005
[Huawei-acl-adv-3005] rule deny ip source 10.1.1.0 0.0.0.255 destination
192.168.196.0 0.0.0.255
[Huawei-acl-adv-3005] quit
```

Configurations on gateway B:

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 192.168.196.0 0.0.0.255
destination 10.1.1.0 0.0.0.255
```

- If the two networks overlap, endpoint A performs NAT for all traffic and then performs IPsec.

If the networks protected by endpoints A and B are both network 10.1.1.0/24, the private addresses are translated to 10.1.2.1, the configurations on endpoints A and B are as follows:

On endpoint A:

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

On endpoint B:

```
[Huawei] acl 3001
[Huawei-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[Huawei-acl-adv-3001] quit
```

L2TP over IPsec

In a scenario where L2TP over IPsec is deployed, IPsec protects data flows that are encapsulated through L2TP, that is, data flows sent from the LAC to the LNS or from the LNS to the LAC.

- If the LAC uses a fixed IP address, source and destination network segments in ACL rules are addresses of public interfaces (LAC-side outbound interface and LNS-side inbound interface) on devices at both ends. Assume that the IP address of the LAC-side outbound interface is 1.1.1.1/24 and the IP address of the LNS-side inbound interface is 1.2.1.1/24.

Configuration on the LAC:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 1.1.1.1 0 destination 1.2.1.1 0
[Huawei-acl-adv-3001] quit
```

Configuration on the LNS:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 1.2.1.1 0 destination 1.1.1.1 0
[Huawei-acl-adv-3001] quit
```

- If the LAC does not use a fixed IP address, specify UDP port 1701 in an ACL to match L2TP over IPsec data flows. On the LAC, configure destination UDP port 1701.

Configuration on the LAC:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit udp destination-port eq 1701
[Huawei-acl-adv-3001] quit
```

Configuration on the LNS:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit udp source-port eq 1701
[Huawei-acl-adv-3001] quit
```

GRE over IPsec

When a GRE over IPsec tunnel is set up using an ACL, data flows protected by IPsec are encapsulated with the GRE header. The source and destination network segments of an ACL are source and destination addresses of the GRE tunnel, that is, addresses of gateway interfaces at both ends.

Assume that the public addresses on endpoints A and B are 1.1.1.1/24 and 1.2.1.1/24, respectively.

Configuration on endpoint A:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 1.1.1.1 0 destination 1.2.1.1 0
[Huawei-acl-adv-3001] quit
```

Configuration on endpoint B:

```
[Huawei] acl number 3001
[Huawei-acl-adv-3001] rule permit ip source 1.2.1.1 0 destination 1.1.1.1 0
[Huawei-acl-adv-3001] quit
```

6.7.2 Configuring an IPsec Proposal

Context

An IPsec proposal, as part of an IPsec policy or an IPsec profile, defines security parameters for IPsec SA negotiation, including the security protocol, encryption and authentication

algorithms, and encapsulation mode. Both ends of an IPsec tunnel must be configured with the same parameters.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec proposal** *proposal-name*

An IPsec proposal is created and the IPsec proposal view is displayed.

Step 3 Run **transform** { **ah** | **esp** | **ah-esp** }

A security protocol is configured.

By default, an IPsec proposal uses ESP.

Step 4 An authentication or encryption algorithm is configured.

- If AH is used, you can only configure the AH-specific authentication algorithm because AH only authenticates packets.

Run **ah authentication-algorithm** { **md5** | **sha1** | **sha2-256** | **sha2-384** | **sha2-512** }

An AH-specific authentication algorithm is configured.

By default, AH uses the SHA2-256 authentication algorithm.

- When ESP is specified, ESP can encrypt/authenticate, or encrypt and authenticate packets. Configure the ESP-specific authentication or encryption algorithm.

- Run **esp authentication-algorithm** { **md5** | **sha1** | **sha2-256** | **sha2-384** | **sha2-512** }

An ESP-specific authentication algorithm is configured.

By default, ESP uses the SHA2-256 authentication algorithm.

- Run **esp encryption-algorithm** { **3des** | **des** | **aes-128** | **aes-192** | **aes-256** }

An ESP-specific encryption algorithm is configured.

By default, ESP uses the AES-256 encryption algorithm.

- When both AH and ESP are used, AH authenticates packets, and ESP can encrypt and authenticate packets. You can choose to configure an AH-specific authentication algorithm, or ESP-specific authentication and encryption algorithms. The device first encapsulates the ESP header, and then the AH header to packets.

NOTE

- Authentication algorithms SHA2-256, SHA2-384, and SHA2-512 are recommended to improve packet transmission security, whereas authentication algorithms MD5 and SHA1 are not recommended.
- Encryption algorithms AES-128, AES-192, and AES-256 are recommended to improve packet transmission security, whereas encryption algorithm DES and 3DES are not recommended.

Step 5 Run **encapsulation-mode** { **transport** | **tunnel** }

An IP packet encapsulation mode is configured.

By default, IPsec uses the tunnel mode to encapsulate IP packets.

When IKEv2 is used, the encapsulation modes in all the IPsec proposals configured on the IKE initiator must be the same; otherwise, IKE negotiation fails.

 **NOTE**

When L2TP over IPsec or GRE over IPsec is configured, a public IP header is added to packets during L2TP or GRE encapsulation. Compared with the transport mode, the tunnel mode adds another public IP header. In tunnel mode, the packet length is longer and packets are more likely to be fragmented. The transport mode is therefore recommended.

Step 6 Run quit

Exit the IPsec proposal view.

Step 7 (Optional) Run ipsec authentication sha2 compatible enable

The SHA-2 algorithm is compatible with earlier software versions.

By default, the SHA-2 algorithm is not compatible with earlier software versions.

When IPsec uses the SHA-2 algorithm, if the devices on two ends of an IPsec tunnel are from different vendors or run different software versions, they may use different encryption and decryption methods. In this situation, traffic between devices is interrupted.

To solve this problem, enable SHA-2 to be compatible with earlier versions.

---End

6.7.3 Configuring an IPsec Policy

Context

An IPsec policy defines the IPsec proposals used to protect data flows of different types, and is the prerequisite for creating an SA. An IPsec policy binds an ACL to an IPsec proposal, and specifies the SA negotiation mode, source and destination of the IPsec tunnel, key, and SA lifetime.

An IPsec policy is identified by its name and sequence number, and multiple IPsec policies with the same IPsec policy name constitute an IPsec policy group. An IPsec policy can be established manually, in ISAKMP mode, or using an IPsec policy template. For IPsec policies that are established in ISAKMP mode and using an IPsec policy template, parameters are generated through IKE negotiation.

Select an IPsec policy establishment mode as needed:

 **NOTE**

- When a GRE over IPsec tunnel is established using an ACL, an IPsec policy in ISAKMP mode can only be configured on gateways at both ends.
- When an L2TP over IPsec tunnel is established using an ACL and the LAC is used as the initiator, an IPsec policy in ISAKMP mode can only be configured on the LAC. When the LNS functions as the responder, an IPsec policy in ISAKMP mode or using an IPsec policy template can be configured on the LNS. In the Hub-Spoke VPN, an IPsec policy using an IPsec policy template is recommended.

6.7.3.1 Configuring an IPsec Policy in Manual Mode

Context

All security parameters of an IPsec policy configuring in manual mode need to be configured manually. The configuration workload is heavy, so the IPsec policy applies to a small-scale network environment.

When configuring an IPsec policy in manual mode, ensure that:

- Inbound and outbound SAs' parameters, including the authentication/encryption key and security parameter index (SPI), are configured on IPsec peers.
- The inbound SA's parameters on the local end is the same as the outbound SA's parameters on the remote end, and the outbound SA's parameters on the local end is the same as the inbound SA's parameters on the remote end.

After an IPsec policy group is applied to an interface, to add or delete an IPsec policy in the IPsec policy group or modify parameters of the IPsec policy, unbind the IPsec policy group from the interface and then apply the IPsec policy group to the interface again so that IPsec policies in the IPsec policy group take effect.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy *policy-name seq-number manual***

An IPsec policy is created in manual mode and the IPsec policy view is displayed.

By default, no IPsec policy is created.

Step 3 Run **security acl *acl-number***

An ACL is referenced in the IPsec policy.

By default, an IPsec policy does not reference an ACL.

acl-number is an advanced ACL that has been created.

An IPsec policy can reference only one ACL. Before referencing a new ACL, you must delete the original ACL that has been referenced.

Step 4 Run **proposal *proposal-name***

An IPsec proposal is referenced in the IPsec policy.

By default, an IPsec policy does not reference an IPsec proposal.

proposal-name is an IPsec proposal that has been created.

One IPsec policy can reference only one IPsec proposal. Before referencing a new IPsec proposal, you must delete the original IPsec proposal that has been referenced.

Step 5 Configure the local and remote IP addresses of an IPsec tunnel.

1. Run **tunnel local *ip-address***

A local IP address is configured.

2. Run **tunnel remote *ip-address***

A remote IP address is configured.

By default, the local and remote IP addresses of an IPsec tunnel are not configured.

The remote IP address at the local end must be the same as the local IP address at the remote end.

Step 6 Configure the SPI for the inbound or outbound SA.

1. Run **sa spi outbound { ah | esp } spi-number**
An SPI is configured for the outbound SA.
2. Run **sa spi inbound { ah | esp } spi-number**
An SPI is configured for the inbound SA.

The security protocol must be the same as that specified in the **transform** command in [6.7.2 Configuring an IPsec Proposal](#). If the security protocol specified in the **transform** command is **ah-esp**, both **ah** and **esp** must be specified in the **sa spi** command.

To retain a unique SA, SPIs for inbound and outbound SAs must be different.

Step 7 Configure authentication and encryption keys for the inbound or outbound SA.

NOTE

- The security protocol specified in authentication and encryption key configuration commands must be the same as that specified in the **transform** command in [6.7.2 Configuring an IPsec Proposal](#). If the security protocol specified in the **transform** command is **ah-esp**, both **ah** and **esp** authentication and encryption keys must be specified.
- The two ends of an IPsec tunnel must use the authentication keys in the same format. For example, if the key on one end is a character string but the key on the other end is a hexadecimal number, the IPsec tunnel cannot be established.
- If **simple** is specified, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you specify **cipher** to save the password in cipher text.
- If the inbound authentication keys in a character string and hexadecimal notation are configured, the one configured later overwrites the original one.

If AH is used, configure an authentication key.

- Run **sa string-key { inbound | outbound } ah { simple | cipher } string-key**
An authentication key in a character string is configured for AH.
- Run **sa authentication-hex { inbound | outbound } ah { simple | cipher } hex-string**
An authentication key in hexadecimal notation is configured for AH.

If ESP is used, configure an authentication key.

- Run **sa string-key { inbound | outbound } esp { simple | cipher } string-key**
An authentication key in a character string is configured for ESP.

NOTE

When ESP is used and the authentication key in a character string is used, the device automatically generates the encryption key of ESP. You do not need to configure the encryption key of ESP.

If ESP is used, configure authentication and encryption keys.

1. (Optional) Run **sa authentication-hex { inbound | outbound } esp { simple | cipher } hex-string**
An authentication key in hexadecimal notation is configured for ESP.
2. (Optional) Run **sa encryption-hex { inbound | outbound } esp { simple | cipher } hex-string**
An encryption key in hexadecimal notation is configured for ESP.

You must run at least one of the preceding commands.

----End

6.7.3.2 Configuring an IPsec Policy in ISAKMP Mode

Context

An IPsec policy configured in Internet Security Association and Key Management Protocol (ISAKMP) mode applies to a scenario where the remote IP address is fixed, and is often used in branch configuration.

Negotiated IPsec parameters of an IPsec policy are defined in the IPsec policy view, and the negotiation initiator and responder must use the same IPsec parameters. The end that has an ISAKMP IPsec policy configured can initiate IKE negotiation.

After an IPsec policy group to which an IPsec policy belongs is applied to an interface, the following situations occur:

- To modify the IPsec proposal parameters, unbind the IPsec policy group from the interface and then apply the IPsec policy group to the interface again.
- If other parameters are modified, these parameters will take effect during the next negotiation and are invalid for the tunnels that have been established through negotiation.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy *policy-name seq-number isakmp***

An IPsec policy is created in ISAKMP mode and the IPsec policy view is displayed.

By default, no IPsec policy is created.

Step 3 (Optional) Run **alias *alias***

The alias of the IPsec policy is specified.

By default, the system uses the combination of the name and sequence number of an IPsec policy as the alias. If the default alias has been used by another IPsec policy, the system uses the combination of the name, sequence number, and current time of an IPsec policy as the alias.

Step 4 Run **security acl *acl-number* [**dynamic-source**]**

An ACL is referenced in the IPsec policy.

By default, an IPsec policy does not reference an ACL.

acl-number is an advanced ACL that has been created.

An IPsec policy can reference only one ACL. Before referencing a new ACL, you must delete the original ACL that has been referenced.

Step 5 Run **proposal *proposal-name***

An IPsec proposal is referenced in the IPsec policy.

By default, an IPsec policy does not reference an IPsec proposal.

proposal-number specifies a created IPsec proposal.

An IPsec policy configured in ISAKMP mode can reference a maximum of 12 IPsec proposals. During IKE negotiation, the two ends of an IPsec tunnel first use the IPsec proposals with the same parameter settings. If IPsec proposals with the same parameter settings cannot be found, an SA cannot be set up.

 **NOTE**

When referencing multiple IPsec proposals in an IPsec policy, ensure that the encapsulation modes of all IPsec proposals referenced by the IPsec policy at both ends are the same. That is, the encapsulation modes are all transport or tunnel modes.

Step 6 Run **ike-peer** *peer-name*

An IKE peer is referenced in the IPsec policy.

By default, an IPsec policy does not reference an IKE peer.

peer-name specifies a created IKE peer. For the detailed configuration of an IKE peer, see [6.10.2 Configuring an IKE Peer](#).

IPsec policies with different sequence numbers in the same IPsec policy group cannot reference IKE peers with the same IP address.

Step 7 (Optional) Run **tunnel local** { *ipv4-address* | **applied-interface** }

A local IP address of an IPsec tunnel is configured.

By default, the local IP address of an IPsec tunnel is not configured.

For the IKE negotiation mode, you do not need to configure an IP address for the local end of an IPsec tunnel. During SA negotiation, the device will select a proper address based on route information. The local address needs to be configured in the following situations:

- If the IP address of the interface to which an IPsec policy is applied varies or is unknown, run the **tunnel local** *ipv4-address* command to specify the IP address of another interface (such as the loopback interface) on the device as the IP address for the local end of an IPsec tunnel. Otherwise, run the **tunnel local applied-interface** command to specify the IP address of the interface to which an IPsec policy is applied as the local address of an IPsec tunnel.
- If the interface to which an IPsec policy is applied has multiple IP addresses (one primary IP address and several secondary IP addresses), run the **tunnel local** *ipv4-address* command to specify one of these IP addresses as the IP address for the local end of an IPsec tunnel. Otherwise, run the **tunnel local applied-interface** command to specify the primary IP address of the interface as the local address of an IPsec tunnel.
- You do not need to specify the **tunnel local** (local address) for the IKE peer referenced in an IPsec profile, because the local address is the source address of the GRE, mGRE or IPsec virtual tunnel interface. For the IKE peer referenced in an IPsec profile, **tunnel local** do not take effect.
- When applying an IPsec policy to a tunnel interface and running the **source** command to specify an IP address for the interface, you must run the **tunnel local** command to configure a tunnel local address. Otherwise, IKE negotiation will fail.
- If equal-cost routes exist between the local and remote ends, run the **tunnel local** command to specify a local IP address for an IPsec tunnel.

 **NOTE**

- If an IPsec policy is created in IKE negotiation mode, the **tunnel local** on the local end must be the same as **remote-address (IKE peer view)** that the remote end references from the IKE peer.
- In an IPsec hot standby scenario, **tunnel local** must be set to a virtual IP address.

Step 8 (Optional) Run **sa trigger-mode { auto | traffic-based }**

An IPsec tunnel trigger mode is configured.

By default, the IPsec tunnel trigger mode is **auto**.

Step 9 (Optional) Run **pfs { dh-group1 | dh-group2 | dh-group5 | dh-group14 | dh-group19 | dh-group20 | dh-group21 }**

The device is configured to use perfect forward secrecy (PFS) when the local end initiates negotiation.

By default, PFS is not used when the local end initiates negotiation.

When the local end initiates negotiation, there is an additional Diffie-Hellman (DH) exchange in IKEv1 phase 2 or IKEv2 CREATE_CHILD_SA exchange. The additional DH exchange ensures security of the IPsec SA key and improves communication security.

If PFS is specified on the local end, you also need to specify PFS on the remote end. The DH group specified on the two ends must be the same; otherwise, negotiation fails. When an IPsec policy in ISAKMP mode is used on the local end while an IPsec policy configured using an IPsec policy template is used on the remote end, no DH group needs to be configured on the remote end. The DH group on the responder is used for negotiation.

Step 10 (Optional) Run **respond-only enable**

The local end is configured not to initiate negotiation.

By default, if the local end establishes an IPsec tunnel based on the IPsec policy configured in ISAKMP mode, the local end initiates an IPsec negotiation.

If two IPsec peers establish an IPsec tunnel based on the IPsec policy configured in ISAKMP mode, both ends initiate negotiation. You can configure one end as the responder that does not initiate negotiation, which can help you check packet processing and locate IPsec faults.

Step 11 (Optional) Run **policy enable**

The IPsec policy is enabled.

By default, IPsec policies in an IPsec policy group are enabled.

----End

6.7.3.3 Configuring an IPsec Policy Using an IPsec Policy Template

Context

When an IPsec policy template is used to configure IPsec policies, the configuration workload for establishing multiple IPsec tunnels can be reduced. This IPsec policy configuration mode is often used in the headquarters in scenarios where the remote IP address is not fixed (for example, the remote end obtains an IP address through PPPoE) or there are multiple remote devices.

When an IPsec tunnel is set up using an IPsec policy through an IPsec policy template, the initiator determines optional parameters, and the responder accepts the parameters delivered by the initiator. The end that has an IPsec policy configured using an IPsec policy template can only function as the responder to receive negotiation requests.

When using an IPsec policy template to configure an IPsec policy, note the following points:

- If one end (responder) of an IPsec tunnel has an IPsec policy configured using an IPsec policy template, the other end (initiator) must have an IPsec policy configured in ISAKMP mode.
- In an IPsec policy template, an IPsec proposal and IKE peer must be referenced, and other parameters are optional. The initiator determines optional parameters in the IPsec policy template, and the responder accepts the parameters delivered by the initiator.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy-template** *template-name seq-number*

An IPsec policy template is created and the IPsec policy template view is displayed.

By default, no IPsec policy template is created.

Step 3 (Optional) Run **alias** *alias*

The alias name of the IPsec policy template is specified.

By default, the system uses the combination of the name and sequence number of an IPsec policy template as the alias. If the default alias has been used by another IPsec policy template, the system uses the combination of the current time as well as the name and sequence number of an IPsec policy template as the alias.

Step 4 (Optional) Run **security acl** *acl-number*

An ACL is referenced in the IPsec policy.

By default, an IPsec policy does not reference an ACL.

acl-number is an advanced ACL that has been created.

One IPsec policy template can reference only one ACL. Before referencing a new ACL, you must delete the ACL that has been referenced.

If data flows to be protected are not specified, the responder accepts the range of data flows to be protected on the initiator. If data flows to be protected are specified, the ACL on the responder must mirror the ACL on the initiator or the range specified by the ACL on the responder must cover the range specified by the ACL on the initiator.

Step 5 Run **proposal** *proposal-name*

An IPsec proposal is referenced in the IPsec policy template.

By default, an IPsec policy template does not reference an IPsec proposal.

proposal-name is an IPsec proposal that has been created.

An IPsec policy template can reference a maximum of 12 IPsec proposals. During IKE negotiation, the two ends of an IPsec tunnel first use the IPsec proposals with the same

parameter settings. If IPsec proposals with the same parameter settings cannot be found, an SA cannot be set up.

 **NOTE**

When referencing multiple IPsec proposals in an IPsec policy template, ensure that the encapsulation mode of IPsec proposals referenced by the IPsec policy template at one end are the same as the encapsulation mode of IPsec proposals referenced by the IPsec policy at the other end. That is, the encapsulation mode at both ends must be transport or tunnel.

Step 6 Run **ike-peer** *peer-name*

An IKE peer is referenced in the IPsec policy template.

By default, an IPsec policy template does not reference an IKE peer.

peer-name is an IKE peer that has been created.

Step 7 (Optional) Run **tunnel local** *ipv4-address*

A local IP address of an IPsec tunnel is configured.

By default, the local IP address of an IPsec tunnel is not configured.

 **NOTE**

- If an IPsec policy is created in IKE negotiation mode, the **tunnel local** on the local end must be the same as **remote-address (IKE peer view)** that the remote end references from the IKE peer.
- In an IPsec hot standby scenario, **tunnel local** must be set to a virtual IP address.

Step 8 (Optional) Run **match ike-identity** *identity-name*

The identity filter set is referenced.

By default, an IPsec policy template does not reference an identity filter set.

identity-name is an identity filter that has been created. For details on how to configure an identity filter set, see [6.10.5 \(Optional\) Configuring an Identity Filter Set](#).

 **NOTE**

When an IPsec policy template references the **identity filter set**, the allowed IKE peer can be specified at the local end. An IPsec tunnel can be established successfully only when the remote end matches one or more access conditions in the identity filter set and IPsec parameters at both ends match.

Step 9 (Optional) Run **pfs { dh-group1 | dh-group2 | dh-group5 | dh-group14 | dh-group19 | dh-group20 | dh-group21 }**

The device is configured to use perfect forward secrecy (PFS) when the local end initiates negotiation.

By default, PFS is not used when the local end initiates negotiation.

When the local end initiates negotiation, there is an additional Diffie-Hellman (DH) exchange in IKEv1 phase 2 or IKEv2 CREATE_CHILD_SA exchange. The additional DH exchange ensures security of the IPsec SA key and improves communication security.

If PFS is specified on the local end, you also need to specify PFS on the remote end. The DH group specified on the two ends must be the same; otherwise, negotiation fails. When an IPsec policy in ISAKMP mode is used on the local end while an IPsec policy configured using an IPsec policy template is used on the remote end, no DH group needs to be configured on the remote end. The DH group on the responder is used for negotiation.

Step 10 (Optional) Run **policy enable**

The IPsec policy is enabled.

By default, IPsec policies in an IPsec policy group are enabled.

Step 11 Run **quit**

Return to the system view.

Step 12 Run **ipsec policy *policy-name seq-number isakmp template *template-name****

An IPsec policy template is referenced in the IPsec policy.

The referenced IPsec policy template name *template-name* must be different from the IPsec policy name *policy-name*.

Only one IPsec policy in an IPsec policy group can reference the policy template, and number of this policy must be larger than that of other policies. If the IPsec policy created using the policy template does not have the lowest priority, other IPsec policies in the same IPsec policy group do not take effect.

----End

6.7.4 (Optional) Setting the IPsec SA Lifetime

Context

NOTE

- The IPsec SA lifetime is only valid for the IPsec SAs established in IKE negotiation mode. Manually established IPsec SAs are always valid.
- The configured IPsec SA lifetime is only valid for the new IPsec SAs established in IKE negotiation mode.

For a dynamic SA, configure the SA hard lifetime so that the SA can be updated in real time, reducing the crash risk and improving security.

There are two methods to measure the lifetime:

- Time-based lifetime
The period from when an SA is set up to when the SA is expired.
- Traffic-based lifetime
The maximum volume of traffic that this SA can process.

The lifetime is classified as follows:

- Hard lifetime: specifies the lifetime of an IPsec SA.
When two devices negotiate an IPsec SA, the actual hard lifetime is the smaller of the two values configured on the two devices.
- Soft lifetime: specifies the time after which a new IPsec SA is negotiated so that the new IPsec SA will be ready before the hard lifetime of the original IPsec SA expires.

Table 6-7 lists the default soft lifetime values.

Table 6-7 Soft lifetime values

Soft Lifetime Type	Description
Time-based soft lifetime (soft timeout period)	The value is 7/10 of the actual hard lifetime (hard timeout period).
Traffic-based soft lifetime (soft timeout traffic)	The value is 7/10 of the actual hard lifetime (hard timeout traffic).

Before an IPsec SA becomes invalid, IKE negotiates a new IPsec SA for the remote end. The remote end uses the new IPsec SA to protect IPsec communication immediately after the new IPsec SA is negotiated. If service traffic is transmitted, the original IPsec SA is deleted immediately. If no service traffic is transmitted, the original IPsec SA will be deleted after 10s or the hard lifetime expires.

If the time-based lifetime and traffic-based lifetime are both set for an IPsec SA, the IPsec SA becomes invalid when either lifetime expires.

You can set the global IPsec SA lifetime or set the IPsec SA lifetime in an IPsec policy. If the IPsec SA lifetime is not set in an IPsec policy, the global lifetime is used. If both the global IPsec SA lifetime or the IPsec SA lifetime in an IPsec policy are set, the IPsec SA lifetime in the IPsec policy takes effect.

Procedure

- Set the global IPsec SA hard lifetime.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **ipsec sa global-duration { time-based *interval* | traffic-based *size* }**

The global IPsec SA hard lifetime is set.

By default, the global time-based SA hard lifetime is 3600 seconds and the global traffic-based SA hard lifetime is 1843200 Kbytes.
- Set the IPsec SA hard lifetime in an IPsec policy.
 - a. Run **system-view**

The system view is displayed.
 - b. Configure an IPsec policy in IPsec ISAKMP mode or using an IPsec policy template.
 - Run **ipsec policy *policy-name seq-number isakmp***

An IPsec policy in IPsec ISAKMP mode is created and the IPsec policy view is displayed.
 - Run **ipsec policy-template *template-name seq-number***

An IPsec policy template is created and the IPsec policy template view is displayed.
 - c. Run **sa duration { time-based *seconds* | traffic-based *kilobytes* }**

The IPsec SA hard lifetime is set in the IPsec policy.

By default, the IPsec SA hard lifetime is not set in an IPsec policy. The system uses the global IPsec SA hard lifetime.

----End

6.7.5 (Optional) Enabling the Anti-replay Function

Context

NOTE

Only SAs established in IKE negotiation mode support the anti-replay function. Manually configured SAs do not support the anti-replay function.

To ensure non-stop service forwarding, the configured IPsec anti-replay window size takes effect only for new or re-negotiated IPsec policies but not for existing ones.

Replayed packets are packets that have been processed. IPsec uses the sliding window (anti-replay window) mechanism to check replayed packets. Each AH or ESP packet has a 32-bit sequence number. In an SA, sequence numbers of packets increase. If the sequence number of a received authenticated packet is the same as that of a decapsulated packet or if the sequence number is out of the sliding window, the device considers the packet as a replayed packet.

Decapsulating replayed packets consumes many resources and makes system performance deteriorate, resulting in a Denial Of Service (DoS) attack. After the anti-replay function is enabled, the system discards replayed packets and does not encapsulate them, saving system resources.

In some situations, for example, when network congestion occurs or QoS is performed for packets, the sequence numbers of some service data packets may be different from those in common data packets. The device that has IPsec anti-replay enabled considers the packets as replayed packets and discards them. You can disable global IPsec anti-replay to prevent packets from being discarded incorrectly or adjust the IPsec anti-replay window size to meet service requirements.

The anti-replay function can be configured globally or in an IPsec policy or profile:

- **Configuring the anti-replay function globally**
The global anti-replay function is valid for all existing IPsec policies. When the same anti-replay window parameters need to be set for many IPsec policies, you do not need to run commands one by one. You only need to set global parameters. The configuration efficiency is therefore improved.
- **Configuring the anti-replay function in an IPsec policy or policy template**
The anti-replay function can be configured separately for an IPsec policy. In this case, the anti-replay function for the IPsec policy is not affected by the global configuration.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Enable the anti-replay function. Run the following commands as required.

- Enable the anti-replay function globally.
 - a. Run **ipsec anti-replay enable**
The anti-replay function is enabled globally.

- b. Run **ipsec anti-replay window** *window-size*
The global IPsec anti-replay window size is configured.
By default, the global IPsec anti-replay window size is 1024 bits.
- Enable the anti-replay function in an IPsec policy.
 - a. Run **ipsec policy** *policy-name seq-number* [**isakmp** | **manual**]
An IPsec policy is created and the IPsec policy view is displayed.
 - b. Run **anti-replay window** *window-size*
The IPsec anti-replay window size is configured in the IPsec policy.
By default, the anti-replay window size of a single IPsec tunnel is not set. The global value is used.
- Enable the anti-replay function in an IPsec policy template.
 - a. Run **ipsec policy-template** *template-name seq-number*
An IPsec policy template is created and the IPsec policy template view is displayed.
 - b. Run **anti-replay window** *window-size*
The IPsec anti-replay window size is configured in the IPsec policy template.
By default, the anti-replay window size of a single IPsec tunnel is not set. The global value is used.

----End

6.7.6 (Optional) Configuring IPsec Fragmentation Before Encryption

Context

The length of IPsec-encapsulated packets may exceed the maximum transmission unit (MTU) of the outbound interface on the local device. If the IPsec remote device does not support fragmentation and reassembly, it cannot decapsulate packets and will discard or incorrectly process packets, affecting packet transmission.

To prevent this problem, configure IPsec fragmentation before encryption on the local device. Subsequently, the local device calculates the length of encapsulated packets. If the length exceeds the MTU, the device fragments the packets and then encapsulates each fragment. After packets reach the IPsec remote device, the remote device can decapsulate the fragments without having to reassemble them. The decapsulated packets will be forwarded normally.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec fragmentation before-encryption**

The fragmentation mode of packets is set to fragmentation before encryption for all IPsec tunnels.

By default, the packet fragmentation mode for all IPsec tunnels is fragmentation after encryption.

The DF flag in IPsec packets determines whether IPsec packets can be fragmented. If DF flag settings disable fragmentation when the fragmentation mode is used, run the **ipsec df-bit { clear | set | copy }** command in the system view to enable fragmentation on IPsec packets.

For the established IPsec tunnels, you need to restart them after running this command. Otherwise, the command function does not take effect.

----End

6.7.7 (Optional) Configuring Route Injection

Context

NOTE

Only SAs established in IKE negotiation mode support the route injection function. Manually configured SAs do not support the route injection function.

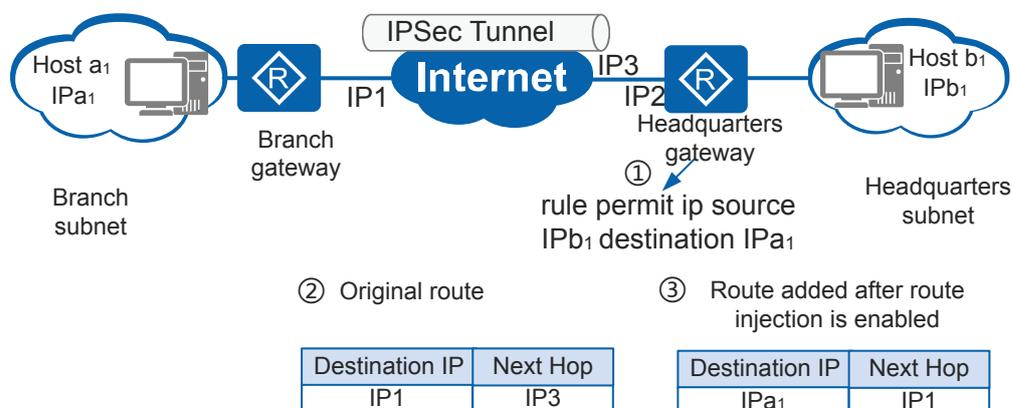
The device does not support route injection function when the IPsec policy group is bound to a Layer 2 interface.

When an enterprise headquarters and its branch establish an IPsec tunnel, a static route to the branch subnet needs to be configured on the headquarters gateway. If there are many branch subnets, a large number of static routes need to be configured on the headquarters gateway. When branch subnets change, the static route configuration needs to be modified on the headquarters gateway, causing a difficulty in network maintenance. Route injection injects routes to branch subnets to the headquarters gateway based on IPsec tunnel information, which reduces manual configuration and improves configuration correctness. If a static route from the branch to the headquarters gateway does not need to be configured manually, configure route injection.

Route injection allows the device to generate routes based on destination addresses in ACL rules referenced in IPsec policies. The next hop IP address in a route is the remote IP address learned by the local device during IPsec SA negotiation.

As shown in **Figure 6-29**, an enterprise headquarters and its branch establish an IPsec tunnel. Host a_1 represents the branch subnet, and host b_1 represents the headquarters subnet. An ACL is configured on the headquarters gateway, and defines data flows that are sent from b_1 to a_1 and are protected by IPsec. If route injection is not enabled, there must be a reachable route from the headquarters gateway to the branch subnet. After route injection is enabled on the headquarters gateway, the headquarters gateway generates a route. The route contains the destination IP address that is the same as that in an ACL rule and the next hop IP address as the branch gateway IP address. If the ACL rule does not define the destination IP address, the route uses the destination IP address 0.0.0.0/0.0.0.0.

Figure 6-29 Route injection



Route injection works in two modes:

- Static mode: The generated route is added to the local device immediately, and is independent of IPsec tunnel status change.
- Dynamic mode: If the IPsec tunnel is Up, the generated route can be added to the local device. If the IPsec tunnel is Down, the generated route can be deleted from the local device.

Compared with static route injection, dynamic route injection is relevant to the IPsec tunnel status. Dynamic route injection prevents IPsec peers from sending IPsec packets over the IPsec tunnel in Down state, reducing packet loss.

You can configure a priority for the route generated through route injection. For example, when there is another route to the same destination as the route, specify the same priority for the routes so that traffic can be load balanced. If different priorities are specified for the routes, the routes can back up each other.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 An IPsec policy in IKE negotiation mode or an IPsec policy template is configured.

- Run **ipsec policy policy-name seq-number isakmp**
 An IPsec policy is created in IKE negotiation mode and the IPsec policy view is displayed.
- Run **ipsec policy-template template-name seq-number**
 An IPsec policy template is created and the IPsec policy template view is displayed.

Step 3 Run route inject [nexthop ipv4-address] { static | dynamic } [preference preference]

Route injection is enabled.

By default, route injection is disabled.

 **NOTE**

static is only available in the ISAKMP IPsec policy view.

After the next hop is specified using the **route inject nexthop** command, the generated route is not used for IPsec packet forwarding if the IPsec tunnel remote address is not within the destination network segment of the injected route.

----End

6.7.8 (Optional) Configuring IPsec Check

Context

IPsec check ensures that data flows are correctly encrypted. After IPsec check is enabled, the device checks packets received on the interface where an IPsec policy is applied. In tunnel mode, the IP header in the decrypted IPsec packet of the inbound SA may be not defined in an ACL, for example, the IP header of attack packets may be out of the range defined in the ACL. After IPsec check is configured, the device re-checks whether the IP header of the decrypted IPsec packet is in the range defined by an ACL. If the decrypted IPsec packet matches the permit action, the device continues to process the IPsec packet. If the decrypted IPsec packet does not match the permit action, the device discards the IPsec packet. This improves network security.

When IPsec is deployed using an IPsec policy template, IPsec check function checks only the data that matches the rule with the smallest number in the ACL referenced in the IPsec policy template. If an ACL rule matches a wide range of packets, for example, **permit ip** is configured, IPsec check function may discard packets even if no tunnel exists. In this situation, if the remote device needs to receive packets, disable the IPsec check function to allow the packets to pass through.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec decrypt check**

Post-IPsec check is enabled.

By default, the device does not check decrypted IPsec packets.

----End

6.7.9 (Optional) Enabling the QoS Function for IPsec Packets

Context

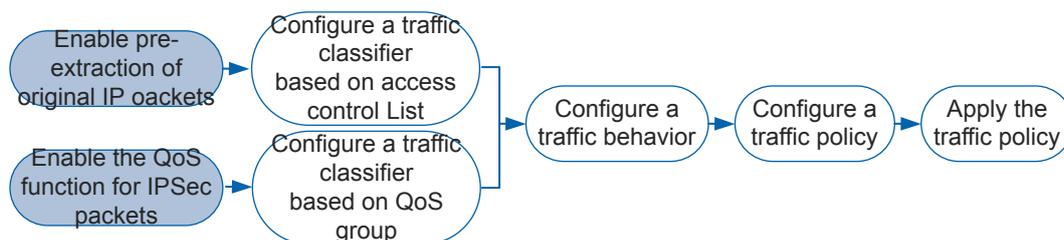
In network planning, QoS needs to be configured to provide differentiated services for different traffic flows to optimize network service capabilities. QoS groups the packets sharing common features into one class and provides the same QoS level for traffic of the same type. In this manner, QoS provides differentiated services for different types of packets.

QoS for IPsec packets implements refined QoS management on IPsec packets, choose either of the following configurations as required:

- After packets are encapsulated using IPsec, the packets do not contain QoS related parameters, such as header of the original packet and protocol number. Pre-extraction of original IP packets needs to be configured if QoS needs to group encapsulated packets based on the 5-tuple information such as original packet header and protocol number.
- When a device implements IPsec encapsulation and decapsulation on packets, it will result in transmission delay and require higher bandwidth. Therefore, the device needs to provide differentiated services for IPsec packets to reduce the delay, lower the packet loss ratio, and maximize bandwidth for IPsec traffic. You can group IPsec packets into one QoS group to allow QoS to implement differentiated services for IPsec packets.

Figure 6-30 shows the procedure for configuring QoS.

Figure 6-30 Procedure for configuring QoS



For details on QoS, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - QoS Configuring MQC*.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Enter the view where QoS for IPsec packets is configured.

- View of the IPsec policy established in manual mode
Run **ipsec policy policy-name seq-number manual**
An IPsec policy is created in manual mode and the IPsec policy view is displayed.
- View of the IPsec policy established in IKE negotiation mode
Run **ipsec policy policy-name seq-number isakmp**
An IPsec policy is created in IKE negotiation mode and the IPsec policy view is displayed.
- IPsec policy template view
Run **ipsec policy-template policy-template-name seq-number**
An IPsec policy template is created and the IPsec policy template view is displayed.

Choose either of the preceding methods.

Step 3 Enable the QoS function for IPsec packets.

- Run **qos pre-classify**
Pre-extraction of original IP packets is enabled.
By default, pre-extraction of original IP packets is disabled.

- Run **qos group qos-group-value**
The QoS group to which IPsec packets belong is configured.
By default, no QoS group is configured.

You only need to run one of the preceding commands.

----End

Follow-up Procedure

- After pre-extraction of original IP packets is enabled, run the **if-match acl { acl-number | acl-name }** command in the traffic classifier view to configure a matching rule based on the ACL.
- After QoS for IPsec packets is enabled, run the **if-match qos-group qos-group-value** command in the traffic classifier view to configure a matching rule based on the QoS group.

6.7.10 (Optional) Configuring IPsec VPN Multi-instance

Context

NOTE

If an SA is established in manual mode, you can bind a VPN instance to an IPsec tunnel in an IPsec policy. If an SA is established in IKE negotiation mode, you can bind a VPN instance to an IPsec tunnel on an IKE peer. For details, see [6.10.8 \(Optional\) Configuring IPsec VPN Multi-instance](#).

When multiple branches connected to the headquarters network across the Internet using IPsec, you can configure IPsec VPN Multi-instance, thereby isolating traffic of different branches.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy policy-name seq-number manual**

An IPsec policy is created in manual mode and the IPsec policy view is displayed.

Step 3 Run **sa binding vpn-instance vpn-instance-name**

A VPN instance is bound to an IPsec tunnel.

The VPN instance specified by *vpn-instance-name* must have been created using the **ip vpn-instance** command, and must be the same as the VPN instance bound to the ACL that is referenced by an IPsec policy.

----End

6.7.11 (Optional) Allowing New Users with the Same Traffic Rule as Original Branch Users to Access the Headquarters Network

Context

After the enterprise branch and its headquarters establish an IPsec tunnel, the IP address of the branch gateway interface to which an IPsec policy group is applied changes due to the link status change. For example, the branch gateway connects to the Internet through dial-up and establishes an IPsec tunnel with the headquarters. The headquarters gateway has an existing IPsec tunnel to protect IPsec packets exchanged between the headquarters gateway and branch gateway (original users). Because data flows of new users are the same, the branch gateway and headquarters gateway cannot reestablish an IPsec tunnel. After the local IP address of the IPsec tunnel on the branch gateway changes, the branch gateway (new users) and headquarters gateway cannot rapidly reestablish an IPsec tunnel to protect IPsec traffic exchanged between them.

You can configure the device to allow new users with the same traffic rule as original branch users to access the headquarters network so that the existing IPsec SAs can be rapidly aged and a new IPsec tunnel can be established.

NOTE

The prerequisites are as follows:

- The headquarters gateway functions as the responder and uses an IPsec policy template to establish an IPsec tunnel with the branch gateway.
- The ACL rules for the new users must be the same as those for original users.
- The interface used by new users to access the headquarters gateway must be the same as that used by original users.

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run `ipsec remote traffic-identical accept`

The device is configured to allow new users with the same traffic rule as original branch users to access the headquarters network.

By default, the device allows branch or other users to quickly access the headquarters network after their IP addresses are changed.

----End

6.7.12 (Optional) Configuring a Multi-link Shared IPsec Policy Group

Context

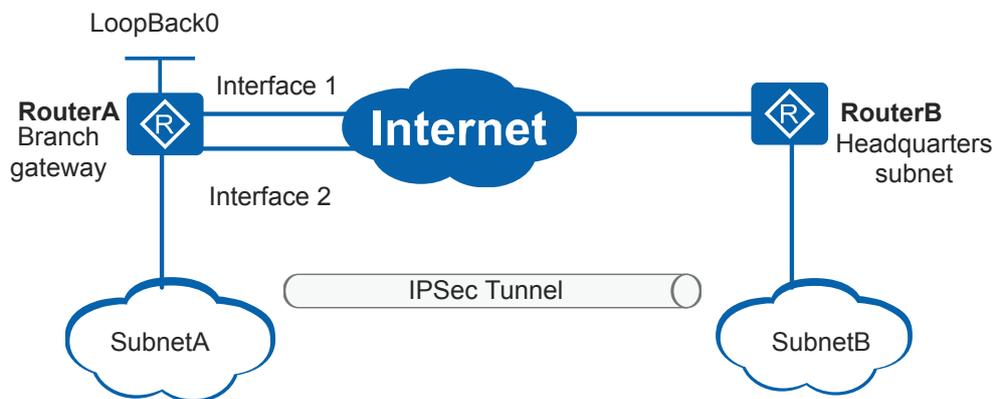
To improve network reliability, the enterprise gateway often connects to the Internet Service Provider (ISP) through two egress links, which work in backup or load balancing mode. When two outbound interfaces are configured with IPsec policies with the same parameter settings, services need to be smoothly switched between the two links corresponding to the

two outbound interfaces. The two outbound interfaces negotiate with their peers to establish IPsec SAs respectively. When an active/standby switchover occurs, the two peers need to perform IKE negotiate again to generate IPsec SAs. The IKE re-negotiation causes IPsec service interruption in a short time.

You can configure a multi-link shared IPsec policy group and use a loopback interface on the local device to establish an IPsec tunnel with the remote device. When an active/standby switchover occurs, IPsec services are not interrupted. The two IPsec-enabled physical interfaces share the same IPsec SA. When services are switched between links corresponding to the physical interfaces, the IPsec SA is not deleted as long as the loopback interface status remains unchanged. In addition, IKE re-negotiation is not required because the same IPsec SA is used to protect IPsec services.

As shown in **Figure 6-31**, packets of branch gateway RouterA reach headquarters gateway RouterB through two egress links. If an egress link is faulty, IPsec communication between RouterA and RouterB is not affected. The multi-link shared mode improves network reliability.

Figure 6-31 Using an IPsec tunnel in multi-link shared mode



NOTE

One loopback interface maps to only one multi-link shared IPsec policy group.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy *policy-name* shared local-interface loopback *interface-number***

An IPsec policy is configured as a multi-link shared security policy.

By default, no IPsec policy is configured as a multi-link shared security policy.

----**End**

6.7.13 (Optional) Configuring Redundancy Control of IPsec Tunnels

Context

On a live network, to improve network reliability, a branch gateway connects to the headquarters using multiple links. The branch gateway needs to determine on which link an IPsec tunnel is established. You can associate IPsec with NQA so that the branch gateway controls IPsec tunnel setup or teardown according to the , which ensures that only one link is available at any time. The association implements redundancy control of the IPsec tunnel.

Prerequisites

- If an NQA test instance is used to control IPsec tunnel setup or teardown, ensure that the NQA test instance has been created and configured. The device supports only association between IPsec and NQA of ICMP type. For details on how to configure an NQA test instance of ICMP type, see *Configuring an ICMP Test Instance*.
- If an NQA group is used to control IPsec tunnel setup or teardown, ensure that the NQA group has been created and bound to an NQA test instance. The device supports only association between IPsec and NQA of ICMP type. For details on how to configure an NQA group, see *Configuring an ICMP Test Instance*.
- If a BFD session is used to control IPsec tunnel setup or teardown, ensure that the BFD session has been created and configured. For details on how to configure a BFD session, see *Configuring Single-Hop BFD*.
- If a BFD group is used to control IPsec tunnel setup or teardown, ensure that the BFD group has been created and bound to a BFD session. For details on how to configure a BFD group, see *Configuring BFD Group*.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy *policy-name seq-number isakmp***

The view of the ISAKMP IPsec policy is displayed.

Step 3 Run **connect track { *nqa admin-name test-name* | *nqa-group nqa-group-name* } { **up** | **down** }**

The device is configured to control IPsec tunnel setup according to status.

By default, the device is configured to not control IPsec tunnel setup according to status.

Step 4 Run **disconnect track { *nqa admin-name test-name* | *nqa-group nqa-group-name* } { **up** | **down** }**

The device is configured to control IPsec tunnel teardown according to status.

By default, the device is configured to not control IPsec tunnel teardown according to status.

----End

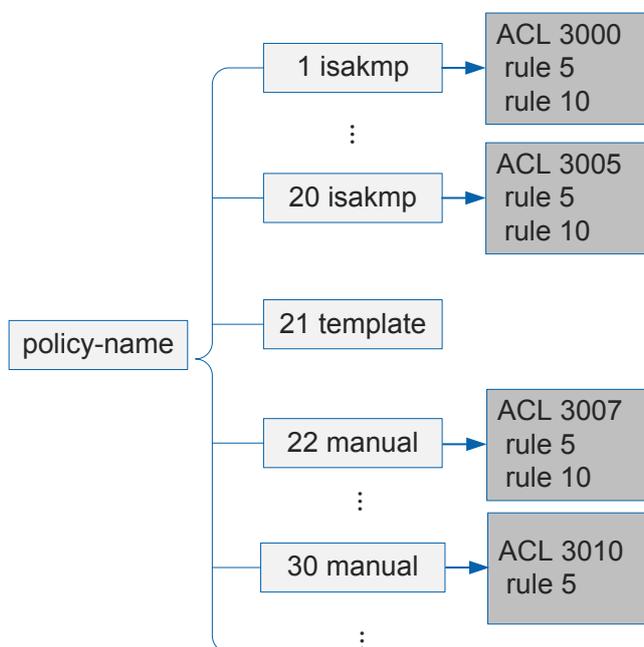
6.7.14 Applying an IPsec Policy Group to an Interface

Context

To use IPsec to protect data flows on an interface, apply an IPsec policy group to the interface. After an IPsec policy group is unbound from an interface, the interface does not provide IPsec protection.

An IPsec policy group is a set of IPsec policies with the same name but different sequence numbers. An IPsec policy group can contain multiple IPsec policies established manually or in IKE negotiation mode but only one IPsec policy template, as shown in [Figure 6-32](#). One IPsec policy corresponds to one advanced ACL. In an IPsec policy group, an IPsec policy with a smaller sequence number has a higher priority.

Figure 6-32 IPsec policy group



After an IPsec policy group is applied to an interface, all IPsec policies in the group are applied to the interface and protect different data flows.

When sending a packet, an interface matches the packet with IPsec policies in an IPsec policy group in ascending order of sequence number. If the packet matches the ACL referenced by an IPsec policy, the packet is processed based on the IPsec policy. If no matching ACL is found after all IPsec policies are checked, the interface sends the packet directly without IPsec protection.

When applying an IPsec policy group to an interface, note the following points:

- The interface where IPsec policies are applied must be the interface where an IPsec tunnel is established, and the interface must be the outbound interface in the private route to the remote end. If an IPsec policy is applied to another interface but not the target interface, VPN service forwarding may fail.
- Only one IPsec policy group can be applied to an interface, and an IPsec policy group can be applied to only one interface.

- After an IPsec policy group is applied to an interface, referenced ACLs and IKE peers in IPsec policies of the IPsec policy group cannot be modified.

 **NOTE**

- When applying an IPsec policy to a tunnel interface and running the **source** command to specify an IP address for the interface, you must run the **tunnel local** command to configure a tunnel local address. Otherwise, IKE negotiation will fail.
- When multiple branches are connected to the headquarters, if some tunnel interfaces at the headquarters borrow an IP address from a physical interface, borrow an IP address from a physical interface as their source address, or borrow a virtual IP address from a physical interface as their tunnel local address, the mappings between IKE peers and tunnel interfaces may be incorrect. As a result, an IPsec tunnel fails to be established.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run the following commands as required.

- Run **interface** *interface-type interface-number*
The interface view is displayed.
- Run **interface** *interface-type interface-number.subinterface-number*
The sub-interface view is displayed.
- Run **interface tunnel** *interface-number*
The virtual tunnel interface view is displayed.

Step 3 Run **ipsec policy** *policy-name*

An IPsec policy group is applied to the interface.

After an IPsec policy established in manual mode is applied to an interface, an SA is generated immediately.

After an IPsec policy established in IKE negotiation mode is applied to an interface, an IPsec tunnel can be triggered in auto or traffic mode using the **sa trigger-mode** { **auto** | **traffic-based** } command.

After an SA is created successfully, data flows are transmitted securely over the IPsec tunnel.

When the number of IPsec tunnels is larger than 50% of the maximum limit, high CPU usage alarms may be generated in a short period of time after the **undo ipsec policy** command is run. After all the SAs are cleared, the CPU usage restores to the normal range.

----**End**

Precautions

If you modify the **tunnel-protocol** parameter of a tunnel interface, the IPsec policy group applied to the tunnel interface will be deleted. After the modification, apply the IPsec policy group to the tunnel interface as required.

In an IPsec policy group, if multiple policies are bound to different IKE peers, the remote addresses specified in the IKE peers cannot be the same. Otherwise, IKE negotiation of some IPsec policies fails.

6.7.15 Verifying the Configuration of IPsec Tunnel Establishment

Prerequisites

The configurations of the ACL-based IPsec tunnel are complete.

Procedure

- Run the **display ipsec proposal** [**brief** | **name** *proposal-name*] or **display ipsec proposal** [**brief** | **name** *proposal-name*] **ctrl-plane** command to check IPsec proposal information.
- Run the **display ipsec policy** [**brief** | **name** *policy-name* [*seq-number*]] or **display ipsec policy** [**brief** | **name** *policy-name* [*seq-number*]] **ctrl-plane** command to check IPsec policy information.
- Run the **display ike identity** [**name** *identity-name*] command to check identity filter set information.
- Run the **display ipsec policy-template** [**brief** | **name** *policy-template-name* [*seq-number*]] or **display ipsec policy-template** [**brief** | **name** *policy-template-name* [*seq-number*]] **ctrl-plane** command to check IPsec policy template information.
- Run the **display ipsec sa** [**brief** | **duration** | **policy** *policy-name* [*seq-number*] | **remote** *ipv4-address*] command to check IPsec SA information.
- Run the **display ipsec global config** command to check global IPsec configuration.
- If an IPsec tunnel is established in IKE negotiation mode, check the IKE configuration. See [6.10.13 Verifying the IKE Configuration](#).

----End

6.8 Using a Virtual Tunnel Interface to Establish an IPsec Tunnel

A virtual tunnel interface is a Layer 3 logical interface where the encapsulation protocol is GRE, mGRE, and IPsec. The device can provide the IPsec service for the virtual tunnel interface. All the packets routed to the virtual tunnel interface are protected by IPsec. The virtual tunnel interface can simplify IPsec parameters.

Pre-configuration Tasks

Before using an IPsec tunnel interface to establish an IPsec tunnel, complete the following tasks:

- Configure a reachable route between source and destination interfaces.
- Determine data flows to be protected by IPsec and importing data flows to the IPsec tunnel interface.
- Determine parameters in an IPsec proposal.

Configuration Procedure

After an IPsec policy is referenced by an IPsec profile, apply the IPsec profile to an IPsec tunnel interface and establish an IPsec tunnel through the virtual tunnel interface.

6.8.1 Configuring an IPsec Proposal

Context

An IPsec proposal, as part of an IPsec policy or an IPsec profile, defines security parameters for IPsec SA negotiation, including the security protocol, encryption and authentication algorithms, and encapsulation mode. Both ends of an IPsec tunnel must be configured with the same parameters.

NOTE

The IPsec proposal referenced by an IPsec tunnel interface supports only the tunnel mode, that is, you must specify **tunnel** in the **encapsulation-mode** command.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec proposal proposal-name**

An IPsec proposal is created and the IPsec proposal view is displayed.

Step 3 Run **transform { ah | esp | ah-esp }**

A security protocol is configured.

By default, an IPsec proposal uses ESP.

Step 4 An authentication or encryption algorithm is configured.

- If AH is used, you can only configure the AH-specific authentication algorithm because AH only authenticates packets.

Run **ah authentication-algorithm { md5 | sha1 | sha2-256 | sha2-384 | sha2-512 }**

An AH-specific authentication algorithm is configured.

By default, AH uses the SHA2-256 authentication algorithm.

- When ESP is specified, ESP can encrypt/authenticate, or encrypt and authenticate packets. Configure the ESP-specific authentication or encryption algorithm.

– Run **esp authentication-algorithm { md5 | sha1 | sha2-256 | sha2-384 | sha2-512 }**

An ESP-specific authentication algorithm is configured.

By default, ESP uses the SHA2-256 authentication algorithm.

– Run **esp encryption-algorithm { 3des | des | aes-128 | aes-192 | aes-256 }**

An ESP-specific encryption algorithm is configured.

By default, ESP uses the AES-256 encryption algorithm.

- When both AH and ESP are used, AH authenticates packets, and ESP can encrypt and authenticate packets. You can choose to configure an AH-specific authentication algorithm, or ESP-specific authentication and encryption algorithms. The device first encapsulates the ESP header, and then the AH header to packets.

 **NOTE**

- Authentication algorithms SHA2-256, SHA2-384, and SHA2-512 are recommended to improve packet transmission security, whereas authentication algorithms MD5 and SHA1 are not recommended.
- Encryption algorithms AES-128, AES-192, and AES-256 are recommended to improve packet transmission security, whereas encryption algorithm DES and 3DES are not recommended.

Step 5 Run **encapsulation-mode { transport | tunnel }**

An IP packet encapsulation mode is configured.

By default, IPsec uses the tunnel mode to encapsulate IP packets.

When IKEv2 is used, the encapsulation modes in all the IPsec proposals configured on the IKE initiator must be the same; otherwise, IKE negotiation fails.

 **NOTE**

When L2TP over IPsec or GRE over IPsec is configured, a public IP header is added to packets during L2TP or GRE encapsulation. Compared with the transport mode, the tunnel mode adds another public IP header. In tunnel mode, the packet length is longer and packets are more likely to be fragmented. The transport mode is therefore recommended.

Step 6 Run **quit**

Exit the IPsec proposal view.

Step 7 (Optional) Run **ipsec authentication sha2 compatible enable**

The SHA-2 algorithm is compatible with earlier software versions.

By default, the SHA-2 algorithm is not compatible with earlier software versions.

When IPsec uses the SHA-2 algorithm, if the devices on two ends of an IPsec tunnel are from different vendors or run different software versions, they may use different encryption and decryption methods. In this situation, traffic between devices is interrupted.

To solve this problem, enable SHA-2 to be compatible with earlier versions.

----End

6.8.2 Configuring an IPsec Profile

Context

An IPsec profile defines how to protect data flows, including IPsec proposals, IKE negotiation parameters for SA setup, SA lifetime, and PFS status. An IPsec profile is similar to an **IPsec Policy**. Compared with the IPsec policy, the IPsec profile is identified by its name and can be configured only in IKE negotiation mode.

In an IPsec profile, you do not need to use ACL rules to define data flows. Instead, all the data flows routed to the IPsec tunnel interface are protected. After an IPsec profile is applied to an IPsec tunnel interface, only one IPsec tunnel is created. The IPsec tunnel protects all the data flows routed to the IPsec tunnel interface, simplifying IPsec policy management.

To ensure successful IKE negotiation, parameters in the IPsec profile on the local and remote ends must match.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec profile *profile-name***

An IPsec profile is created and the IPsec profile view is displayed.

By default, no IPsec profile is created.

Step 3 Run **proposal *proposal-name***

An IPsec proposal is referenced in the IPsec profile.

By default, no IPsec proposal is referenced in an IPsec profile.

The IPsec proposal must have been created.

Step 4 Run **ike-peer *peer-name***

An IKE peer is referenced in the IPsec profile.

By default, no IKE peer is referenced in an IPsec profile.

The IKE peer must have been created.

NOTE

- You do not need to specify the **tunnel local** (local address) for the IKE peer referenced in an IPsec profile, because the local address is the source address of the GRE, mGRE or IPsec virtual tunnel interface. For the IKE peer referenced in an IPsec profile, **tunnel local** do not take effect.
- When an IPsec profile is used, the destination address of the IPsec tunnel interface configured using the **destination** command is preferentially used as the remote address for IKE negotiation. When the **remote-address** and **destination** commands are configured at the same time, ensure that the configured IP addresses are the same; otherwise, IKE negotiation will fail. To implement IKE peer redundancy, do not configure the **destination** command on the IPsec tunnel interface. Instead, configure the **remote-address** command on the IKE peer referenced by the IPsec profile.
- For the detailed configuration of an IKE peer, see [6.10.2 Configuring an IKE Peer](#).

Step 5 (Optional) Run **match ike-identity *identity-name***

The identity filter set is referenced.

identity-name is an identity filter that has been created.

NOTE

For details on how to configure an identity filter set, see [6.10.5 \(Optional\) Configuring an Identity Filter Set](#).

Step 6 (Optional) Run **pfs { *dh-group1* | *dh-group2* | *dh-group5* | *dh-group14* | *dh-group19* | *dh-group20* | *dh-group21* }**

The device is configured to use perfect forward secrecy (PFS) when the local end initiates negotiation.

By default, PFS is not used when the local end initiates negotiation.

When the local end initiates negotiation, there is an additional Diffie-Hellman (DH) exchange in IKEv1 phase 2 or IKEv2 CREATE_CHILD_SA exchange. The additional DH exchange ensures security of the IPsec SA key and improves communication security.

If PFS is specified on the local end, you also need to specify PFS on the remote end. The DH group specified on the two ends must be the same; otherwise, negotiation fails. When an IPsec policy in ISAKMP mode is used on the local end while an IPsec policy configured using an IPsec policy template is used on the remote end, no DH group needs to be configured on the remote end. The DH group on the responder is used for negotiation.

---End

6.8.3 (Optional) Setting the SA Lifetime

Context

NOTE

- The configured IPsec SA lifetime is only valid for the new IPsec SAs established in IKE negotiation mode.

For a dynamic SA, configure the SA hard lifetime so that the SA can be updated in real time, reducing the crash risk and improving security.

There are two methods to measure the lifetime:

- Time-based lifetime
The period from when an SA is set up to when the SA is expired.
- Traffic-based lifetime
The maximum volume of traffic that this SA can process.

The lifetime is classified as follows:

- Hard lifetime: specifies the lifetime of an IPsec SA.
When two devices negotiate an IPsec SA, the actual hard lifetime is the smaller of the two values configured on the two devices.
- Soft lifetime: specifies the time after which a new IPsec SA is negotiated so that the new IPsec SA will be ready before the hard lifetime of the original IPsec SA expires.

[Table 6-8](#) lists the default soft lifetime values.

Table 6-8 Soft lifetime values

Soft Lifetime Type	Description
Time-based soft lifetime (soft timeout period)	The value is 7/10 of the actual hard lifetime (hard timeout period).
Traffic-based soft lifetime (soft timeout traffic)	The value is 7/10 of the actual hard lifetime (hard timeout traffic).

Before an IPsec SA becomes invalid, IKE negotiates a new IPsec SA for the remote end. The remote end uses the new IPsec SA to protect IPsec communication immediately after the new IPsec SA is negotiated. If service traffic is transmitted, the original IPsec SA is deleted immediately. If no service traffic is transmitted, the original IPsec SA will be deleted after 10s or the hard lifetime expires.

If the time-based lifetime and traffic-based lifetime are both set for an IPsec SA, the IPsec SA becomes invalid when either lifetime expires.

You can set the global SA hard lifetime or set the SA hard lifetime in an IPsec profile. If the SA hard lifetime is not set in an IPsec profile, the global hard lifetime is used. If both the global SA hard lifetime and the SA hard lifetime in an IPsec profile are set, the SA hard lifetime in the IPsec profile takes effect.

Procedure

- Set the global IPsec SA hard lifetime.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **ipsec sa global-duration** { **time-based** *interval* | **traffic-based** *size* }
The global IPsec SA hard lifetime is set.

By default, the global time-based SA hard lifetime is 3600 seconds and the global traffic-based SA hard lifetime is 1843200 Kbytes.
- Setting the IPsec SA hard lifetime in an IPsec profile.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **ipsec profile** *profile-name*
An IPsec profile is created and the IPsec profile view is displayed.
 - c. Run **sa duration** { **time-based** *interval* | **traffic-based** *size* }
The IPsec SA hard lifetime is set in the IPsec profile.

By default, the IPsec SA hard lifetime is not set in an IPsec profile. The system uses the global IPsec SA hard lifetime.

----End

6.8.4 (Optional) Enabling the Anti-replay Function

Context

Replayed packets are packets that have been processed. IPsec uses the sliding window (anti-replay window) mechanism to check replayed packets. Each AH or ESP packet has a 32-bit sequence number. In an SA, sequence numbers of packets increase. If the sequence number of a received authenticated packet is the same as that of a decapsulated packet or if the sequence number is out of the sliding window, the device considers the packet as a replayed packet.

Decapsulating replayed packets consumes many resources and makes system performance deteriorate, resulting in a Denial Of Service (DoS) attack. After the anti-replay function is enabled, the system discards replayed packets and does not encapsulate them, saving system resources.

In some situations, for example, when network congestion occurs or QoS is performed for packets, the sequence numbers of some service data packets may be different from those in common data packets. The device that has IPsec anti-replay enabled considers the packets as replayed packets and discards them. You can disable global IPsec anti-replay to prevent packets from being discarded incorrectly or adjust the IPsec anti-replay window size to meet service requirements.

The anti-replay function can be configured globally or in an IPsec profile.

- Configuring the anti-replay function globally
The global anti-replay function is valid for all created IPsec profiles. When the same anti-replay window parameters need to be set for many IPsec profiles, you do not need to run commands one by one. You just need to set global parameters. The configuration efficiency is therefore improved.
- Configuring the anti-replay function in an IPsec profile
The anti-replay function can be configured separately for an IPsec profile. In this case, the anti-replay function for the IPsec profile is not affected by the global configuration.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Enable the anti-replay function. Run the following commands as required.

- Enable the anti-replay function globally.
 - a. Run **ipsec anti-replay enable**
The anti-replay function is enabled globally.
 - b. Run **ipsec anti-replay window *window-size***
The global IPsec anti-replay window size is configured.
By default, the IPsec anti-replay window size is 1024 bits.
- Enable the anti-replay function in an IPsec policy.
 - a. Run **ipsec profile *profile-name***
An IPsec profile is created and the IPsec profile view is displayed.
 - b. Run **anti-replay window *window-size***
The IPsec anti-replay window size is configured in the IPsec profile.
By default, the anti-replay window size of a single IPsec tunnel is not set. The global value is used.

----End

6.8.5 (Optional) Configuring IPsec Fragmentation Before Encryption

Context

The length of IPsec-encapsulated packets may exceed the maximum transmission unit (MTU) of the outbound interface on the local device. If the IPsec remote device does not support fragmentation and reassembly, it cannot decapsulate packets and will discard or incorrectly process packets, affecting packet transmission.

To prevent this problem, configure IPsec fragmentation before encryption on the local device. Subsequently, the local device calculates the length of encapsulated packets. If the length exceeds the MTU, the device fragments the packets and then encapsulates each fragment. After packets reach the IPsec remote device, the remote device can decapsulate the fragments without having to reassemble them. The decapsulated packets will be forwarded normally.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec fragmentation before-encryption**

The fragmentation mode of packets is set to fragmentation before encryption for all IPsec tunnels.

By default, the packet fragmentation mode for all IPsec tunnels is fragmentation after encryption.

The DF flag in IPsec packets determines whether IPsec packets can be fragmented. If DF flag settings disable fragmentation when the fragmentation mode is used, run the **ipsec df-bit { clear | set | copy }** command in the system view to enable fragmentation on IPsec packets.

For the established IPsec tunnels, you need to restart them after running this command. Otherwise, the command function does not take effect.

----End

6.8.6 (Optional) Configuring IPsec Check

Context

IPsec check ensures that data flows are correctly encrypted. After IPsec check is enabled, the device checks packets received on the interface where an IPsec policy is applied.

After an IPsec profile is applied to a virtual tunnel interface, the device generates an ACL based on the encapsulation mode of the tunnel interface. In tunnel mode, the IP header in the decapsulated IPsec packet of the inbound SA may be not defined in an ACL. For example, the IP header of attack packets may be out of the range defined in the ACL. After post-IPsec check is configured, the device re-checks whether the IP header of the decapsulated IPsec packet is in the range defined in an ACL. If the IP header matches the permit rule, the device performs subsequent operations on the packet. Otherwise, the device discards the IPsec packet. The network security is therefore improved.

The generated ACL varies depending on the encapsulation mode of the virtual tunnel interface:

- When the encapsulation mode is set to IPsec, the source and destination addresses in the ACL are both any, indicating that all data flows destined for the tunnel interface are protected.
- When the encapsulation mode is set to GRE, the source and destination addresses in the ACL are the source and destination addresses of the tunnel interface respectively.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec decrypt check**

Post-IPsec check is enabled.

By default, the device does not check decrypted IPsec packets.

---End

6.8.7 (Optional) Enabling the QoS Function for IPsec Packets

Context

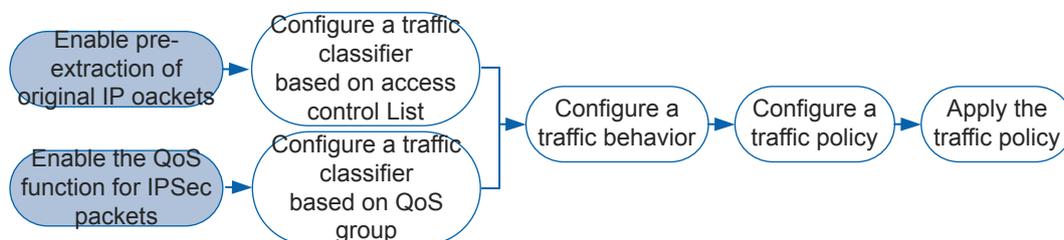
In network planning, QoS needs to be configured to provide differentiated services for different traffic flows to optimize network service capabilities. QoS groups the packets sharing common features into one class and provides the same QoS level for traffic of the same type. In this manner, QoS provides differentiated services for different types of packets.

QoS for IPsec packets implements refined QoS management on IPsec packets, choose either of the following configurations as required:

- After packets are encapsulated using IPsec, the packets do not contain QoS related parameters, such as header of the original packet and protocol number. Pre-extraction of original IP packets needs to be configured if QoS needs to group encapsulated packets based on the 5-tuple information such as original packet header and protocol number.
- When a device implements IPsec encapsulation and decapsulation on packets, it will result in transmission delay and require higher bandwidth. Therefore, the device needs to provide differentiated services for IPsec packets to reduce the delay, lower the packet loss ratio, and maximize bandwidth for IPsec traffic. You can group IPsec packets into one QoS group to allow QoS to implement differentiated services for IPsec packets.

Figure 6-33 shows the procedure for configuring QoS.

Figure 6-33 Procedure for configuring QoS



For details on QoS, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - QoS Configuring MQC*.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Enter the view where QoS for IPsec packets is configured.

- IPsec profile view
Run **ipsec profile** *profile-name*
An IPsec profile is created and the IPsec profile view is displayed.
- The tunnel interface view

Run **interface tunnel** *interface-number*

The tunnel interface view is displayed.

Choose either of the preceding methods.

Step 3 Enable the QoS function for IPsec packets.

- Run **qos pre-classify**

Pre-extraction of original IP packets is enabled.

By default, pre-extraction of original IP packets is disabled.

- Run **qos group** *qos-group-value*

The QoS group to which IPsec packets belong is configured.

By default, no QoS group is configured.

 **NOTE**

This command can only be executed in the IPsec profile view.

You only need to run one of the preceding commands.

----End

Follow-up Procedure

- After pre-extraction of original IP packets is enabled, run the **if-match acl** { *acl-number* | *acl-name* } command in the traffic classifier view to configure a matching rule based on the ACL.
- After QoS for IPsec packets is enabled, run the **if-match qos-group** *qos-group-value* command in the traffic classifier view to configure a matching rule based on the QoS group.

6.8.8 (Optional) Configuring Requesting, Sending or Accepting of Subnet Route Information

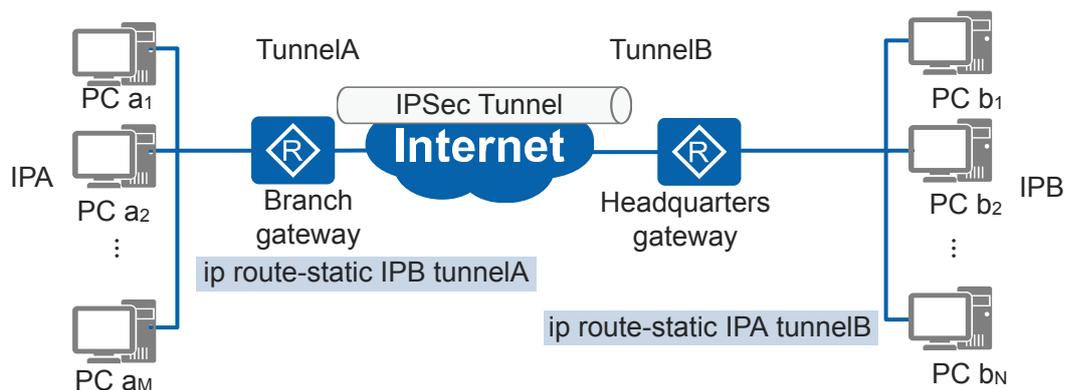
Context

As shown in [Figure 6-34](#), a headquarters sets up an IPsec tunnel with its branch using a tunnel interface. If requesting, sending or accepting of subnet route information is not configured, data flows to be protected through IPsec need to be imported to the tunnel interface based on the static or dynamic routes.

In the IPsec configuration,

- When the remote network topology changes, you must modify routes on the local device to ensure IPsec protection for nonstop data transmission.
- When a new branch is added to the network and wants to establish an IPsec tunnel with the headquarters, you must configure routes to the branch on the headquarters gateway.

Figure 6-34 Requesting, sending or accepting of subnet route information not configured

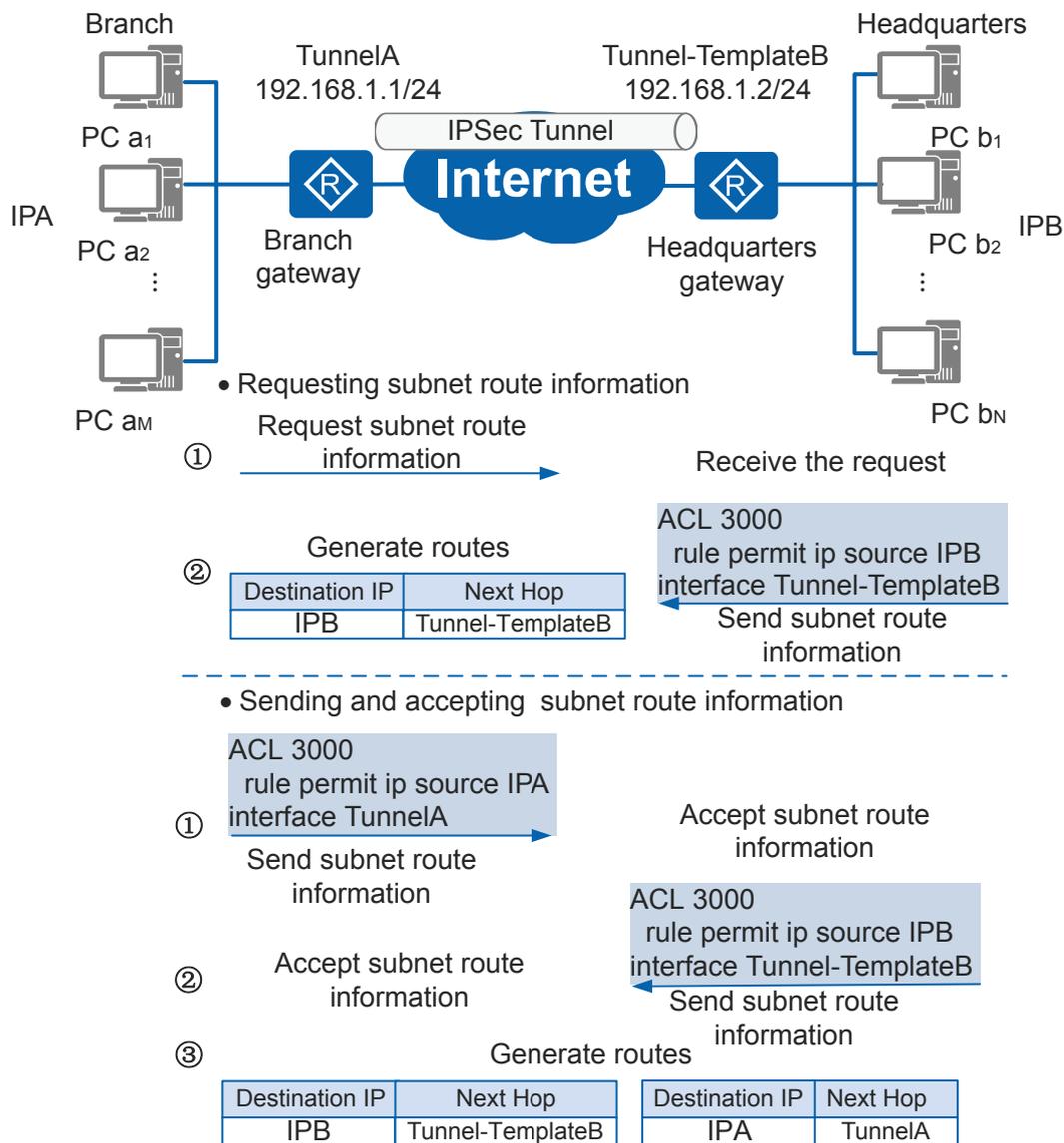


As shown in [Figure 6-35](#), you only need to configure the subnet address to be protected through IPsec on the local device. The local device then sends subnet route information to the remote device that generates routes based on the information.

In the IPsec configuration,

- When the remote network topology changes, you do not need to modify routes on the local device.
- When a new branch is added to the network and wants to establish an IPsec tunnel with the headquarters, you do not need to add routes to the branch on the headquarters gateway.

Figure 6-35 Requesting, sending or accepting of subnet route information configured



NOTE

This function is supported by IKEv2 only.

Procedure

- Configuring requesting of subnet route information

After a local device is enabled to request subnet route information, the remote device will send subnet route information directly without enabling send subnet route information.

NOTE

Requesting of subnet route information takes effect on the IKE negotiation initiator only.

The local device requests subnet route information, generates a route based on the received subnet route information.

- a. Run **system-view**
The system view is displayed.
- b. Run **ike peer** *peer-name*
An IKE peer is created and the IKE peer view is displayed.
- c. Run **undo version 1**
The IKE protocol version used by IKE peers is configured.
By default, IKE peers support IKEv1 and IKEv2.
If IKEv1 and IKEv2 are enabled, IKEv2 is used in the negotiation initiation, and IKEv1 and IKEv2 are used in negotiation response.
- d. (Optional) Run **config-exchange request**
The device is enabled to request subnet route information from a peer.
By default, the device does not request subnet route information from a peer.
- e. Run **route accept** [**preference** *preference-number*] [**tag** *tag-value*]
The device is enabled to generate a route based on the received subnet route information and define the priority and tag value for the route.
By default, the device does not generate routes based on the received subnet route information.

The remote device configures subnet route information to be sent, sends subnet route information directly after receiving the request.

- a. Run **system-view**
The system view is displayed.
 - b. Run **aaa**
The AAA view is displayed.
 - c. Run **service-scheme** *service-scheme-name*
A service scheme is created and the service scheme view is displayed.
 - d. Run **route set acl** *acl-number*
Local subnet information to be sent to the peer is configured.
By default, no local subnet information to be sent to the peer is configured.
acl-number specifies an advanced ACL that has been created.
 - e. Run **route set interface**
An IP address which is to be sent to the peer is configured for the interface.
By default, no IP address which is to be sent to the peer is configured for the interface.
- **Configuring sending and accepting of subnet route information**
After a local device is enabled to send subnet route information and a remote device is enabled to accept subnet route information, sending of subnet route information is enabled in one direction. To enable bidirectional sending of subnet route information, the headquarters and branch devices must be enabled to send and accept subnet route information at the same time.

The local device sends subnet route information.

- a. Configure subnet route information to be sent.

- i. Run **system-view**
The system view is displayed.
 - ii. Run **aaa**
The AAA view is displayed.
 - iii. Run **service-scheme** *service-scheme-name*
A service scheme is created and the service scheme view is displayed.
 - iv. Run **route set acl** *acl-number*
Local subnet information to be sent to the peer is configured.
By default, no local subnet information to be sent to the peer is configured.
acl-number specifies an advanced ACL that has been created.
 - v. Run **route set interface**
An IP address which is to be sent to the peer is configured for the interface.
By default, no IP address which is to be sent to the peer is configured for the interface.
- b. Configure sending of subnet route information.
- i. Run **quit**
Return to the AAA view.
 - ii. Run **quit**
Return to the system view.
 - iii. Run **ike peer** *peer-name*
An IKE peer is created and the IKE peer view is displayed.
 - iv. Run **undo version 1**
The IKE protocol version used by IKE peers is configured.
By default, IKE peers support IKEv1 and IKEv2.
If IKEv1 and IKEv2 are enabled, IKEv2 is used in the negotiation initiation, and IKEv1 and IKEv2 are used in negotiation response.
 - v. Run **service-scheme** *service-scheme-name*
Binds a service scheme to an IKE peer.
By default, no service scheme binds to an IKE peer.
 - vi. Run **config-exchange set send**
The device is enabled to send subnet route information to a peer.
By default, the device does not send subnet route information to a peer.

The remote device accepts subnet route information.

- a. Run **system-view**
The system view is displayed.
- b. Run **ike peer** *peer-name*
An IKE peer is created and the IKE peer view is displayed.
- c. Run **undo version 1**
The IKE protocol version used by IKE peers is configured.
By default, IKE peers support IKEv1 and IKEv2.

If IKEv1 and IKEv2 are enabled, IKEv2 is used in the negotiation initiation, and IKEv1 and IKEv2 are used in negotiation response.

d. Run **config-exchange set accept**

The device is enabled to accept subnet route information from a peer.

By default, the device does not accept subnet route information from a peer.

e. Run **route accept [preference *preference-number*] [tag *tag-value*]**

The device is enabled to generate a route based on the received subnet route information and define the priority and tag value for the route.

By default, the device does not generate routes based on the received subnet route information.

----End

6.8.9 Configuring a Tunnel Interface or a Tunnel Template Interface

Context

A tunnel interface is a Layer 3 logical interface where the encapsulation protocol of GRE, mGRE, and IPsec, the device can provide IPsec service. The IPsec tunnel interface is established based on IKE negotiation. After you configure a tunnel interface and apply an IPsec profile to the tunnel interface, the IPsec tunnel is set up.

The IP address of an IPsec tunnel interface can be manually configured or dynamically requested through IKEv2 negotiation. Dynamically requesting an IP address of the IPsec tunnel interface through IKEv2 negotiation reduces the configuration and maintenance workload of branch devices in scenarios where many branches connect to the headquarters.

A tunnel template interface is similar to a tunnel interface; however, the tunnel template interface can only function as the responder but not the initiator. Generally, a tunnel template interface is created on the headquarters gateway. When a new branch gateway is added to the network, the headquarters gateway will generate a virtual tunnel interface dynamically.

NOTE

If you apply an IPsec profile to the tunnel template interface, the IKE peer referenced in the IPsec profile can only be IKEv2.

When multiple branches are connected to the headquarters, if some tunnel interfaces at the headquarters borrow an IP address from a physical interface and borrow an IP address from a physical interface as their source address, the mappings between IKE peers and tunnel interfaces may be incorrect. As a result, an IPsec tunnel fails to be established.

Procedure

- Configuring a Tunnel Interface
 - a. Run **system-view**
The system view is displayed.
 - b. Run **interface tunnel *interface-number***
The tunnel interface view is displayed.
 - c. Run **tunnel-protocol { gre [*p2mp*] | ipsec }**
The encapsulation mode of a tunnel interface is configured.

 **NOTE**

An IPsec profile can be bound to an IPsec tunnel interface only when the tunnel encapsulation mode is set to IPsec, GRE, or Multipoint GRE (mGRE):

- IPsec: An IPsec tunnel established on a tunnel interface ensures security of unicast data transmitted on the Internet.
- GRE: The IPsec tunnel interface provides GRE over IPsec and transmits unicast and multicast data. The IPsec tunnel interface first adds a GRE header to packets, and then adds an IPsec header to the packets so that packets are reliably transmitted.
- mGRE (specified by **gre** and **p2mp**): The IPsec tunnel interface provides Dynamic Smart Virtual Private Network (DSVPN) functions. See [5 DSVPN Configuration](#).

d. Run the following commands as required.

- Run **ip address ip-address { mask | mask-length } [sub]**

A private IPv4 address is configured for the tunnel interface.

- On the IPsec tunnel interface, run **ip address ike-negotiated**

An IPv4 address is requested for the tunnel interface through IKEv2 negotiation.

e. Run **source { [vpn-instance vpn-instance-name] source-ip-address | interface-type interface-number }**

The source address or source interface is configured.

You can specify the **vpn-instance vpn-instance-name** parameter only when the encapsulation mode of a tunnel interface is set to IPsec or mGRE.

 **NOTE**

It is recommended that the source interface be specified. This is because a dynamic IP address may affect IPsec configuration recovery.

f. (Optional) Run **destination [vpn-instance vpn-instance-name] dest-ip-address**

The destination address is configured.

When the destination address of an IPsec tunnel interface is not configured, the remote address of the IKE peer referenced by the IPsec profile can be used for initiating negotiation. When the destination address of an IPsec tunnel interface and remote address of an IKE peer are not configured, the local end can only accept the negotiation request initiated by the remote end.

If the encapsulation mode of a tunnel interface is set to GRE, you need to configure destination addresses at both ends.

g. (Optional) Run **tunnel pathmtu enable**

The device is enabled to learn the maximum transmission unit (MTU) of packets allowed on an IPsec tunnel.

By default, the device cannot learn the MTU of packets allowed on an IPsec tunnel.

 **NOTE**

This command takes effect only when the encapsulation mode of the tunnel interface is **IPsec** or **GRE** and the **destination** command has been configured on the tunnel interface.

h. Run **ipsec profile profile-name**

An IPsec profile is applied to the tunnel interface.

By default, no IPsec profile is applied to a tunnel interface.

Only one IPsec profile can be applied to a tunnel interface, and an IPsec profile can be applied to only one tunnel interface.

When the number of IPsec tunnels is larger than 50% of the maximum limit, high CPU usage alarms may be generated in a short period of time after the **undo ipsec profile** command is run. After all the SAs are cleared, the CPU usage restores to the normal range.

- i. (Optional) Run **standby interface** *interface-type interface-number* [*priority*]

A standby tunnel interface is configured and its priority is specified.

By default, no standby tunnel interface is configured.

The headquarters provides two gateways and more than two gateways for the branch gateway to improve network reliability. When an IPsec tunnel is set up using virtual tunnel interfaces, you can configure a standby tunnel interface on the branch gateway and apply an IPsec profile to the standby interface to provide a standby link for IPsec setup. Meanwhile, you need to configure the **heartbeat** or **DPD** mechanism to implement fast switching between the active and standby tunnels upon a tunnel fault.

- Configuring a Tunnel Template Interface

- a. Run **system-view**

The system view is displayed.

- b. Run **interface tunnel-template** *interface-number*

The tunnel template interface view is displayed.

- c. Configuring the IP address of the tunnel template interface.

- Run **ip address** *ip-address* { *mask* | *mask-length* } [**sub**]

The IPv4 address of the tunnel template interface is configured.

- Run **ip address unnumbered interface** *interface-type interface-number*

The tunnel template interface is configured to borrow an IP address from another interface.

You only need to run one of the preceding commands.

- d. Run **tunnel-protocol ipsec**

The encapsulation mode of the tunnel template interface is set to IPsec.

- e. Run **source** { [**vpn-instance** *vpn-instance-name*] *source-ip-address* | *interface-type interface-number* }

The source address or source interface is configured for the tunnel template interface.

 **NOTE**

If the source address of the tunnel template interface is dynamically obtained, you are advised to specify the source interface when running the **source** command. This prevents the impact of address change on the IPsec configuration.

- f. (Optional) Run **tunnel pathmtu enable**

The device is enabled to learn the MTU of packets allowed on an IPsec tunnel.

By default, the device cannot learn the MTU of packets allowed on an IPsec tunnel.

- g. Run **ipsec profile** *profile-name*

The IPsec profile is applied to a tunnel template interface so that data flows on the interface are protected by IPsec.

By default, no IPsec profile is applied to the tunnel template interface.

You can apply only one IPsec profile to a tunnel template interface. An IPsec profile can be applied to only one tunnel template interface.

When the number of IPsec tunnels is larger than 50% of the maximum limit, high CPU usage alarms may be generated in a short period of time after the **undo ipsec profile** command is run. After all the SAs are cleared, the CPU usage restores to the normal range.

---End

Configuration Guidelines

- The IPsec profile configuration applied to a tunnel interface is deleted if you modify the value of the parameter **source** or **destination** on the tunnel interface. Apply the IPsec profile to the tunnel interface again.
- If you modify the **tunnel-protocol** parameter of a tunnel interface, the IPsec policy group applied to the tunnel interface will be deleted. After the modification, apply IPsec policy group to the tunnel interface as required.
- The IPsec profile configuration applied to a tunnel template interface is deleted if you modify the value of the parameter **source** on the tunnel template interface. Apply the IPsec profile to the tunnel template interface again.
- To disable IPsec negotiation, you must run the **shutdown** command to shut down the corresponding physical interface but not the tunnel interface.

6.8.10 Verifying the Configuration of IPsec Tunnel Establishment Using a Virtual Tunnel Interface

Prerequisites

The configurations of the IPsec tunnel that is established using a virtual tunnel interface are complete.

Procedure

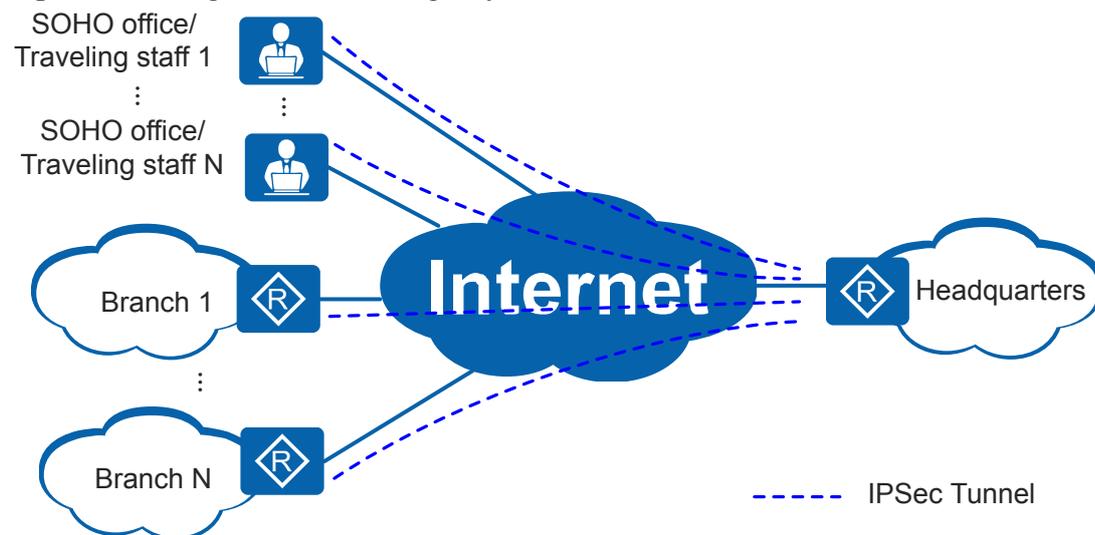
- Run the **display ipsec proposal [brief | name *proposal-name*]** or **display ipsec proposal [brief | name *proposal-name*] ctrl-plane** command to check IPsec proposal information.
- Run the **display ipsec profile [brief | name *profile-name*]** or **display ipsec profile [brief | name *profile-name*] ctrl-plane** command to check IPsec profile information.
- Run the **display ike identity [name *identity-name*]** command to check IKE identity information.
- Run the **display ipsec sa [brief | duration | policy *policy-name* [*seq-number*] | remote *ipv4-address*]** command to check IPsec SA information.
- Run the **display ipsec global config** command to check global IPsec configuration.
- Check IKE information. See [6.10.13 Verifying the IKE Configuration](#).

---End

6.9 Establishing an IPsec Tunnel Using an Efficient VPN Policy

Context

Figure 6-36 Using an Efficient VPN policy to establish an IPsec tunnel



Pre-configuration Tasks

Before using an Efficient VPN policy to establish an IPsec tunnel, complete the following tasks:

- Configure a reachable route between source and destination interfaces.
- Determine the initiator as the remote device and the responder as the Efficient VPN server.
- Determine data flows to be protected by IPsec.
- Determine parameters in an IPsec proposal.

Configuration Procedure

Configure an Efficient VPN policy on a remote device, and configure an IPsec policy template on the Efficient VPN server to establish an IPsec tunnel.

6.9.1 Configuring the Remote Device

Context

Only mandatory parameters, such as the Efficient VPN server IP address and pre-shared key, need to be configured on a remote device. Other parameters, such as authentication and encryption algorithms used in IKE negotiation, and the IPsec proposal, are preconfigured on

the Efficient VPN server. Configuring parameters on the remote device includes configuring basic and optional parameters:

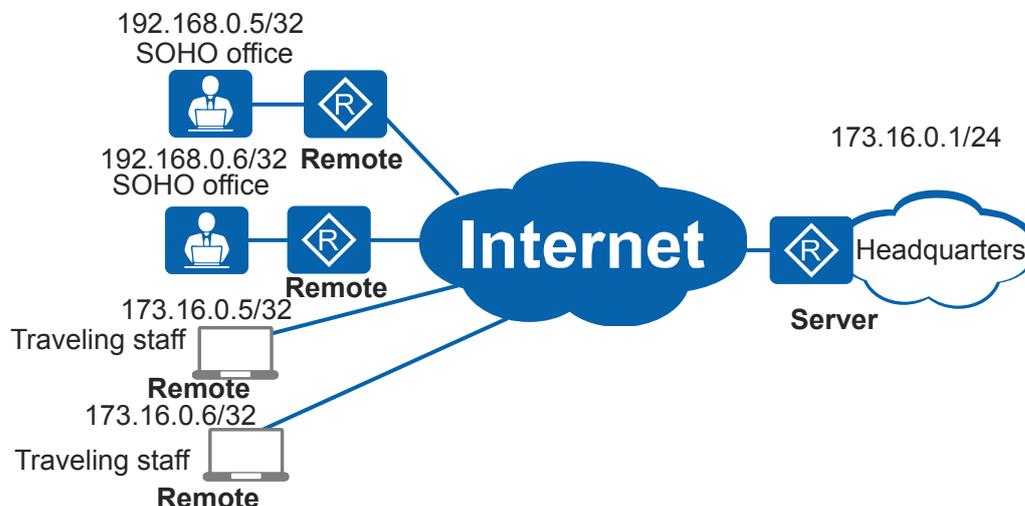
1. Basic parameters: Efficient VPN operation mode, IP address of the remote device connected to the Efficient VPN server, and authentication key.
2. Optional parameters can be set on the remote device and Efficient VPN server. If optional parameters are configured at one end, the two ends use these parameters. If optional parameters are configured at two ends, the two ends must use the same parameters to implement successful IKE negotiation.

Efficient VPN provides the following modes:

- Client mode
 - a. When a remote device requests an IP address from the Efficient VPN server, a loopback interface is dynamically created on the remote device and the IP address obtained from the server is assigned to the loopback interface.
 - b. The remote device automatically enables NAT to translate its original IP address into the obtained IP address, and then uses this IP address to establish an IPsec tunnel with the headquarters.

The client mode applies to scenarios where traveling staff or small-scale branches connect to the headquarters network through private networks, as shown in [Figure 6-37](#). In client mode, devices connected to the Efficient VPN server or remote devices can use the same IP address. However, the number of devices allowed depends on the number of IP addresses assigned by the Efficient VPN server.

Figure 6-37 Client mode

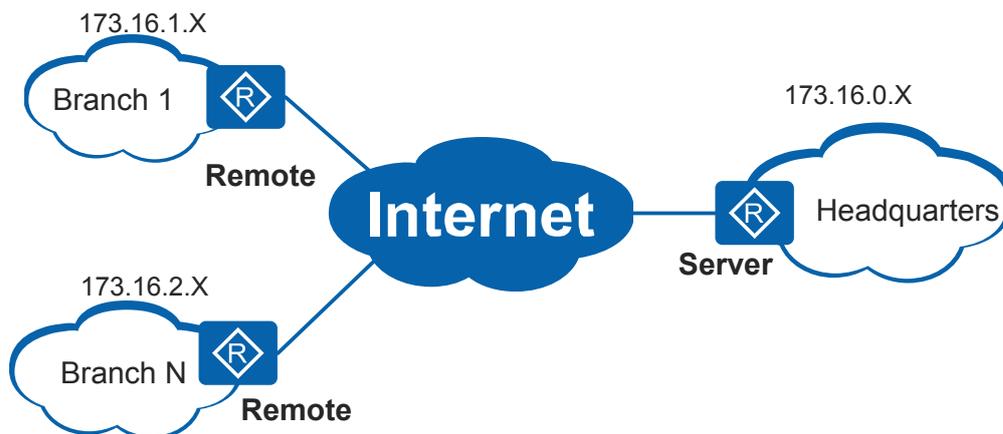


NOTE

Traveling staff use software to establish a virtual network adapter on a PC. The virtual network adapter then uses parameters such as addresses sent by the Efficient VPN server.

- Network mode
In network mode, a remote device does not apply to the Efficient VPN server for an IP address. Therefore, NAT is not enabled in network mode. [Figure 6-38](#) shows the network mode.

Figure 6-38 Network mode



The network mode applies to scenarios where IP addresses of the headquarters and branches are planned uniformly. Ensure that IP addresses do not conflict.

- Network-plus mode

Compared with the network mode, the remote device applies to the Efficient VPN server for an IP address in network-plus mode. IP addresses of branches and headquarters are configured beforehand. A remote device applies to the Efficient VPN server for an IP address. The Efficient VPN server uses the IP address to perform ping, STelnet, or other management and maintenance operations on the remote device. NAT is not enabled on the remote device.

- Network-auto-cfg mode

Compared with the network-plus mode, the remote device applies to the Efficient VPN server for an IP address pool in network-auto-cfg mode. The IP address pool is used for allocating addresses to users.

The Efficient VPN server also delivers the following resources in addition to parameters for establishing an IPsec tunnel:

- Network resources including the DNS domain name, DNS server IP addresses, and WINS server IP addresses

The Efficient VPN server delivers the preceding resources so that branches can access them on the Efficient VPN server.

- ACL resources

The Efficient VPN server delivers headquarters network information defined in an ACL to the remote device. The ACL defines the headquarters subnets that branches can access. Traffic not destined for the subnets specified in the ACL is directly forwarded to the Internet. Such traffic does not pass through the IPsec tunnel.

 **NOTE**

In the Network-auto-cfg mode, delivering of parameters defined in the ACL is not supported.

Procedure

Step 1 Set basic parameters and optional parameters on the remote device.

1. Run **system-view**

The system view is displayed.

2. Create an Efficient VPN policy and determine whether to reference an ACL based on the Efficient VPN mode.
 - Create an IPsec Efficient VPN policy in client mode.

Run **ipsec efficient-vpn** *efficient-vpn-name* [**mode client**]

An IPsec Efficient VPN policy in client mode is created and the IPsec Efficient VPN policy view is displayed.

By default, no IPsec Efficient VPN policy is created in the system.

The remote device in client mode applies to the headquarters for an IP address to establish an IPsec tunnel with the Efficient VPN server. The source address in packets sent from the branch to the headquarters is the requested IP address, so ACLs are not required.
 - Create an IPsec Efficient VPN policy in network-auto-cfg mode.

Run **ipsec efficient-vpn** *efficient-vpn-name* [**mode network-auto-cfg**]

An IPsec Efficient VPN policy in network-auto-cfg mode is created and the IPsec Efficient VPN policy view is displayed.

By default, no IPsec Efficient VPN policy is created in the system.

The network-auto-cfg mode is supported in IKEv1 only.
 - Create an IPsec Efficient VPN policy in network or network-plus mode and reference an ACL.
 - i. Run **ipsec efficient-vpn** *efficient-vpn-name* [**mode { network | network-plus }**]An IPsec Efficient VPN policy in network or network-plus mode is created and the IPsec Efficient VPN policy view is displayed.

By default, no IPsec Efficient VPN policy is created in the system.

 - ii. Run **security acl** *acl-number*An ACL is referenced in the IPsec Efficient VPN policy.

By default, no ACL is referenced.

acl-number is an advanced ACL that has been created.

If an ACL is referenced, the rule can only match IP packets, that is, **permit ip**.
3. Run **remote-address** { *ip-address* | **host-name** *host-name* } { **v1** | **v2** }
- A peer address or a domain name in IKE negotiation is configured.
- By default, no IP address or domain name is configured for the remote IKE peer during IKE negotiation.
- You can configure a maximum of two IP addresses or two domain names in the same view.
- To improve network reliability, two devices can be deployed at the headquarters to connect to the branch gateway. In an IPsec policy, two IP addresses or domain names of the remote IKE peer can be configured on the branch gateway. The branch gateway first attempts to use the first configured IP address or domain name to establish an IKE connection with the headquarters gateway. If establishing an IKE connection fails, the branch gateway uses the second IP address or domain name to establish an IKE connection.
4. Configure an authentication key according to the authentication mode in the IKE proposal.

 **NOTE**

The system uses the pre-shared key authentication by default. The remote device and Efficient VPN server select the authentication mode using the **authentication-method** command.

- If pre-shared key authentication is used, configure a pre-shared key.

Run **pre-shared-key** { **simple** | **cipher** } *key*

A pre-shared key is configured.

By default, no pre-shared key is configured on IKE peers.

The pre-shared key at the two ends must be the same.

If **simple** is used, passwords are saved in the configuration file in plain text, resulting in security risks. Therefore, **cipher** is recommended to save passwords in cipher text.

- When RSA signature authentication is used, obtain a digital signature.

i. Run **pki realm** *realm-name*

A PKI domain that the digital signature in the Efficient VPN policy belongs to is specified. The system obtains the local CA certificate and device certificate according to the PKI configuration.

By default, no PKI domain is bound to an IKE peer or an Efficient VPN policy.

realm-name specifies a PKI domain that has been created using the **pki realm** command.

ii. (Optional) Run **inband oosp**

The Online Certificate Status Protocol (OCSP) is enabled.

By default, IKEv2 is not used for OCSP requests and responses.

This command takes effect only when the **certificate-check** command with **oosp** specified is executed in the created PKI domain.

iii. (Optional) Run **inband crl**

The device is configured to use IKEv2 to transmit Certificate Revocation List (CRL) requests and responses during digital certificate authentication.

By default, CRL requests and responses are not transmitted using IKEv2.

This command takes effect only when **crl** is specified in the **certificate-check** command configured in the created PKI domain.

5. Run **dh** { **group1** | **group2** | **group5** | **group14** | **group19** | **group20** | **group21** }

A Diffie-Hellman group used in IKE negotiation is configured.

By default, group14 is used in IKE negotiation.

The security levels of the following Diffie-Hellman groups are in descending order of priority: group21 > group20 > group19 > group14 > group5 > group2 > group1.

You are advised not to use group1, group2, or group5; otherwise, security defense requirements may be not met.

6. (Optional) Set optional parameters.

- Run **authentication-method** { **pre-share** | **rsa-signature** }

An authentication method is specified for an IKE proposal.

By default, pre-shared key authentication is used.

- Run **local-id-type** { **dn** | **ip** | **key-id** | **fqdn** | **user-fqdn** }
The local ID type used in IKE negotiation is set.
By default, the IP address of the local end is used as the local ID.
When the device functions as the remote end to communicate with a Cisco device in the Efficient VPN policy, you need to specify the **key-id** parameter in the command. Meanwhile, you also need to run the **service-scheme** command to specify the service scheme that the Cisco device uses.
- Run **service-scheme** *service-scheme-name*
A server-end service scheme is configured in an Efficient VPN policy.
By default, no server-end service scheme is configured in an Efficient VPN policy.
If an AAA service scheme is configured in the Efficient VPN policy, you need to specify the AAA service scheme configured on the server before the server can authorize the remote device. Meanwhile, you also need to specify the **key-id** parameter in the **local-id-type** command. If the **key-id** parameter is not specified, the configuration does not take effect. If authorization is performed using the service scheme used on the server, this step is not required.
If the **aaa authorization** command is configured on the server to enable AAA RADIUS server authorization, run the **service-scheme** command to specify the AAA domain configured on the server.
- Run **sim-based-username type** { **imei** | **imsi** } **password** *password*
The type of the user name used by the remote device to be authenticated by the RADIUS server is configured.
By default, the type of the user name used by the remote device to be authenticated by the RADIUS server is not configured.
The configuration of this step takes effect in the network-auto-cfg mode only.
- Run **dpd msg** { **seq-hash-notify** | **seq-notify-hash** }
The sequence of the payload in DPD packets is set.
By default, the sequence of the payload in DPD packets is **notify-hash**.
The two ends must use the same sequence of the payload in DPD packets; otherwise, DPD is invalid.
- Run **tunnel local** { *ip-address* | **applied-interface** }
A local IP address is configured.
By default, the local IP address is not configured.
Generally, you do not need to configure a local IP address for an IPsec policy established in IKE negotiation mode. During SA negotiation, the device selects the local IP address according to a route.
 - If the IP address of an interface bound to an IPsec policy is variable or unknown, run the **tunnel local** *ip-address* command to specify the IP address of another interface such as a loopback interface as the local IP address or run the **tunnel local** **applied-interface** command to specify an interface IP address as the local IP address.
 - If an interface bound to an IPsec policy is configured with one primary IP address and multiple secondary IP addresses, run the **tunnel local** *ip-address* command to specify one IP address as the local IP address or run the **tunnel local** **applied-interface** command to specify the primary IP address of the interface as the local IP address.

- If the local and remote ends have equal-cost routes, run the **tunnel local** { *ip-address* | **applied-interface** } command to specify the local IP address so that IPsec packets can be sent out from the specified interface.
 - Run **remote-id** *id*
The remote ID for IKE negotiation is configured.
By default, the remote ID for IKE negotiation is not configured.
 - Run **sa binding vpn-instance** *vpn-instance-name*
A VPN instance is bound to an IPsec tunnel.
This command specifies the VPN that the remote end of the IPsec tunnel belongs to. The tunnel initiator then can obtain the outbound interface and send packets through the outbound interface.
 - Run **qos group** *qos-group-value*
The QoS group to which the IPsec packets belong is set.
By default, no QoS group to which the IPsec packets belong is set.
 - Run **qos pre-classify**
Pre-extraction of original IP packets is enabled.
By default, pre-extraction of original IP packets is disabled.
 - Run **pfs** { **dh-group1** | **dh-group2** | **dh-group5** | **dh-group14** | **dh-group19** | **dh-group20** | **dh-group21** }
The device is configured to use Perfect Forward Secrecy (PFS) in IPsec negotiation.
By default, PFS is not used in IPsec negotiation.
 - Run **anti-replay window** *window-size*
The IPsec anti-replay window size is set.
By default, the IPsec anti-replay window size is 1024 bits.
7. Apply the Efficient VPN policy to an interface.
- a. Run **quit**
Return to the system view.
 - b. Run **interface** *interface-type interface-number*
The interface view is displayed.
 - c. Run **ipsec efficient-vpn** *efficient-vpn-name*
The Efficient VPN policy is applied to the interface.

You can bind only one Efficient VPN policy to the remote device in a scenario except that the remote device has multiple egress links.

Step 2 (Optional) Set optional parameters in the system view on the remote device.

- [6.7.8 \(Optional\) Configuring IPsec Check](#)
- [6.7.6 \(Optional\) Configuring IPsec Fragmentation Before Encryption](#)
- [Set the global IPsec SA lifetime.](#)
- [6.7.5 \(Optional\) Enabling the Anti-replay Function](#)

----End

6.9.2 Configuring the Efficient VPN Server

Context

Parameters on the Efficient VPN server include network resource parameters and IPsec parameters:

1. Network resource parameters include the IP address, domain name, DNS server address, and WINS server address. The Efficient VPN server can deliver network resource parameters to the remote device over the IPsec tunnel.
2. An SA must be set up through an IPsec policy template. There are limitations on other IPsec parameters.

Procedure

Step 1 (Optional) Set network resource parameters on the Efficient VPN server.

1. (Optional) Configure a global address pool and deliver the IP address used to establish an IPsec tunnel to the remote device.

 **NOTE**

If the Efficient VPN policy on the remote device uses the client, network-plus, or network-auto-cfg mode, the Efficient VPN server must deliver the IP address.

- a. Run **system-view**
The system view is displayed.
 - b. Run **ip pool ip-pool-name**
A global address pool is created.
 - c. Run **network ip-address [mask { mask | mask-length }]**
An allocatable network segment address is specified for the global address pool.
 - d. Run **gateway-list ip-address &<1-8>**
An egress gateway address is configured for the global address pool.
2. Configure resources to be delivered in the service scheme view.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **aaa**
The AAA view is displayed.
 - c. Run **service-scheme service-scheme-name**
A service scheme is created and the service scheme view is displayed.
 - d. (Optional) Run **ip-pool pool-name [move-to new-position]**
An IP address pool is configured.
pool-name specifies the global address pool configured in step a.
 - e. (Optional) Run **auto-update url url-string version version-number**
The URL and version number are configured.
The remote device can download the version file, patch file, and configuration file through the URL and branch devices can be upgraded automatically.

- f. (Optional) Run **dns-name** *domain-name*
A DNS domain name is configured.
- g. (Optional) Configure DNS server IP addresses and WINS server IP addresses.
 - i. Run **dns ip-address**
The IP address of the primary DNS server is configured.
 - ii. (Optional) Run **dns ip-address secondary**
The IP address of the secondary DNS server is configured.
 - iii. Run **wins ip-address**
The IP address of the primary WINS server is configured.
 - iv. (Optional) Run **wins ip-address secondary**
The IP address of the secondary WINS server is configured.

Step 2 Set IPsec parameters on the Efficient VPN server.

1. Run **system-view**

The system view is displayed.

2. Configure an IPsec proposal.

 **NOTE**

- **encapsulation-mode** must be set to **tunnel** to establish an IPsec tunnel using an Efficient VPN policy.
- When IKEv1 is used, IPsec supports non-authentication and non-encryption. When IKEv2 is used, IPsec does not support non-authentication or non-encryption.
- The Efficient VPN policy supports only ESP.

For details on how to configure an IPsec proposal, see [6.7.2 Configuring an IPsec Proposal](#).

3. Configure an IKE proposal.

 **NOTE**

- The encryption algorithm used during IKEv1 negotiation must be **3des-cbc**, and the authentication algorithm must be **md5**, **sha1** or **sha2**. **3des-cbc**, **md5**, and **sha1** are insecure. Exercise caution when you use **3des-cbc**, **md5**, or **sha1**. **md5** is recommended.

For details on how to configure an IKE proposal, see [6.10.1 Configuring an IKE Proposal](#).

4. Configure an IKE peer.

 **NOTE**

- When IKEv1 is used, **exchange-mode** must be set to **aggressive**.
- You can run the **resource acl *acl-number*** command in IKEv1 to implement ACL delivery.
ACL delivering is not supported in the Network-auto-cfg mode.
- Run the **service-scheme** command to bind the IKE peer to the AAA service scheme so that network resources including the IP address, domain name, DNS server IP addresses, and WINS server IP addresses can be delivered.
- In the Efficient VPN policy, run the **aaa authorization [domain *domain-name*]** command on an IKE peer to enable AAA RADIUS server authorization.
If the **domain** parameter is specified, the remote device obtains authorization information using the specified **domain**. If the **domain** parameter is not specified, the remote device obtains authorization information using the domain name it sends to the server. The domain name is specified using the **service-scheme** command in the Efficient VPN policy view.
If the **aaa authorization** command is configured on the IKE peer, the **service-scheme** command configured on the server does not take effect.

For details on how to configure an IKE peer, see [6.10.2 Configuring an IKE Peer](#).

5. Configure an IPsec policy using an IPsec policy template.

For details on how to configure an IPsec policy using an IPsec policy template, see [6.7.3.3 Configuring an IPsec Policy Using an IPsec Policy Template](#).

6. (Optional) Configure the following extensions.
 - [6.7.8 \(Optional\) Configuring IPsec Check](#)
 - [6.7.6 \(Optional\) Configuring IPsec Fragmentation Before Encryption](#)
 - [Set the global IPsec SA lifetime](#)
 - [Enabling the Anti-replay Function](#)
 - [Allowing New Users with the Same Traffic Rule as Original Branch Users to Access the Headquarters Network](#)
7. Apply an IPsec policy group to an interface.

For details on how to apply an IPsec policy group to an interface, see [6.7.14 Applying an IPsec Policy Group to an Interface](#).

----End

6.9.3 Verifying the Efficient VPN Configuration

Prerequisites

The Efficient VPN configurations are complete.

Procedure

- Run the **display ipsec proposal [brief | name *proposal-name*]** or **display ipsec proposal [brief | name *proposal-name*] ctrl-plane** command to check IPsec proposal information.
- Run the **display ipsec efficient-vpn [brief | capability | name *efficient-vpn-name* | remote]** command to check Efficient VPN policy information.
- Run the **display ipsec sa [brief | duration | efficient-vpn *efficient-vpn-name* | policy *policy-name* [*seq-number*] | profile *profile-name* | remote *ip-address*]** command to check IPsec SA information.

- Run the **display ipsec interface brief** command to check information about the IPsec policies applied to interfaces.
- Check IKE information. See [6.10.13 Verifying the IKE Configuration](#).

----End

6.10 Configuring IKE

IKE provides key negotiation and SA establishment to simplify IPsec use and management.

Pre-configuration Tasks

Before configuring IKE, complete the following tasks:

- Determine parameters in an IKE proposal.
- Determine the PKI domain that the IKE peer belongs to if RSA signature authentication is used.

NOTE

For details on how to configure PKI, see *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - Security*.

Configuration Process

Configure matching parameters at the two ends to complete IKE negotiation.

6.10.1 Configuring an IKE Proposal

Context

An IKE proposal defines parameters used in IKE negotiation, including the encryption algorithm, authentication mode and algorithm, Diffie-Hellman group, and SA lifetime.

During IKE negotiation, the initiator sends its IKE proposal to the responder, and the responder searches for its own matching IKE proposal. The responder first searches for the IKE proposal with the lowest sequence number and proceeds in ascending order of sequence number until a matching IKE proposal is found. The matching IKE proposal will be used to establish a secure tunnel.

The priority of an IKE proposal is represented by its sequence number. A smaller sequence number indicates a higher priority of an IKE proposal. You can create multiple IKE proposals with different priorities. The two ends must have at least one matching IKE proposal for successful IKE negotiation.

Two matching IKE proposals define the same encryption algorithm, authentication mode, authentication algorithm, and Diffie-Hellman group. A smaller SA lifetime at the two sides is used.

NOTE

By default, there is an IKE proposal that has the lowest priority and uses default parameter settings. If only the sequence number is specified during IKE proposal creation, this IKE proposal also uses default parameter settings.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike proposal *proposal-number***

An IKE proposal is created and the IKE proposal view is displayed.

Step 3 Run **authentication-method { *pre-share* | *rsa-signature* }**

An authentication mode is specified for the IKE proposal.

By default, an IKE proposal uses pre-shared key authentication.

The authentication methods in the IKE proposals used by the IKE peer must be the same. Otherwise, IKE negotiation fails.

NOTE

When IKEv2 is used and the authentication mode is modified dynamically, you need to run the **re-authentication interval** command to configure re-authentication to make the modification take effect.

Step 4 Run **authentication-algorithm { *md5* | *sha1* | *sha2-256* | *sha2-384* | *sha2-512* }**

An authentication algorithm is specified for IKEv1 negotiation.

By default, an IKE proposal uses the SHA2-256 algorithm.

The security levels of the following authentication algorithms are in descending order of priority: SHA2-512 > SHA2-384 > SHA2-256 > SHA1 > MD5.

MD5 and SHA1 algorithms are not recommended because they cannot meet your security defense requirements.

NOTE

In IKEv1 certificate negotiation, if the authentication algorithm **sha2-512** is configured, the RSA key length must be greater than 1024.

Step 5 Run **encryption-algorithm { *des* | *3des* | *aes-128* | *aes-192* | *aes-256* }**

An encryption algorithm is specified for the IKE proposal.

By default, an IKE proposal uses AES-256.

The security levels of the following encryption algorithms are in descending order of priority: AES-256 > AES-192 > AES-128 > 3DES > DES.

DES and 3DES algorithms are not recommended because they cannot meet your security defense requirements.

Step 6 Run **dh { *group1* | *group2* | *group5* | *group14* | *group19* | *group20* | *group21* }**

A Diffie-Hellman group used in IKE negotiation is configured.

By default, group14 is used in IKE negotiation.

The security levels of the following Diffie-Hellman groups are in descending order of priority: group21 > group20 > group19 > group14 > group5 > group2 > group1.

You are advised not to use group1, group2, or group5; otherwise, security defense requirements may be not met.

Step 7 Run **prf { aes-xcbc-128 | hmac-md5 | hmac-sha1 | hmac-sha2-256 | hmac-sha2-384 | hmac-sha2-512 }**

An algorithm is configured to generate a pseudo random number.

By default, the HMAC-SHA2-256 algorithm is used.

The HMAC-MD5 and HMAC-SHA1 algorithms are not recommended because they cannot meet your security defense requirements.

 **NOTE**

Only IKEv2 requires the PRF algorithm.

Step 8 (Optional) Run **integrity-algorithm { aes-xcbc-96 | hmac-md5-96 | hmac-sha1-96 | hmac-sha2-256 | hmac-sha2-384 | hmac-sha2-512 }**

The integrity algorithm used in IKEv2 negotiation is specified.

By default, the integrity algorithm HMAC-SHA2-256 is used in IKEv2 negotiation.

---End

6.10.2 Configuring an IKE Peer

Context

When an IPsec tunnel is established in IKE negotiation mode, you need to reference an IKE peer and configure attributes between IKE peers during IKE negotiation. When configuring attributes between IKE peers, note the following points:

- The IKE peers must use the same IKE version.
- The IKE peers that use IKEv1 must use the same negotiation mode.
- Identity authentication parameters of the IKE peers must match.

IKE has two versions: IKEv1 and IKEv2. The differences between IKEv1 and IKEv2 are as follows:

- IKEv1 requires the phase 1 negotiation mode, whereas IKEv2 does not.
- IKEv1 does not support the Online Certificate Status Protocol (OCSP) for an IKE peer, whereas IKEv2 supports.
- IKEv2 supports the re-authentication interval to enhance security, whereas IKEv1 does not support the re-authentication interval.

Pre-configuration Tasks

Before configuring an IKE peer, complete the following tasks:

- Import the local certificate and CA root certificate on the authenticated end and import the CA root certificate on the authenticating end if RSA signature authentication is used.
- Generate the RSA key pair on the authenticated end if RSA key authentication is used.

In IKEv1 certificate negotiation, if the authentication algorithm **sha2-512** is configured, the RSA key length must be greater than 1024.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike peer peer-name**

An IKE peer is created and the IKE peer view is displayed.

By default, no IKE peer is created in the system.

Step 3 Run **version { 1 | 2 }**

The IKE protocol version used by IKE peers is configured.

By default, IKE peers support IKEv1 and IKEv2.

Step 4 (Optional) Run **exchange-mode { main | aggressive }**

IKEv1 phase 1 negotiation mode is set.

By default, the main mode is used in IKEv1 phase 1.

- Main mode: protects identities.
- Aggressive mode: provides faster negotiation speed, but cannot protect identities.

Step 5 (Optional) Run **local-address ipv4-address**

The local IP address in IKE negotiation is configured.

By default, the system selects an outbound interface according to a route and uses the IP address of the outbound interface as the local IP address.

Generally, you do not need to configure the local IP address.

- If the IP address of an interface bound to an IPsec policy is variable or unknown, run the **local-address ipv4-address** command to specify the IP address of another interface such as a loopback interface as the local IP address.
- If an interface bound to an IPsec policy is configured with one primary IP address and multiple secondary IP addresses, run the **local-address ipv4-address** command to specify one IP address as the local IP address.
- If the local and remote ends have equal-cost routes, run the **local-address ipv4-address** command to specify the local IP address so that IPsec packets can be sent out from the specified interface.

Step 6 (Optional) Run **remote-address { [vpn-instance vpn-instance-name] { ipv4-address | host-name host-name } | authentication-address start-ipv4-address [end-ipv4-address] }**

The remote IP address or domain name used in IKE negotiation is configured.

By default, an IKE peer has no domain name, remote IP address, or remote IP address range configured.

If a VPN instance has been configured on the interface applied with an IPsec policy, the *vpn-instance-name* parameter needs to be configured. After this parameter is configured, the device searches for a route to the remote IP address in the specified VPN during IPsec tunnel negotiation. If more than one remote IP address or domain name is configured, the specified *vpn-instance-name* must be the same.

If the remote device has a variable IP address and a fixed domain name, you must specify *host-name* to configure the remote domain name. In this situation, the remote end must have Dynamic Domain Name System (DDNS) configured to bind domain names to dynamic IP addresses, and the local end must have DNS configured for domain name resolution.

You can specify **authentication-address** to configure the pre-NAT IP address as the remote authentication address when the following conditions are met:

- Two devices use IKEv2.
- The remote device uses the internal IP address.
- Packets traverse the NAT device.
- The IP address is used for authentication.

The post-NAT IP address needs to be used as the remote address in this situation.

Step 7 Run **ike-proposal** *proposal-number*

An IKE proposal is referenced.

The IKE proposal must have been created.

By default, an IKE peer does not reference an IKE proposal

Step 8 Configure identity authentication parameters.

Identity authentication parameters are different in different authentication modes. Configure identity authentication parameters based on the authentication mode defined in an IKE proposal.

- Pre-shared key authentication
 - a. Configure a pre-shared key for pre-shared key authentication.

You can use the following methods to configure the pre-shared key. When both of the two methods are used, the pre-shared key configured in the IKE user table is preferred.

 - A single IKE peer or multiple IKE peers use the same ID and pre-shared key.

Run the **pre-shared-key** { **simple** | **cipher** } *key* command to configure a pre-shared key.

The pre-shared key at the two ends must be the same.
 - Multiple IKE peers use different IDs and pre-shared keys.

NOTE

- In a point-to-multipoint scenario, the device functions as the VPN gateway of the headquarters, an IPsec policy is created using an IPsec policy template, and the VPN gateway receives IPsec connection setup requests of different branches. When the pre-shared key is used for identity authentication and all branches use the same ID and pre-shared key, there are security risks. That is, if the ID and pre-shared key of one branch leak, the ID and pre-shared key of all branches leak. The IKE user table can prevent this problem.
- When the IKE user table is used, you do not need to use the **remote-id-type** command subsequently.

1) Run **quit**

Return to the system view.

2) Run **ike user-table** *user-table-id*

An IKE user table is created and its view is displayed, or the view of an existing IKE user table is displayed directly.

- 3) Run **user** *user-name*
 An IKE user is created and its view is displayed, or the view of an existing IKE user is displayed directly.
- 4) Run **pre-shared-key** *key*
 The pre-shared key used by IKE peers is configured when IKE peers use pre-shared key authentication during IKE negotiation.
 By default, the pre-shared key used by IKE peers is not configured when IKE peers use pre-shared key authentication during IKE negotiation.
- 5) Run **id-type** { **any** *any-id* | **fqdn** *remote-fqdn* | **ip** *ipv4-address* | **user-fqdn** *remote-user-fqdn* }
 The IKE user ID type and ID are configured.
 By default, the IKE user ID type and ID are not configured.
 The value of **id-type** is the remote ID.
 When IKEv1 in main mode is used, the value of **id-type** must be set to **ip**. In NAT traversal scenarios, *ipv4-address* should be set to the IP address that is translated using NAT.
- 6) (Optional) Run **description** *description*
 The description of an IKE user is configured.
 By default, the description of an IKE user is not configured.
- 7) Run **quit**
 Return to the IKE user table view.
- 8) Run **quit**
 Return to the system view.
- 9) Run **ike peer** *peer-name*
 The IKE peer view is displayed.
- 10) Run **user-table** *user-table-id*
 An IKE user table is reference in the IKE peer.

b. Configure the ID type and ID value.

For pre-shared key authentication, the local ID type or local ID on the local end must be the same as the remote ID type or remote ID on the remote end. Configure the ID type and ID value according to [Table 6-9](#).

Table 6-9 Relationship between the local ID type, local ID, remote ID type, and remote ID

Local ID Type	Local ID	Remote ID Type	Remote ID
FQDN	Local name: local-id device	FQDN	remote-id device
IP	Local IP address: 10.1.1.3	IP	remote-address 10.1.1.3

Local ID Type	Local ID	Remote ID Type	Remote ID
User-FQDN	Local user domain name: local-id devicea@example.com	User-FQDN	remote-id devicea@example.com

i. Run **local-id-type { fqdn | ip | user-fqdn }**

The local ID type used in IKE negotiation is set.

By default, the IP address of the local end is used as the local ID.

ii. (Optional) Run **local-id id**

The local ID used in IKE negotiation is set.

You need to configure the local ID when the ID type of an IKE peer is FQDN or User-FQDN.

 **NOTE**

- When the local ID type is **IP**, you do not need to perform this step. In this case, the device uses the IP address of the interface for establishing an IPsec tunnel as the local ID by default. If the interface has multiple IP addresses, for example, primary and secondary IP addresses are configured, the device uses the IP address configured by the **tunnel local** command as the local ID.
- You can also run the **ike local-name local-name** command in the system view to configure the local ID during IKE negotiation. Then all IKE peers of the device use this local ID for identity authentication.
- The local ID configured by the **local-id** command takes precedence over the local ID configured by the **ike local-name** command.

iii. Run **remote-id-type { any | fqdn | ip | user-fqdn | none }**

The remote ID type used in IKE negotiation is set.

By default, no remote ID type is set.

iv. (Optional) Run **remote-id id**

The remote ID used in IKE negotiation is set.

When the remote ID type is **IP**, the value configured by the **remote-address** command is used as the remote ID by default, regardless of whether the **remote-id** command is configured.

● RSA signature authentication

a. Configure the ID type and ID value.

For RSA signature authentication, the remote ID type or remote ID on the local end must be consistent with corresponding fields in the local certificate on the remote end. Configure the ID type and ID value according to [Table 6-10](#).

Table 6-10 Relationship between the local ID type, local ID, remote ID type, and remote ID

Local ID Type	Local ID	Remote ID Type	Remote ID
DN	Subject: CN=devicea	DN	remote-id / CN=devicea
FQDN	DNS:devicea.example.com	FQDN	remote-id devicea.example.com
IP	Local IP address: 10.1.1.3	IP	remote-address 10.1.1.3
User-FQDN	email: devicea@example.com	User-FQDN	remote-id devicea@example.com

- i. Run **rsa signature-padding { pkcs1 | pss }**
 A padding mode is configured for the RSA signature.
 By default, the padding mode of the RSA signature is PKCS1.
- ii. Run **pki realm realm-name**
 The PKI domain that the digital certificate of the IKE peer belongs to is specified. The local digital certificate is obtained based on the configuration of the PKI domain.
 If the authentication mode is RSA signature, the local certificate contains the IP address, DN, name, and User-FQDN information about the local end.
- iii. (Optional) Run **local-id-reflect enable**
 During IKEv2 negotiation, the local ID of the responder is used as the remote ID carried in the IKE packets sent by the initiator.
 By default, this function is disabled.
 During IKEv2 negotiation, if the user does not know the remote ID configured for the initiator, you can perform this step on the responder. When the responder receives an IKE packet from the initiator, the responder uses the IDr payload (remote ID) in the received packet as its local ID. If the responder does not obtain the IDr payload, it obtains its local ID based on the local configuration.
 Currently, the ID type can only be IP address, FQDN, or User-FQDN.
 When both the **local-id-reflect enable** and **local-id-preference certificate enable** commands are configured, the **local-id-reflect enable** command takes effect.
- iv. (Optional) Run **local-id-preference certificate enable**
 The device is enabled to preferentially obtain the local ID from a field in a certificate when IKE uses certificate negotiation.
 By default, the device does not preferentially obtain the local ID from a field in a certificate when IKE uses certificate negotiation.

When IKE uses certificate negotiation, the device can obtain its local ID from a field (IP address, FQDN, or email address) in the certificate, removing the need to configure the local ID.

After this command is configured, the device preferentially obtains its local ID from a field in the certificate. If this method fails, it obtains its local ID based on the local configuration. If this method also fails, IKE negotiation fails.

v. Run **local-id-type** { **dn** | **fqdn** | **ip** | **user-fqdn** }

The ID type is configured.

By default, the IP address of the local end is used as the local ID.

The specified ID type must be the same as that displayed in the **display pki certificate** command output.

vi. (Optional) Run **local-id** *id*

The local ID used in IKE negotiation is set.

You need to configure the local ID when the ID type of an IKE peer is FQDN or User-FQDN.

vii. Run **remote-id-type** { **any** | **dn** | **fqdn** | **ip** | **user-fqdn** | **none** }

The remote ID type used in IKE negotiation is set.

By default, no remote ID type is set.

The local and remote ID types of an IKE peer must be the same.

viii. (Optional) Run **remote-id** *id*

The remote ID used in IKE negotiation is set.

If the remote ID type is IP, you do not need to configure the **remote-id** command. In this case, the device uses the value specified by the **remote-address** command as the remote ID by default.

 **NOTE**

If the local ID type is FQDN or User-FQDN, the remote end uses the **remote-id** specified in the IKE peer as the remote ID for IKEv1 negotiation. However, for IKEv2 negotiation, the remote end preferentially uses the value of the DNS (corresponding to FQDN) or email (corresponding to User-FQDN) field in the certificate. If the fields are unavailable, the remote end uses the **remote-id** specified in the IKE peer for negotiation.

ix. (Optional) Run **inband ocs**

The device is configured to validate the remote certificate based on the OCS validation result sent from the remote device when IKEv2 uses RSA signature authentication.

By default, the device does not validate the remote certificate based on the OCS validation result sent from the remote device when IKEv2 uses RSA signature authentication.

x. (Optional) Run **inband c**

The device is configured to validate the remote certificate based on the CRL sent from the remote device when IKEv2 uses RSA signature authentication.

By default, the device does not validate the remote certificate based on the CRL sent from the remote device when IKEv2 uses RSA signature authentication.

If both the **inband ocs** and **inband c** commands are run, the certificate is considered valid only when the certificate verifications in OCS and CRL modes are passed.

xi. (Optional) Run **ikev2 id-match-certificate enable**

The device is enabled to check certificate identity information of the remote device during IKEv2 certificate negotiation.

By default, the device does not check certificate identity information of the remote device during IKEv2 certificate negotiation.

After this command is configured, the local device only checks the Subject field, IP address, FQDN, or email of the remote device. If the information differs from the ID (DN, IP address, FQDN, or User-FQDN) of the remote device, IKEv2 negotiation fails.

xii. (Optional) Run **certificate-request empty-payload enable**

The certificate request payload is empty.

By default, certificate request payloads carry CA information.

When a template-based IPsec policy is configured for the Router in headquarters and certificate-based authentication is used, you can run the **certificate-request empty-payload enable** command to empty certificate request payloads so that users who use different CA certificates in branch offices can access the Router. Based on the certificate information about branch offices, the Router obtains certificates from associated certificate domains for authentication.

If the access devices cannot process certificate request packets with an empty authentication and authorization field, do not configure this command. Otherwise, tunnel negotiation fails.

Step 9 (Optional) Run **lifetime-notification-message enable**

The device is enabled to send IKE SA lifetime notification messages.

By default, the device does not send IKE SA lifetime notification messages.

IKEv1 peers negotiate the lifetime, and a smaller SA lifetime at the two ends is used. When a Huawei device and a non-Huawei device establish an IPsec tunnel and they use different IKE SA lifetimes, run this command to enable the device to send IKE SA lifetime notification messages. By doing this, two ends can successfully perform IKE negotiation.

Step 10 (Optional) Run **re-authentication interval** *interval*

The IKEv2 re-authentication interval is set.

By default, IKEv2 does not perform re-authentication.

In remote access, IPsec peers periodically send re-authentication packets, which reduces potential risks of attacks and improves IPsec network security.

----End

6.10.3 (Optional) Setting the IKE SA Lifetime

Context

After the SA lifetime is set, SAs are updated in real time and difficult to decipher, enhancing security.

The IKE SA lifetime is classified as follows:

- **Hard lifetime (hard timeout period):** specifies the lifetime of an IKE SA.
When two devices negotiate an IKE SA, the actual hard lifetime is the smaller of the two values configured on the two devices.
- **Soft lifetime (soft timeout period):** refers to the time after which a new IKE SA is negotiated so that the new IKE SA will be ready before the hard lifetime of the original IKE SA expires.

Table 6-11 lists the default soft lifetime values.

Table 6-11 Soft lifetime values

IKE Protocol Type	Description
IKEv1	7/10 of the actual hard SA lifetime
IKEv2	7/10 of the actual hard SA lifetime

Before an IKE SA becomes invalid, IKE negotiates a new IKE SA for the remote end. The remote end uses the new IKE SA to protect IPsec communication immediately after the new IKE SA is negotiated. If service traffic is transmitted, the original IKE SA is deleted immediately. If no service traffic is transmitted, the original IKE SA will be deleted after 10s or the hard lifetime expires.

Changing the lifetime does not affect the established IKE SAs, and the changed value is used for establishing new IKE SAs in subsequent negotiation.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike proposal *proposal-number***

The IKE proposal view is displayed.

Step 3 Run **sa duration *time-value***

The IKE SA hard lifetime is set.

By default, the IKE SA lifetime is 86400s.

If the hard lifetime ends, IKE SAs are updated automatically. IKE negotiation involves Diffie-Hellman key calculation, which takes a long period of time. To ensure that IKE SA update does not affect secure communication, you are advised to set the lifetime to a value greater than 600s.

---End

6.10.4 (Optional) Configuring IKE Peer Status Detection

Context

IKE does not provide peer status detection. In IPsec communication, if one end becomes faulty, the other end may not detect the fault because of system failures and continues to send

IPsec packets to the faulty end. The problem can be solved only when the SA lifetime ends. Before the SA lifetime ends, the SA between IKE peers exists, causing traffic loss. Unreachability of an IKE peer can result in black holes where traffic is discarded. IPsec communication can be restored rapidly only when black holes are identified and detected in a timely manner.

The device provides heartbeat detection and dead peer detection (DPD) to detect the IKE peer status. Configure heartbeat detection or DPD as needed.

6.10.4.1 (Optional) Configuring Heartbeat Detection

Context

Heartbeat detection enables the local end to periodically send heartbeat packets to the remote end. If the local end does not receive heartbeat packets within the timeout interval, the local end considers the remote end as unreachable and deletes the IKE SA or IPsec SA between IKE peers.

There are limitations on heartbeat detection:

- Enabling heartbeat detection will consume CPU resources used to process IKE keepalive messages, so the number of established IPsec sessions is limited.
- There are no uniform standards, so devices from different vendors may fail to interwork.

The interval at which heartbeat packets are sent at the local end must be used with the timeout interval of heartbeat packets at the remote end. If the remote end does not receive any heartbeat packet within the timeout interval and the IKE SA carries a timeout tag, the IKE SA and its corresponding IPsec SA are deleted. If the IKE SA does not carry a timeout tag, it is marked as timeout.

NOTE

If IKE peers use IKEv1 during negotiation, the device supports heartbeat detection. If IKE peers use IKEv2 during negotiation, the device does not support heartbeat detection.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike heartbeat { seq-num { new | old } | spi-list }**

Parameters of heartbeat packets are set.

By default, a heartbeat packet uses old type sequence number mechanism and does not carry the SPI list.

Step 3 Run **ike heartbeat-timer interval interval**

The interval at which heartbeat packets are sent by an IKE SA is set.

By default, an IKE SA does not send heartbeat packets.

Step 4 Run **ike heartbeat-timer timeout seconds**

The timeout interval of heartbeat packets is set.

By default, the timeout interval during which an IKE SA waits for a heartbeat packet is not configured.

When **ike heartbeat-timer interval** is configured at one end, the **ike heartbeat-timer timeout** command must be used at the other end.

The timeout interval of heartbeat packets must be longer than the interval at which heartbeat packets are sent. Generally, packet loss does not occur for more than three consecutive times on a network. Therefore, it is recommended that the timeout interval of heartbeat packets be three times the interval at which heartbeat packets are sent.

---End

6.10.4.2 (Optional) Configuring DPD

Context

In IPsec communication, heartbeat detection technology detects faults at the remote end and prevents packet loss. However, periodically sending heartbeat messages consumes CPU resources at both ends and limits the number of established IPsec sessions.

Dead Peer Detection (DPD) technology sends DPD packets based on IPsec packets between IKE peers, and does not periodically send heartbeat packets. When the local end can receive IPsec traffic from the remote end, the local end considers the remote end as active. The local end sends DPD packets to detect the status of the remote end when the local end does not receive IPsec traffic from the remote end within a given period of time. If the local end does not receive response packets after sending DPD packets several times, the local end considers the remote end as unreachable and deletes the IKE SA or IPsec SA between IKE peers.

If heartbeat detection is used, the two ends periodically send heartbeat packets and settings at the two ends must match. If DPD is used, settings except the payload sequence in DPD packets at the two ends do not need to match. When IPsec packets are exchanged between IKE peers, DPD packets are not sent. DPD packets are sent only when one end does not receive IPsec packets from the other end in a period of time. This saves resources.

NOTE

When both heartbeat detection and DPD are used, DPD takes effect.

The detection mode and DPD are configured based on the **dpd type** or **ike dpd type** command. Two DPD modes are available:

- On-demand DPD
When the local end needs to send IPsec packets to the remote end, the local end determines that the DPD idle time is reached and sends a DPD request packet to the remote end.
- Periodic DPD
The local end determines that the DPD idle time is reached, and periodically sends a DPD request packet to the remote end based on the DPD idle time.

If the local end does not receive a DPD response packet from the remote end within the DPD packet retransmission interval, the local end retransmits the DPD request packet. If the local end still does not receive a DPD response packet after the DPD packet retransmission count is reached, the local end considers that the remote end goes offline, and deletes the IKE SA and IPsec SA.

Procedure

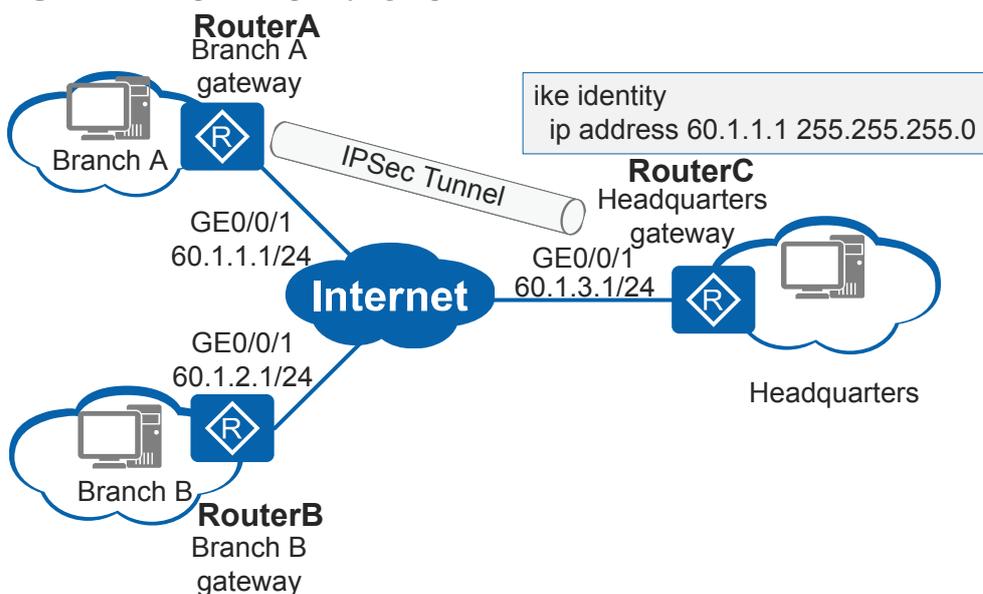
1. Run **system-view**
The system view is displayed.
2. Run **ike peer *peer-name***
The IKE peer view is displayed.
3. (Optional) Run **dpd msg { *seq-hash-notify* | *seq-notify-hash* }**
The sequence of the payload in DPD packets is configured.
By default, the sequence of the payload in DPD packets is **seq-notify-hash**.
The two ends must use the same sequence of the payload in DPD packets; otherwise, DPD does not take effect.
4. Run **dpd { *idle-time interval* | *retransmit-interval interval* | *retry-limit times* }**
The DPD idle time, DPD packet retransmission interval, and maximum number of DPD packet retransmissions are set.
By default, the DPD idle time is 30s, the DPD packet retransmission interval is 15s, and the maximum number of DPD packet retransmissions is 3.
5. Run **dpd type { *on-demand* | *periodic* }**
The on-demand or periodic DPD mode is configured.
By default, the DPD mode is not configured on an IKE peer.

6.10.5 (Optional) Configuring an Identity Filter Set

Context

- When a device functions as a responder during IKE negotiation, it can specify the peer allowed to connect to it to improve security.
An IPsec policy template or an IPsec profile specifies the peer based on the identity filter set.
As shown in [Figure 6-39](#), the headquarters gateway RouterC functions as the responder. An IPsec policy template is configured on RouterC, only RouterA matches the IP address in the identity filter set.

Figure 6-39 Responder specifying a qualified initiator

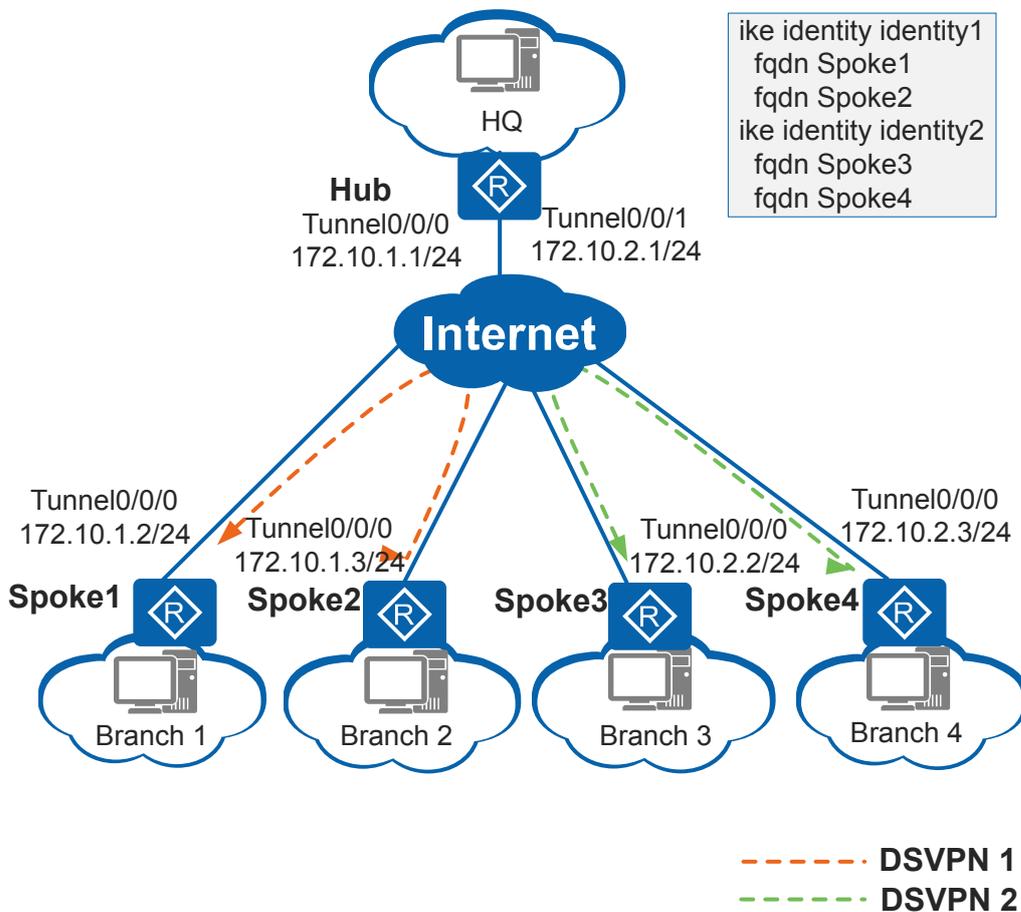


- In an IPsec over DSVPN application, multiple mGRE tunnel interfaces are configured on the hub which provides only one IP address for spoke access. The mGRE tunnel interfaces use the same source address or source interface. To solve this problem, set parameters in the identity filter set to specify the mGRE tunnel interface of each IKE packet.

For the detailed configuration of DSVPN, see [DSVPN Configuration](#).

In a DSVPN application as shown in [Figure 6-40](#), Spoke1 and Spoke2 belong to DSVPN 1, while Spoke3 and Spoke4 belong to DSVPN 2. Spoke1 wants to communicate with Spoke2, and Spoke3 wants to communicate with Spoke4. However, the hub provides only one public network IP address for spokes access. After you configure an identity filter set on the hub, it determines the mGRE tunnel interface of each spoke based on parameters in the identity filter set.

Figure 6-40 Responder specifying a qualified initiator in the IPsec over DSVPN application



Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike identity identity-name**

The identity filter set view is displayed.

NOTE

You can set one or more parameters in the identity filter set view, depending on the device specification.

Step 3 (Optional) Run **dn name**

The DN of an allowed peer for IKE negotiation is configured.

By default, no DN of allowed peer for IKE negotiation is configured.

Step 4 (Optional) Run **ip address ip-address { mask | mask-length }**

The IP address of an allowed peer for IKE negotiation is configured.

By default, no IP address of allowed peer for IKE negotiation is configured.

Step 5 (Optional) Run **fqdn** *fqdn-name*

The name of an allowed peer for IKE negotiation is configured.

By default, no name of allowed peer for IKE negotiation is configured.

Step 6 (Optional) Run **user-fqdn** *fqdn-name*

The domain name of an allowed peer for IKE negotiation is configured.

By default, no domain name of allowed peer for IKE negotiation is configured.

---End

Follow-up Procedure

Run the **match ike-identity** *identity-name* command in the ipsec policy template view or IPsec profile view to reference the identity filter set.

6.10.6 (Optional) Configuring DSCP Priority for IKE Packets

Context

IKE packets are used to negotiate IKE SAs and IPsec SAs or used for Deal Peer Detection (DPD). If IKE packets are lost during transmission, problems such as failures to negotiate IKE SAs and IPsec SAs may occur. As a result, packets cannot be protected by IPsec SAs. To solve these problems, ensure that network devices process IKE packets before service packets.

You can increase IKE packet priority by specifying the DSCP priority for IKE packets. This configuration ensures that IKE packets are processed in time on a busy network, improving transmission reliability of IKE packets.

You can configure the DSCP priority for IKE packets globally or on an IKE peer. The DSCP priority configured on an IKE peer takes precedence over that configured globally. When the DSCP priority is not configured on an IKE peer, the DSCP priority configured globally takes effect.

Procedure

- Configure the DSCP priority globally.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **ike dscp** *dscp-value*

The DSCP priority of IKE packets is configured globally.

By default, the global DSCP priority of IKE packets is 0.
- Configure the DSCP priority on an IKE peer.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **ike peer** *peer-name*

An IKE peer is created and the IKE peer view is displayed.

c. Run **dscp dscp-value**

The DSCP priority of IKE packets is configured on the IKE peer.

By default, the DSCP priority of IKE packets on a specified IKE peer is 0.

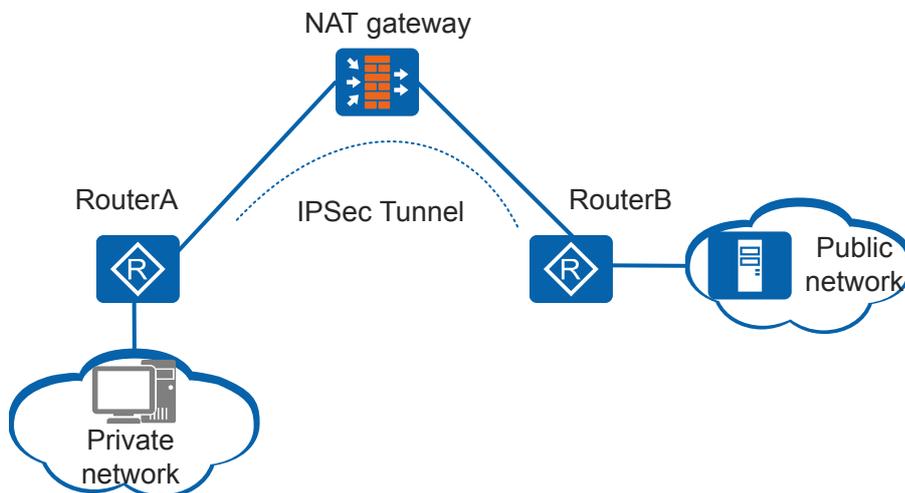
---End

6.10.7 (Optional) Configuring NAT Traversal

Context

During IPsec VPN deployment, the initiator on a private network may need to establish an IPsec tunnel with the responder on a public network. To ensure that an IPsec tunnel can be established when a network address translation (NAT) device exists, NAT traversal is required, as shown in [Figure 6-41](#).

Figure 6-41 NAT traversal in IPsec



Authentication Header (AH) integrity check involves the hash calculation for IP packets including the IP address, whereas NAT changes the IP address and causes the hash value change. Packets on the IPsec tunnel that has AH enabled cannot traverse a NAT device.

During Encapsulating Security Payload (ESP) integrity check, an outer IP header is not checked. If only address translation is performed, ESP can work properly. ESP is a Layer 3 protocol and does not support port setting, so there is also a problem in ESP when NAT port translation is used. To address this issue, NAT traversal encapsulates ESP packets with a UDP header. In transport mode, a standard UDP header is inserted between the original IP header and an ESP header during NAT traversal. In tunnel mode, a standard UDP header is inserted between the new IP header and an ESP header during NAT traversal. When ESP packets pass through a NAT device, the NAT device translates the IP address and port number of the outer IP header and inserted UDP header. After the translated packets reach the remote end of the IPsec tunnel, the remote end processes these packets in the same manner as IPsec packets.

If no IPsec packets are transmitted on an IPsec tunnel in a period of time, NAT session entries may be aged out and deleted because the NAT session entries have the keepalive time.

As a result, the IPsec tunnel cannot transmit IPsec packets between the NAT device and the IKE peer connected to the public network. To prevent NAT session entries from being aged, an IKE SA on the private network side of the NAT device sends NAT Keepalive packets to its remote end at an interval to maintain the NAT session.

 **NOTE**

If NAT traversal is enabled, the IPsec proposal referenced using the **ipsec proposal** command supports only ESP. AH authenticates the entire IP packet. Any modification in the IP header causes an AH check failure. Therefore, NAT traversal cannot be implemented on an IPsec tunnel protected by AH.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike peer peer-name**

An IKE peer is created and the IKE peer view is displayed.

Step 3 Run **nat traversal**

NAT traversal is enabled.

By default, the NAT traversal is enabled.

Step 4 Run **quit**

Return to the system view.

Step 5 Run **ipsec nat-traversal source-port port-number**

The UDP port number for IPsec NAT traversal is configured.

By default, the UDP port number for IPsec NAT traversal is 4500.

Step 6 Run **ike nat-keepalive-timer interval interval**

The interval for sending Keepalive packets is set.

By default, the interval for sending NAT Keepalive packets is 20 seconds.

---End

6.10.8 (Optional) Configuring IPsec VPN Multi-instance

Context

When multiple branches connected to the headquarters network across the Internet using IPsec, you can configure IPsec VPN Multi-instance, thereby isolating traffic of different branches.

You can use the following two modes to configure a VPN instance that IPsec tunnel traffic belongs to according to the IKE negotiation mode:

- Binding a VPN instance in SA mode
- Binding a VPN instance in IKE user mode

When a VPN instance is bound to traffic in SA mode, the device determines the VPN instance to which site traffic passing through the IPsec tunnel belongs by the user type, isolating traffic

from different sites. A VPN instance bound in SA mode has a higher priority than a VPN instance bound in IKE user mode.

 **NOTE**

The configuration takes effect only on the initiator of an IPsec tunnel. The initiator needs to obtain the outbound interface when sending packets. The packets received by the remote peer contain the VPN attribute, so the remote peer can still receive packets when no VPN is specified for it.

Procedure

- Binding a VPN instance in SA mode

- a. Run **system-view**

The system view is displayed.

- b. Run **ike peer** *peer-name*

An IKE peer is created and the IKE peer view is displayed.

- c. Run **sa binding vpn-instance** *vpn-instance-name*

A VPN instance that IPsec tunnel traffic belongs to is specified.

By default, a VPN instance that IPsec tunnel traffic belongs to is not configured.

The VPN instance has been created using the **ip vpn-instance** command and the route distinguisher (RD) has been configured for the VPN instance using the **route-distinguisher** command.

The specified VPN instance must be the same as the VPN instance bound to the ACL rule that is referenced by the [6.7.3 Configuring an IPsec Policy](#).

- Binding a VPN instance in IKE user mode

- a. Run **system-view**

The system view is displayed.

- b. Run **ike user-table** *user-table-id*

An IKE user table is created and its view is displayed, or the view of an existing IKE user table is displayed directly.

- c. Run **user** *user-name*

An IKE user is created and its view is displayed, or the view of an existing IKE user is displayed directly.

- d. Run **vpn-instance-traffic** { **public** | **name** *vpn-instance-name* }

A VPN instance corresponding to user traffic of the IKE user table is configured.

By default, the VPN instance corresponding to user traffic of the IKE user table is not configured.

The VPN instance has been created using the **ip vpn-instance** command and the route distinguisher (RD) has been configured for the VPN instance using the **route-distinguisher** command.

The specified VPN instance must be the same as the VPN instance bound to the ACL rule that is referenced by the [6.7.3 Configuring an IPsec Policy](#).

- e. Run **quit**

Return to the IKE user table view.

- f. Run **quit**

Return to the system view.

- g. Run **ike peer** *peer-name*
The IKE peer view is displayed.
- h. Run **user-table** *user-table-id*
An IKE user table is reference in the IKE peer.

----End

6.10.9 (Optional) Configuring Network Resource Delivery

Context

Efficient VPN uses the client/server model. It concentrates IPsec and other configurations on the Efficient VPN server (headquarters gateway). When basic parameters for establishing an SA are configured on the remote devices (branch gateways), the remote devices initiate a negotiation and establish an IPsec tunnel with the server. After IPsec tunnels are established, the Efficient VPN server allocates other IPsec attributes and network resources to the remote devices. Efficient VPN simplifies configurations and maintenance of IPsec and network resources of branches.

1. If an Efficient VPN policy in client or network-plus mode is used, the Efficient VPN server delivers an IP address to the remote device. The remote device then uses this IP address to establish an IPsec tunnel with the Efficient VPN server.
2. The Efficient VPN server delivers network resources including the DNS domain name, DNS server IP address, and WINS server IP address so that the branch can access server resources on the headquarters network.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike peer** *peer-name*

An IKE peer is created and the IKE peer view is displayed.

Step 3 Run **service-scheme** *service-scheme-name*

A service scheme is bound to the IKE peer.

By default, no service scheme is bound to an IKE peer.

service-scheme-name specifies a service scheme that has been created using the **service-scheme (AAA view)** command.

----End

Follow-up Procedure

Configure an Efficient VPN policy and reference the IKE peer on the Efficient VPN server so that the IP address, DNS domain name, DNS server IP address, and WINS server IP address can be delivered to the branch gateway.

6.10.10 (Optional) Configuring ACL Delivery

Context

NOTE

Only IKEv1 supports ACL delivery.

Efficient VPN uses the client/server model. It concentrates IPsec and other configurations on the Efficient VPN server (headquarters gateway). When basic parameters for establishing an SA are configured on the remote devices (branch gateways), the remote devices initiate a negotiation and establish an IPsec tunnel with the server. After IPsec tunnels are established, the Efficient VPN server allocates other IPsec attributes and network resources to the remote devices. Efficient VPN simplifies configurations and maintenance of IPsec and network resources for the branches.

The Efficient VPN server delivers headquarters network information defined in an ACL to the remote device. The ACL defines the headquarters subnets that branches can access. Traffic not destined for the subnets specified in the ACL is directly forwarded to the Internet. Such traffic does not pass through the IPsec tunnel.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ike peer peer-name**

An IKE peer is created and the IKE peer view is displayed.

Step 3 Run **resource acl acl-number**

An ACL is created to define subnet information about the headquarters in the Efficient VPN.

By default, no ACL is created to define subnet information about the headquarters in the Efficient VPN.

acl-number is an advanced ACL.

The sum of ACL rules pushed by the headquarters and ACL rules configured on the branch cannot exceed 512. Otherwise, the IPsec tunnels cannot be established.

---End

Follow-up Procedure

Configure an Efficient VPN policy and reference the IKE peer on the Efficient VPN server to implement ACL delivery.

6.10.11 (Optional) Enabling Dependency Between IPsec SA and IKE SA During IKEv1 Negotiation

Context

By default, no dependency exists between IPsec SA and IKE SA, that is, the two SAs can be deleted separately. If the IKE SA is deleted but the corresponding IPsec SA still exists, traffic forwarding will be effected. You can enable dependency between IPsec SA and IKE SA to ensure that an IPsec SA is deleted when its corresponding IKE SA is deleted.

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run `ikev1 phase1-phase2 sa dependent`

Dependency between IPsec SA and IKE SA during IKEv1 negotiation is enabled.

By default, no dependency exists between IPsec SA and IKE SA during IKEv1 negotiation.

----End

6.10.12 (Optional) Configuring Rapid Switchover and Revertive Switching of an IKE Peer

Context

To improve network reliability, two devices can be deployed at the headquarters to connect to the branch gateway. In an IPsec policy, two IP addresses or domain names of the remote IKE peer can be configured on the branch gateway. The branch gateway first attempts to use the first configured IP address or domain name to establish an IKE connection with the headquarters gateway. If establishing an IKE connection fails or the dead peer detection (DPD) fails, the branch gateway uses the second IP address or domain name to establish an IKE connection.

If the IP address of the first IKE peer is unreachable in the scenario that two IP addresses are configured, the branch gateway uses the second IP address to establish an IKE connection only when establishing an IKE connection fails or DPD fails. It takes a long time, and traffic cannot be switched back to the faulty gateway when it recovers.

You can use the Network Quality Analysis (NQA) or Bidirectional Forwarding Detection (BFD) to detect the status of the IP address of an IKE peer and to make sure whether the IP address of the IKE peer is valid according to the detection result. When the IP address of one IKE peer is invalid, the traffic can be quickly switched to another IKE peer. This ensures that when one headquarters gateway fails, the traffic can be quickly switched to another headquarters gateway. You can also configure revertive switching of an IKE peer to ensure that traffic can be switched back to the originally restored headquarters gateway.

Prerequisites

- If the NQA test instance status is used to determine whether the remote address of an IKE peer is valid, ensure that the NQA test instance has been created. The device supports only association between IPsec and NQA of ICMP type. For details on how to configure an NQA test instance of ICMP type, see [Configuring an ICMP Test Instance](#).
- If the BFD session instance status is used to determine whether the remote address of an IKE peer is valid, ensure that the BFD session has been created. For details on how to configure a BFD session, see [Configuring Single-Hop BFD](#).

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run **ike peer** *peer-name*

An IKE peer is created and the IKE peer view is displayed.

Step 3 Run **remote-address** [**vpn-instance** *vpn-instance-name*] { *ip-address* | **host-name** *host-name* } **track** { **nqa** *admin-name test-name* | **bfd-session** *session-name* } { **up** | **down** }

The device is configured to determine whether the remote address of an IKE peer is valid according to the NQA test instance or BFD session status.

By default, the device is configured to not determine whether the remote address of a IKE peer is valid according to the NQA test instance or BFD session status.

Step 4 (Optional) Run **switch-back enable**

Revertive switching is enabled for the IKE peer.

By default, revertive switching of an IKE peer is disabled.

----End

6.10.13 Verifying the IKE Configuration

Prerequisites

The IKE configurations are complete.

Procedure

- Run the **display ike identity** [**name** *identity-name*] command to check information about an identity filter set.
- Run the **display ike peer** [**brief** | **name** *peer-name*] or **display ike peer** [**brief** | **name** *peer-name*] **ctrl-plane** command to check IKE peer information.
- Run the **display ike proposal** [**number** *proposal-number*] or **display ike proposal** [**number** *proposal-number*] **ctrl-plane** command to check parameters in the IKE proposal.
- Run the **display ike sa** [**remote** *ipv4-address*] command to check brief information about IKE SAs.
- Run the **display ike sa** [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* command to check brief information about IKE SAs based on the remote ID.
- Run the **display ike sa verbose** { **remote** *ipv4-address* | **connection-id** *connection-id* | [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* } command to check detailed information about IKE SAs.
- Run the **display ike global config** command to check global IKE configuration.
- Run the **display ike user-table** [**number** *user-table-id* [**user-name** *user-name*]] or **display ike user-table** [**number** *user-table-id* [**user-name** *user-name*]] **ctrl-plane** command to check IKE user table information.

----End

6.11 Maintaining IPsec

6.11.1 Monitoring the IPsec Running Status

Prerequisites

All IPsec configurations are complete.

In routine maintenance, you can run the following commands in any view to check whether IPsec is functioning properly.

Procedure

- Run the **display ike sa** [**remote** *ipv4-address*] command to check brief information about IKE SAs.
- Run the **display ike sa** [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* command to check brief information about IKE SAs based on the remote ID.
- Run the **display ike sa verbose** { **remote** *ipv4-address* | **connection-id** *connection-id* | [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* } command to check detailed information about IKE SAs.
- Run the **display ipsec sa** [**brief** | **duration** | **policy** *policy-name* [*seq-number*] | **remote** *ipv4-address*] command to check IPsec SA information.
- Run the **display ipsec history record** [**remote-address** *remote-address*] command to check history information about IPsec tunnels.
- Run the **display ipsec statistics** command to check IPsec packet statistics.
- Run the **display ike statistics** { **v1** | **v2** } command to check IKE statistics.
- Run the **display ikev2 statistics** { **eap** | **error** | **notify-info** | **packet** | **sa** } command to check statistics on IPsec tunnels negotiated using IKEv2.
- Run the **display ike error-info** [**verbose**] [**peer** *remote-address*] command to check information about IPsec tunnel negotiation failures using IKE.
- Run the **display ike offline-info** [**peer** *remote-address*] command to check information about deleted IPsec tunnels established through IKE negotiation.

---End

6.11.2 Clearing IPsec Statistics

Context



NOTICE

Statistics cannot be restored after being cleared. Exercise caution when you run the reset commands.

When the number of IPsec tunnels is larger than 50% of the maximum limit, high CPU usage alarms may be generated in a short period of time after the **reset ipsec sa** or **reset ike sa** command is run. After all the SAs are cleared, the CPU usage restores to the normal range.

Procedure

- Run the **reset ipsec sa [remote *ipv4-address* | policy *policy-name* [*seq-number*] | parameters *ipv4-address* { ah | esp } spi | efficient-vpn *efficient-vpn-name* | profile *profile-name*]** command in the user view to clear established SAs.
- Run the **reset ipsec sa efficient-vpn *efficient-vpn-name*** command in the user view to clear SAs established using an Efficient VPN policy.
- Run the **reset ipsec statistics** command in the user view to clear statistics about IPsec packets.
- Run the **reset ike error-info** command in the user view to clear information about IPsec tunnel negotiation failures using IKE.
- Run the **reset ike offline-info** command in the user view to clear information about deleted IPsec tunnels established through IKE negotiation.
- Run the **reset ike sa [conn-id *conn-id* | remote [*ipv4-address*]]** command in the user view to clear the SA established using IKE.
- Run the **reset ike statistics** command in the user view to clear statistics about IKE packets.
- Run the **reset ipsec history record** command in the user view to clear history information about IPsec tunnels.

----End

6.12 Configuration Examples for IPsec

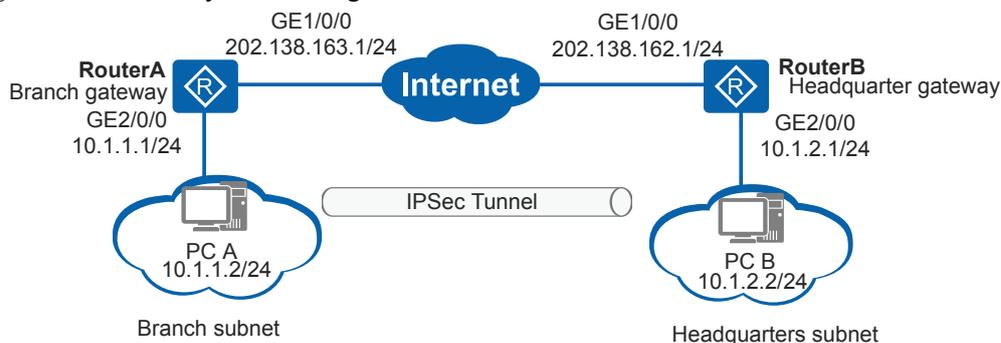
6.12.1 Example for Manually Establishing an IPsec Tunnel

Networking Requirements

As shown in [Figure 6-42](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise wants to protect data flows between the branch subnet and the headquarters subnet. An IPsec tunnel can be manually set up between the branch gateway and headquarters gateway because they communicate over the Internet and only a few branches gateway need to be maintained.

Figure 6-42 Manually establishing an IPsec tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure ACLs to define data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IPsec policies and reference ACLs and IPsec proposals in the IPsec policies to determine the methods used to protect data flows.
5. Apply IPsec policy groups to interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 202.138.163.2.

```
[RouterA] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
[RouterA] ip route-static 10.1.2.0 255.255.255.0 202.138.163.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 202.138.162.2.

```
[RouterB] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
```

Step 2 Configure ACLs on RouterA and RouterB to define data flows to be protected.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3101
[RouterA-acl-adv-3101] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[RouterA-acl-adv-3101] quit
```

Configure an ACL on **RouterB** to define data flows sent from 10.1.2.0/24 to 10.1.1.0/24.

```
[RouterB] acl number 3101
[RouterB-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[RouterB-acl-adv-3101] quit
```

Step 3 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Run the **display ipsec proposal** command on RouterA and RouterB to view the IPsec proposal configuration.

Step 4 Create IPsec policies on RouterA and RouterB.

Manually create an IPsec policy on RouterA.

```
[RouterA] ipsec policy map1 10 manual
[RouterA-ipsec-policy-manual-map1-10] security acl 3101
[RouterA-ipsec-policy-manual-map1-10] proposal tran1
[RouterA-ipsec-policy-manual-map1-10] tunnel remote 202.138.162.1
[RouterA-ipsec-policy-manual-map1-10] tunnel local 202.138.163.1
[RouterA-ipsec-policy-manual-map1-10] sa spi outbound esp 12345
[RouterA-ipsec-policy-manual-map1-10] sa spi inbound esp 54321
[RouterA-ipsec-policy-manual-map1-10] sa string-key outbound esp cipher huawei
[RouterA-ipsec-policy-manual-map1-10] sa string-key inbound esp cipher huawei
[RouterA-ipsec-policy-manual-map1-10] quit
```

Manually create an IPsec policy on RouterB.

```
[RouterB] ipsec policy use1 10 manual
[RouterB-ipsec-policy-manual-use1-10] security acl 3101
[RouterB-ipsec-policy-manual-use1-10] proposal tran1
[RouterB-ipsec-policy-manual-use1-10] tunnel remote 202.138.163.1
[RouterB-ipsec-policy-manual-use1-10] tunnel local 202.138.162.1
[RouterB-ipsec-policy-manual-use1-10] sa spi outbound esp 54321
[RouterB-ipsec-policy-manual-use1-10] sa spi inbound esp 12345
[RouterB-ipsec-policy-manual-use1-10] sa string-key outbound esp cipher huawei
[RouterB-ipsec-policy-manual-use1-10] sa string-key inbound esp cipher huawei
[RouterB-ipsec-policy-manual-use1-10] quit
```

Run the **display ipsec policy** command on RouterA and RouterB to view the configurations of the IPsec policies.

Step 5 Apply IPsec policy groups to interfaces on RouterA and RouterB.

Apply the IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec policy map1
[RouterA-GigabitEthernet1/0/0] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy use1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 6 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ipsec sa** command on RouterA and RouterB to view the IPsec configuration. The display on RouterA is used as an example.

```
[RouterA] display ipsec sa
ipsec sa information:
=====
Interface: GigabitEthernet1/0/0
=====

-----
IPSec policy name: "map1"
Sequence number: 10
Acl group: 3101
Acl rule: -
Mode: Manual
-----

Encapsulation mode: Tunnel
Tunnel local      : 202.138.163.1
Tunnel remote     : 202.138.162.1

[Outbound ESP SAs]
SPI: 12345 (0x3039)
Proposal: ESP-ENCRYPT-AES-128 SHA2-256-128
No duration limit for this SA

[Inbound ESP SAs]
SPI: 54321 (0xd431)
Proposal: ESP-ENCRYPT-AES-128 SHA2-256-128
No duration limit for this SA
Anti-replay : Disable
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3101
rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ipsec policy map1 10 manual
security acl 3101
proposal tran1
tunnel local 202.138.163.1
tunnel remote 202.138.162.1
sa spi inbound esp 54321
sa string-key inbound esp cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!
%^%#
sa spi outbound esp 12345
sa string-key outbound esp cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4n0j
%5W5=)%^%#
#
interface GigabitEthernet1/0/0
ip address 202.138.163.1 255.255.255.0
ipsec policy map1
#
```

```
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
 #
 ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
 ip route-static 10.1.2.0 255.255.255.0 202.138.163.2
 #
 return
```

- Configuration file of RouterB

```
#
 sysname RouterB
 #
 acl number 3101
 rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
 #
 ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
 #
 ipsec policy use1 10 manual
 security acl 3101
 proposal tran1
 tunnel local 202.138.162.1
 tunnel remote 202.138.163.1
 sa spi inbound esp 12345
 sa string-key inbound esp cipher %^%#IRFGElFPJl$a'Qy,L*XQL_+*Grq-
=yMb)ULZdS6%^%#
 sa spi outbound esp 54321
 sa string-key outbound esp cipher %^%#(3fr1!&60=)!GN#~()n,2fq>4#4+
%;lMTs5():c%^%#
 #
 interface GigabitEthernet1/0/0
 ip address 202.138.162.1 255.255.255.0
 ipsec policy use1
 #
 interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
 #
 ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
 ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
 #
 return
```

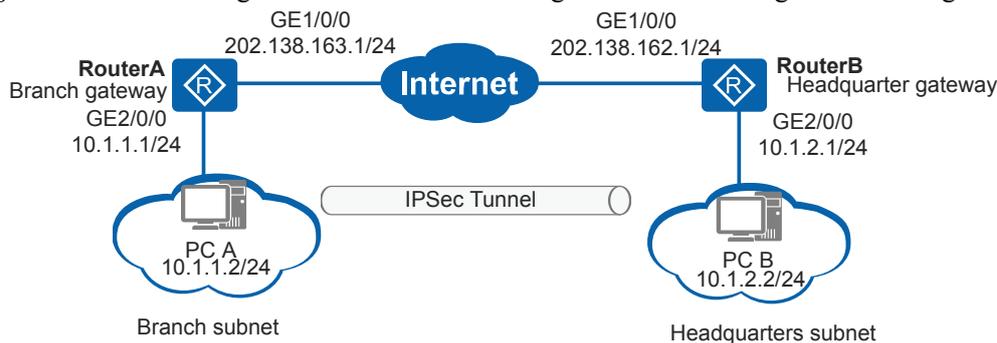
6.12.2 Example for Establishing an IPsec Tunnel in IKE Negotiation Mode Using Default Settings

Networking Requirements

As shown in [Figure 6-43](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise wants to protect data flows between the branch subnet and the headquarters subnet. An IPsec tunnel can be set up between the branch gateway and headquarters gateway because they communicate over the Internet.

Figure 6-43 Establishing an IPsec tunnel in IKE negotiation mode using default settings



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure ACLs to define data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec policies and reference ACLs and IPsec proposals in the IPsec policies to determine the methods used to protect data flows.
6. Apply IPsec policy groups to interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 202.138.163.2.

```
[RouterA] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
[RouterA] ip route-static 10.1.2.0 255.255.255.0 202.138.163.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 202.138.162.2.

```
[RouterB] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
```

Step 2 Configure ACLs on RouterA and RouterB to define data flows to be protected.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3101
[RouterA-acl-adv-3101] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[RouterA-acl-adv-3101] quit
```

Configure an ACL on RouterB to define data flows sent from 10.1.2.0/24 to 10.1.1.0/24.

```
[RouterB] acl number 3101
[RouterB-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[RouterB-acl-adv-3101] quit
```

Step 3 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Run the **display ipsec proposal** command on RouterA and RouterB to view the IPsec proposal configuration.

Step 4 Configure IKE peers on RouterA and RouterB.

Configure an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA and set the pre-shared key and remote ID according to default settings.

```
[RouterA] ike peer spub
[RouterA-ike-peer-spub] ike-proposal 5
[RouterA-ike-peer-spub] pre-shared-key cipher huawei@123
[RouterA-ike-peer-spub] remote-address 202.138.162.1
[RouterA-ike-peer-spub] quit
```

Configure an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB and set the pre-shared key and remote ID according to default settings.

```
[RouterB] ike peer spua
[RouterB-ike-peer-spua] ike-proposal 5
[RouterB-ike-peer-spua] pre-shared-key cipher huawei@123
[RouterB-ike-peer-spua] remote-address 202.138.163.1
[RouterB-ike-peer-spua] quit
```

Step 5 Create IPsec policies on RouterA and RouterB.

Create an IPsec policy in IKE negotiation mode on RouterA.

```
[RouterA] ipsec policy map1 10 isakmp
[RouterA-ipsec-policy-isakmp-map1-10] ike-peer spub
[RouterA-ipsec-policy-isakmp-map1-10] proposal tran1
[RouterA-ipsec-policy-isakmp-map1-10] security acl 3101
[RouterA-ipsec-policy-isakmp-map1-10] quit
```

Create an IPsec policy in IKE negotiation mode on RouterB.

```
[RouterB] ipsec policy use1 10 isakmp
[RouterB-ipsec-policy-isakmp-use1-10] ike-peer spua
[RouterB-ipsec-policy-isakmp-use1-10] proposal tran1
[RouterB-ipsec-policy-isakmp-use1-10] security acl 3101
[RouterB-ipsec-policy-isakmp-use1-10] quit
```

Run the **display ipsec policy** command on RouterA and RouterB to view the configurations of the IPsec policies.

Step 6 Apply IPsec policy groups to interfaces on RouterA and RouterB.

Apply the IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec policy map1
[RouterA-GigabitEthernet1/0/0] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy use1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 7 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID   Peer                VPN   Flag(s)   Phase   RemoteType   RemoteID
-----
  16        202.138.162.1:500   RD|ST v2:2     IP      202.138.162.1
  14        202.138.162.1:500   RD|ST v2:1     IP      202.138.162.1

Number of IKE SA : 2

-----
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3101
rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm
aes-128
dh
group14
authentication-algorithm
sha2-256
authentication-method pre-
share
integrity-algorithm hmac-
sha2-256
prf hmac-sha2-256
#
ike peer spub
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
ike-proposal 5
remote-address 202.138.162.1
#
ipsec policy map1 10 isakmp
security acl 3101
ike-peer spub
proposal tran1
#
interface GigabitEthernet1/0/0
ip address 202.138.163.1 255.255.255.0
ipsec policy map1
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
ip route-static 10.1.2.0 255.255.255.0 202.138.163.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
acl number 3101
rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm
aes-128
dh
group14
authentication-algorithm
sha2-256
authentication-method pre-
share
integrity-algorithm hmac-
```

```
sha2-256
prf hmac-sha2-256
#
ike peer spua
pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%^%#
ike-proposal 5
remote-address 202.138.163.1
#
ipsec policy use1 10 isakmp
security acl 3101
ike-peer spua
proposal tran1
#
interface GigabitEthernet1/0/0
ip address 202.138.162.1 255.255.255.0
ipsec policy use1
#
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
#
return
```

6.12.3 Example for Establishing an IPsec Tunnel Between the Enterprise Headquarters and Branch Using an IPsec Policy Template

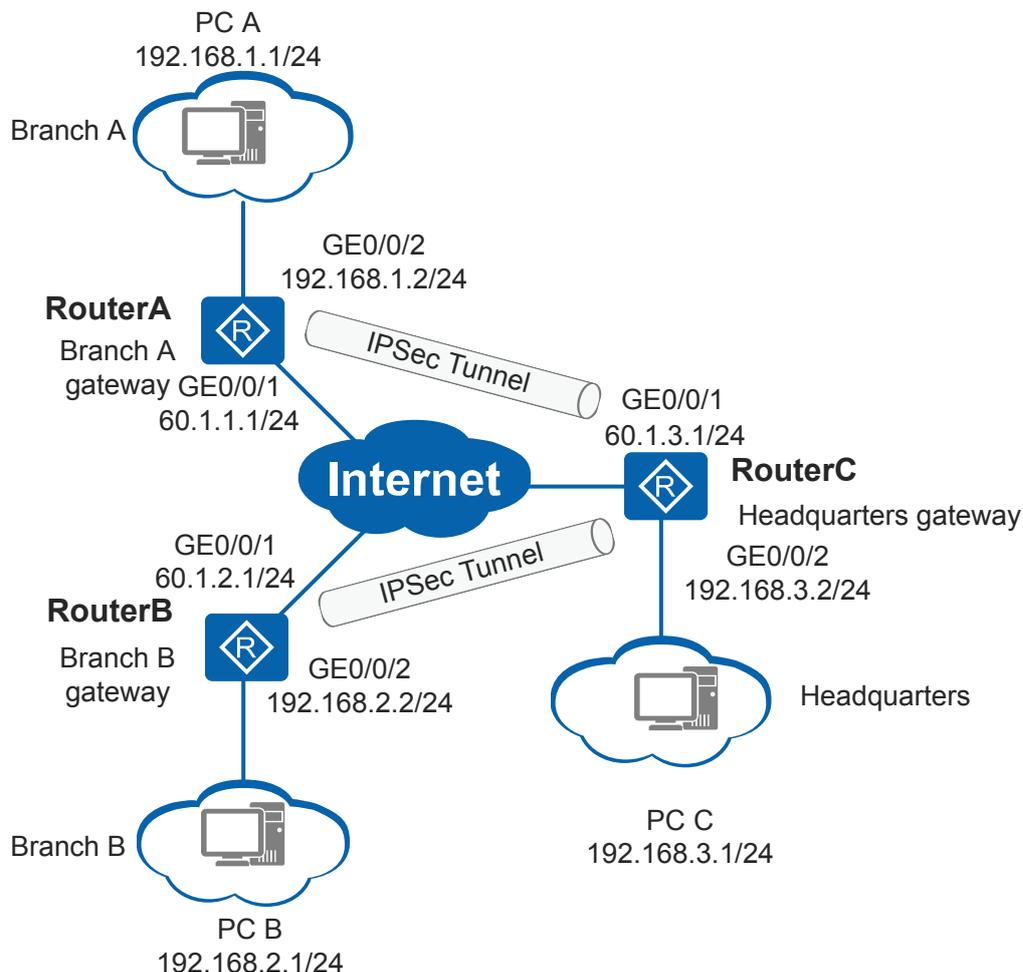
Networking Requirements

In [Figure 6-44](#), RouterA and RouterB are enterprise branch gateways. RouterA uses a fixed IP address to access the public network, whereas RouterB uses a dynamic IP address to access the public network. RouterC is the enterprise headquarters gateway. Branches and the headquarters communicate through the public network.

The enterprise wants to protect the data flows between the branch subnets and the headquarters subnet and wants the headquarters gateway to specify a branch gateway meeting specific criteria to access it for security.

IPsec tunnels can be set up between the branch gateways and headquarters gateway because they communicate over the Internet.

Figure 6-44 Establishing an IPsec tunnel between the enterprise headquarters and branch using an IPsec policy template



Configuration Roadmap

The headquarters gateway can only respond to IPsec negotiation requests initiated by branch gateways because it cannot identify IP addresses of branch gateways. An IPsec policy template is configured on RouterC and is referenced in an IPsec policy so that RouterC can receive IPsec negotiation requests initiated by branch gateways to complete setup of multiple IPsec tunnels.

1. Configure IP addresses and static routes for interfaces so that routes among the three gateways are reachable.
2. Configure ACLs to define the data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
 - Configure RouterA to use its IP address for authentication with RouterC because RouterA uses a fixed IP address for access.
 - Configure RouterB to use its name for authentication with RouterC because RouterB uses a dynamic IP address for access.

5. Configure an identity filter set on RouterC to permit access from RouterA and RouterB. This prevents other unauthorized initiators from establishing an IPsec tunnel with RouterC.
 - Check the IP address of RouterA.
 - Check the name of RouterB.
6. Configure IPsec policies on RouterA, RouterB, and RouterC. RouterC uses an IPsec policy template to create an IPsec policy.
7. Apply IPsec policy groups to interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA, RouterB, and RouterC so that routes among them are reachable.

Assign an IP address to each interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next-hop address of the route to RouterC is 60.1.1.2.

```
[RouterA] ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
```

Assign an IP address to each interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ip address dhcp-alloc
[RouterB-GigabitEthernet0/0/1] quit
[RouterB] interface gigabitethernet 0/0/2
[RouterB-GigabitEthernet0/0/2] ip address 192.168.2.2 255.255.255.0
[RouterB-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterB. This example assumes that the outbound interface of the route to the headquarters is GE0/0/1.

```
[RouterB] ip route-static 60.1.3.0 255.255.255.0 gigabitethernet 0/0/1
[RouterB] ip route-static 192.168.3.0 255.255.255.0 60.gigabitethernet 0/0/1
```

Assign an IP address to each interface on RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ip address 60.1.3.1 255.255.255.0
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ip address 192.168.3.2 255.255.255.0
[RouterC-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterC. This example assumes that the next-hop address of the route to RouterA and RouterB is 60.1.3.2.

```
[RouterC] ip route-static 0.0.0.0 0.0.0.0 60.1.3.2
```

Step 2 Configure ACLs on RouterA and RouterB to define the data flows to be protected.

NOTE

Because RouterC uses an IPsec policy template to create an IPsec policy, so referencing an ACL is optional. If an ACL is configured on RouterC, specify the destination address in the ACL.

Configure an ACL on RouterA to define the data flows sent from 192.168.1.0/24 to 192.168.3.0/24.

```
[RouterA] acl number 3002
[RouterA-acl-adv-3002] rule permit ip source 192.168.1.0 0.0.0.255 destination
192.168.3.0 0.0.0.255
[RouterA-acl-adv-3002] quit
```

Configure an ACL on RouterB to define the data flows sent from 192.168.2.0/24 to 192.168.3.0/24.

```
[RouterB] acl number 3002
[RouterB-acl-adv-3002] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.3.0 0.0.0.255
[RouterB-acl-adv-3002] quit
```

Step 3 Create IPsec proposals on RouterA, RouterB, and RouterC.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterC.

```
[RouterC] ipsec proposal tran1
[RouterC-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterC-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterC-ipsec-proposal-tran1] quit
```

Step 4 Configure IKE peers on RouterA, RouterB, and RouterC.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterA-ike-peer-rut1] remote-address 60.1.3.1
[RouterA-ike-peer-rut1] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
```

```
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

NOTE

Configure the local name as huaweirt1 and the local ID type as FQDN for IKE negotiation because RouterB uses a dynamic IP address for access.

```
[RouterB] ike local-name huaweirt1
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterB-ike-peer-rut1] local-id-type fqdn
[RouterB-ike-peer-rut1] remote-address 60.1.3.1
[RouterB-ike-peer-rut1] quit
```

Create an IKE peer on RouterC.

```
[RouterC] ike proposal 5
[RouterC-ike-proposal-5] encryption-algorithm aes-128
[RouterC-ike-proposal-5] authentication-algorithm sha2-256
[RouterC-ike-proposal-5] dh group14
[RouterC-ike-proposal-5] quit
```

Configure an IKE peer on RouterC.

NOTE

RouterC functions as the IKE responder and uses an IPsec policy template to create an IPsec policy, so the **remote-address** command does not need to be used.

```
[RouterC] ike peer rut1
[RouterC-ike-peer-rut1] ike-proposal 5
[RouterC-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterC-ike-peer-rut1] quit
```

Step 5 Configure an identity filter set on RouterC.

```
[RouterC] ike identity identity1
[RouterC-ike-identity-identity1] ip address 60.1.1.1 24
[RouterC-ike-identity-identity1] fqdn huaweirt1
[RouterC-ike-identity-identity1] quit
```

Step 6 Configuring IPsec policies on RouterA, RouterB, and RouterC. RouterC uses an IPsec policy template to create an IPsec policy.

Create an IPsec policy on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterA-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy on RouterB.

```
[RouterB] ipsec policy policy1 10 isakmp
[RouterB-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterB-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterB-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterB-ipsec-policy-isakmp-policy1-10] quit
```

Configure an IPsec policy template on RouterC and reference the IPsec policy template in the IPsec policy.

```
[RouterC] ipsec policy-template use1 10
[RouterC-ipsec-policy-templet-use1-10] ike-peer rut1
[RouterC-ipsec-policy-templet-use1-10] proposal tran1
[RouterC-ipsec-policy-templet-use1-10] match ike-identity identity1
```

```
[RouterC-ipsec-policy-templet-use1-10] quit
[RouterC] ipsec policy policy1 10 isakmp template use1
```

Step 7 Apply IPsec policy groups to interfaces on RouterA, RouterB, and RouterC.

Apply an IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to the interface of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
```

Step 8 Verify the configuration.

After the configurations are complete, PC A and PC B can ping PC C successfully. The data transmitted between PC A, PC B, and PC C is encrypted.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer           VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  24366    60.1.3.1:500   RD|ST  v2:2    IP     60.1.3.1
  24274    60.1.3.1:500   RD|ST  v2:1    IP     60.1.3.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

Run the **display ike sa** command on RouterC. The following information is displayed:

```
[RouterC] display ike sa
IKE SA information :
  Conn-ID  Peer           VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  961      60.1.2.1:500   RD    v2:2    FQDN   huaweirt1
  933      60.1.2.1:500   RD    v2:1    FQDN   huaweirt1
  937      60.1.1.1:500   RD    v2:2    IP     60.1.1.1
  936      60.1.1.1:500   RD    v2:1    IP     60.1.1.1

Number of IKE SA : 4
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3002
rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.3.0
0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
ike-proposal 5
remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
security acl 3002
ike-peer rut1
proposal tran1
#
interface GigabitEthernet0/0/1
ip address 60.1.1.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.1.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ike local-name huaweirt1
#
acl number 3002
rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.3.0
0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%^%#
ike-proposal 5
```

```
local-id-type fqdn
remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
security acl 3002
ike-peer rut1
proposal tran1
#
interface GigabitEthernet0/0/1
ip address dhcp-alloc
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.2.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 GigabitEthernet0/0/1
ip route-static 192.168.3.0 255.255.255.0 GigabitEthernet0/0/1
#
return
```

● Configuration file of RouterC

```
#
sysname RouterC
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %%#IRFGEiFPJ1$&a'Qy,L*XQL_+*Grq-=yMb}ULZdS6%^%#
ike-proposal 5
#
ike identity identity1
fqdn huaweirt1
ip address 60.1.1.0 255.255.255.0
#
ipsec policy-template use1 10
ike-peer rut1
proposal tran1
match ike-identity identity1
#
ipsec policy policy1 10 isakmp template use1
#
interface GigabitEthernet0/0/1
ip address 60.1.3.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.3.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 60.1.3.2
#
return
```

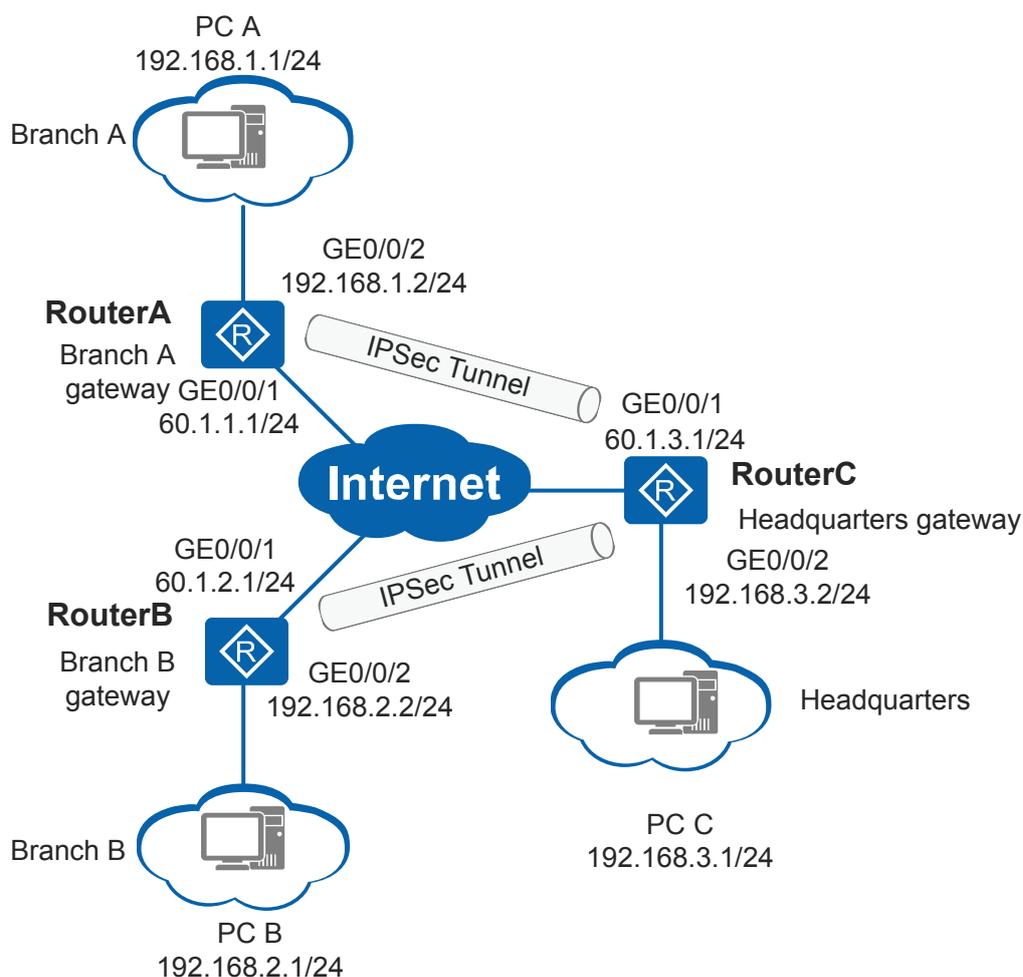
6.12.4 Example for Establishing Multiple IPsec Tunnels Between the Enterprise Headquarters and Branches Using IPsec Policy Groups

Networking Requirements

As shown in **Figure 6-45**, RouterA and RouterB are branch gateways, and RouterC is the headquarters gateway. The headquarters and branches communicate through the Internet. The gateways' IP addresses are fixed. The subnets of branch A, branch B, and headquarters are 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24 respectively.

The enterprise wants to protect data flows between the branch subnets and the headquarters subnet. IPsec tunnels can be set up between the branch gateways and headquarters gateway because they communicate over the Internet. Because branch gateways' IP addresses can be specified on the headquarters gateway, an IPsec policy group can be configured on RouterC. Then the headquarters gateway can initiate IPsec negotiation to each branch gateway or receive IPsec negotiation requests from each branch gateway to complete setup of multiple IPsec tunnels.

Figure 6-45 Establishing multiple IPsec tunnels between the enterprise headquarters and branches using IPsec policies



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces so that routes among the three gateways are reachable.
2. Configure ACLs to define data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec policies on RouterA and RouterB. Create IPsec policy groups on RouterC to define protection methods for data flows between RouterA and RouterC, and between RouterB and RouterC.
6. Apply IPsec policy groups to interfaces.

Procedure

- Step 1** Configure IP addresses and static routes for interfaces on RouterA, RouterB, and RouterC so that routes among them are reachable.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterC is 60.1.1.2.

```
[RouterA] ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet0/0/1] quit
[RouterB] interface gigabitethernet 0/0/2
[RouterB-GigabitEthernet0/0/2] ip address 192.168.2.2 255.255.255.0
[RouterB-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterC is 60.1.2.2.

```
[RouterB] ip route-static 60.1.3.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
```

Assign an IP address to an interface on RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ip address 60.1.3.1 255.255.255.0
```

```
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ip address 192.168.3.2 255.255.255.0
[RouterC-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterC. This example assumes that the next hop address in the route to RouterA and RouterB is 60.1.3.2.

```
[RouterC] ip route-static 60.1.1.0 255.255.255.0 60.1.3.2
[RouterC] ip route-static 60.1.2.0 255.255.255.0 60.1.3.2
[RouterC] ip route-static 192.168.1.0 255.255.255.0 60.1.3.2
[RouterC] ip route-static 192.168.2.0 255.255.255.0 60.1.3.2
```

Step 2 Configure ACLs on RouterA, RouterB, and RouterC to define data flows to be protected.

Configure an ACL on RouterA to define data flows sent from 192.168.1.0/24 to 192.168.3.0/24.

```
[RouterA] acl number 3002
[RouterA-acl-adv-3002] rule permit ip source 192.168.1.0 0.0.0.255 destination
192.168.3.0 0.0.0.255
[RouterA-acl-adv-3002] quit
```

Configure an ACL on RouterB to define data flows sent from 192.168.2.0/24 to 192.168.3.0/24.

```
[RouterB] acl number 3002
[RouterB-acl-adv-3002] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.3.0 0.0.0.255
[RouterB-acl-adv-3002] quit
```

Configure an ACL on RouterC to define data flows sent from 192.168.3.0/24 to 192.168.1.0/24 and 192.168.2.0/24.

```
[RouterC] acl number 3002
[RouterC-acl-adv-3002] rule permit ip source 192.168.3.0 0.0.0.255 destination
192.168.1.0 0.0.0.255
[RouterC-acl-adv-3002] quit
[RouterC] acl number 3003
[RouterC-acl-adv-3003] rule permit ip source 192.168.3.0 0.0.0.255 destination
192.168.2.0 0.0.0.255
[RouterC-acl-adv-3003] quit
```

Step 3 Create IPsec proposals on RouterA, RouterB, and RouterC.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterC.

```
[RouterC] ipsec proposal tran1
[RouterC-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterC-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterC-ipsec-proposal-tran1] quit
```

Run the **display ipsec proposal** command on RouterA, RouterB, and RouterC to view the IPsec proposal configuration. The display on RouterA is used as an example.

```
[RouterA] display ipsec proposal name tran1
IPsec proposal name: tran1
Encapsulation mode: Tunnel
Transform          : esp-new
ESP protocol       : Authentication SHA2-HMAC-256
                   Encryption      AES-128
```

Step 4 Configure IKE peers on RouterA, RouterB, and RouterC.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterA-ike-peer-rut1] remote-address 60.1.3.1
[RouterA-ike-peer-rut1] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

```
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterB-ike-peer-rut1] remote-address 60.1.3.1
[RouterB-ike-peer-rut1] quit
```

Create an IKE proposal on RouterC.

```
[RouterC] ike proposal 5
[RouterC-ike-proposal-5] encryption-algorithm aes-128
[RouterC-ike-proposal-5] authentication-algorithm sha2-256
[RouterC-ike-proposal-5] dh group14
[RouterC-ike-proposal-5] quit
```

Configure an IKE peer on RouterC.

```
[RouterC] ike peer rut1
[RouterC-ike-peer-rut1] ike-proposal 5
[RouterC-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterC-ike-peer-rut1] remote-address 60.1.1.1
[RouterC-ike-peer-rut1] quit
[RouterC] ike peer rut2
[RouterC-ike-peer-rut2] ike-proposal 5
[RouterC-ike-peer-rut2] pre-shared-key cipher huawei@123
[RouterC-ike-peer-rut2] remote-address 60.1.2.1
[RouterC-ike-peer-rut2] quit
```

Step 5 Configure IPsec policies on RouterA and RouterB, and configure an IPsec policy group on RouterC.

Create an IPsec policy on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut1
```

```
[RouterA-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy on RouterB.

```
[RouterB] ipsec policy policy1 10 isakmp
[RouterB-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterB-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterB-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterB-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy group on RouterC.

```
[RouterC] ipsec policy policy1 10 isakmp
[RouterC-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterC-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterC-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterC-ipsec-policy-isakmp-policy1-10] quit
[RouterC] ipsec policy policy1 11 isakmp
[RouterC-ipsec-policy-isakmp-policy1-11] ike-peer rut2
[RouterC-ipsec-policy-isakmp-policy1-11] proposal tran1
[RouterC-ipsec-policy-isakmp-policy1-11] security acl 3003
[RouterC-ipsec-policy-isakmp-policy1-11] quit
```

Run the **display ipsec policy** command on RouterA and RouterB to view the configurations of the IPsec policies.

Run the **display ipsec policy** command on RouterC.

Step 6 Apply IPsec policy groups to interfaces on RouterA, RouterB, and RouterC.

Apply the IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy group to the interface of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
```

Step 7 Verify the configuration.

After the configurations are complete, PC A and PC B can ping PC C successfully. The data transmitted between PC A, PC B, and PC C is encrypted.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
Conn-ID  Peer          VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
24366   60.1.3.1:500      RD|ST  v2:2    IP     60.1.3.1
24274   60.1.3.1:500      RD|ST  v2:1    IP     60.1.3.1

Number of IKE SA : 2
-----
```

```

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE     NEG--NEGOTIATING

[RouterA] display ike sa
IKE SA information :
Conn-ID  Peer          VPN   Flag(s)  Phase  RemoteType  RemoteID
-----
24366   60.1.3.1:500  RD|ST v2:2     IP     60.1.3.1
24274   60.1.3.1:500  RD|ST v2:1     IP     60.1.3.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE     NEG--NEGOTIATING
  
```

Run the **display ike sa** command on RouterC. The following information is displayed:

```

[RouterC] display ike sa
IKE SA information :
Conn-ID  Peer          VPN   Flag(s)  Phase  RemoteType  RemoteID
-----
961     60.1.2.1:500  RD    v2:2     IP     60.1.2.1
933     60.1.2.1:500  RD    v2:1     IP     60.1.2.1
937     60.1.1.1:500  RD    v2:2     IP     60.1.1.1
936     60.1.1.1:500  RD    v2:1     IP     60.1.1.1

Number of IKE SA : 4
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE     NEG--NEGOTIATING
  
```

---End

Configuration Files

- Configuration file of RouterA

```

#
 sysname RouterA
#
acl number 3002
 rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.3.0
 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\!*%#
 ike-proposal 5
 remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
  
```

```
security acl 3002
ike-peer rut1
proposal tran1
#
interface GigabitEthernet0/0/1
ip address 60.1.1.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.1.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
#
return
```

● Configuration file of RouterB

```
#
sysname RouterB
#
acl number 3002
rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.3.0
0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %%K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
ike-proposal 5
remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
security acl 3002
ike-peer rut1
proposal tran1
#
interface GigabitEthernet0/0/1
ip address 60.1.2.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.2.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 60.1.2.2
ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
#
return
```

● Configuration file of RouterC

```
#
sysname RouterC
#
acl number 3002
rule 5 permit ip source 192.168.3.0 0.0.0.255 destination 192.168.1.0
0.0.0.255
acl number 3003
rule 5 permit ip source 192.168.3.0 0.0.0.255 destination 192.168.2.0
0.0.0.255
#
ipsec proposal tran1
```

```
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%#IRFGEiFPJl$&a'Qy,L*XQL_+*Grq-=yMb}ULZdS6%^%#
 ike-proposal 5
 remote-address 60.1.1.1
#
ike peer rut2
 pre-shared-key cipher %%#(3fr1!&6O=)!GN#~{)n,2fq>4#4+%;lMTs5(]:c)%%#
 ike-proposal 5
 remote-address 60.1.2.1
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
ipsec policy policy1 11 isakmp
 security acl 3003
 ike-peer rut2
 proposal tran1
#
interface GigabitEthernet0/0/1
 ip address 60.1.3.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.3.2 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.3.2
ip route-static 60.1.2.0 255.255.255.0 60.1.3.2
ip route-static 192.168.1.0 255.255.255.0 60.1.3.2
ip route-static 192.168.2.0 255.255.255.0 60.1.3.2
#
return
```

6.12.5 Example for Establishing IPsec Tunnels for Branch Access to the Headquarters Using Different Pre-shared Keys

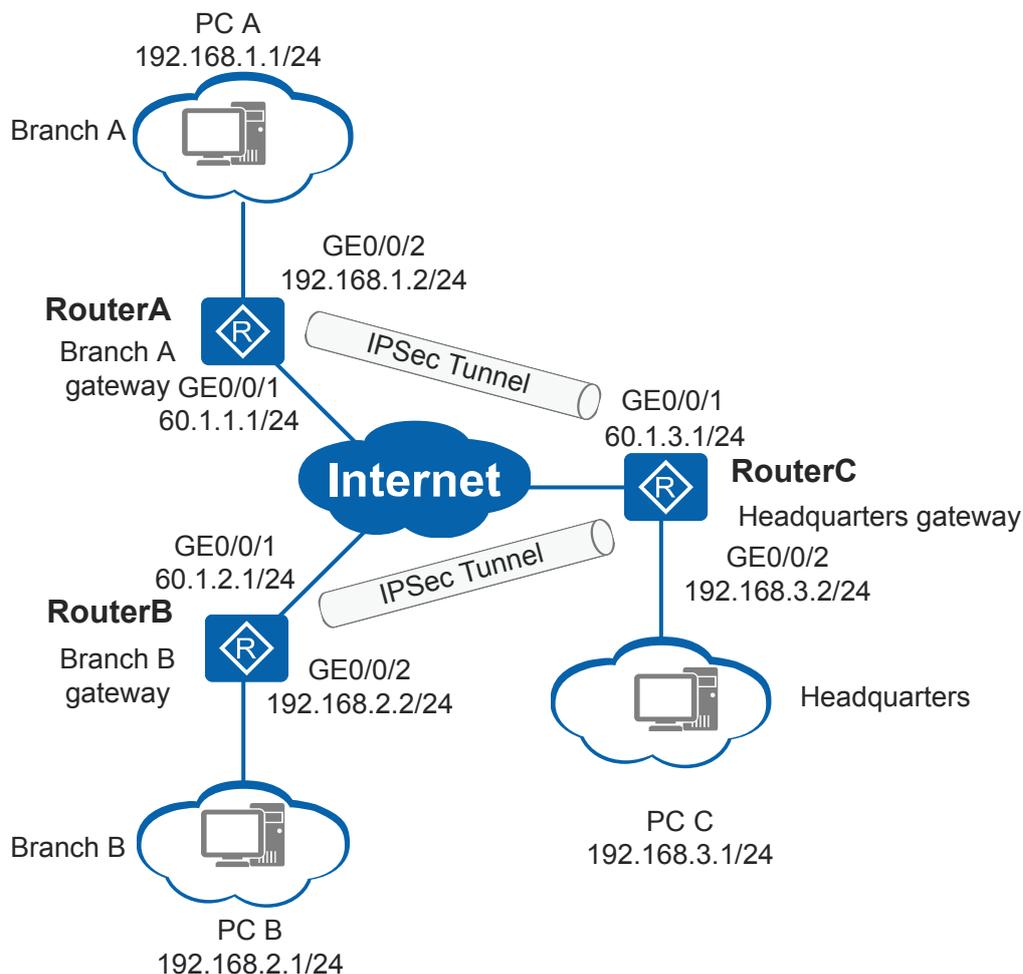
Networking Requirements

In [Figure 6-46](#), RouterA and RouterB are the branch gateways of an enterprise, and RouterC is the gateway of the headquarters. They communicate over the Internet.

The enterprise wants to protect traffic transmitted between the branches and headquarters. To improve security, branch gateways are required to use different pre-shared keys to connect to the headquarters gateway.

IPsec tunnels can be established between the headquarters gateway and branch gateways to protect communication between the headquarters and branches over the Internet.

Figure 6-46 Establishing IPsec tunnels for branch access to the headquarters using different pre-shared keys



Configuration Roadmap

The headquarters gateway can only respond to IPsec negotiation requests initiated by branch gateways because it is difficult to specify IP addresses for branch gateways on the headquarters gateway. As a result, you can deploy a policy template on RouterC and reference this template in an IPsec policy. To allow branch gateways to connect to the headquarters using different pre-shared keys, configure an IKE user table on RouterC to allocate pre-shared keys for branches. The branches initiate IPsec negotiation using allocated pre-shared keys to establish IPsec tunnels.

1. Configure an IP address and a static route on each interface to implement communication between both ends.
2. Configure an ACL to define the data flows to be protected by IPsec.
3. Configure an IPsec proposal to define the traffic protection method.
4. Configure an IKE peer and define the attributes used for IKE negotiation. The IKE user table on RouterC allocates pre-shared keys for branches.

5. Create an IPsec policy on RouterA, RouterB, and RouterC respectively to determine protection methods used for protecting different types of data flows. On RouterC, an IPsec policy is created through a policy template.
6. Apply the IPsec policy group to an interface so that the interface can protect traffic.

Procedure

- Step 1** Configure an IP address and a static route for each interface on RouterA, RouterB, and RouterC to ensure that there are reachable routes among them.

Configure an IP address for each interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to the headquarters is 60.1.1.2.

```
[RouterA] ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
```

Configure an IP address for each interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet0/0/1] quit
[RouterB] interface gigabitethernet 0/0/2
[RouterB-GigabitEthernet0/0/2] ip address 192.168.2.2 255.255.255.0
[RouterB-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to the headquarters is 60.1.2.2.

```
[RouterB] ip route-static 60.1.3.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
```

Configure an IP address for each interface on RouterC.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ip address 60.1.3.1 255.255.255.0
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ip address 192.168.3.2 255.255.255.0
[RouterC-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterC. This example assumes that the next hop address in the route to the branch gateways A and B is 60.1.3.2.

```
[RouterC] ip route-static 0.0.0.0 0.0.0.0 60.1.3.2
```

- Step 2** Configure an ACL on RouterA and RouterB to define the data flows to be protected.

NOTE

RouterC creates an IPsec policy through the IPsec policy template; therefore, this step is optional. If you configure an ACL on RouterC, you must specify the destination address in the ACL rule.

Configure an ACL on RouterA to define the data flows from 192.168.1.0/24 to 192.168.3.0/24.

```
[RouterA] acl number 3002
[RouterA-acl-adv-3002] rule permit ip source 192.168.1.0 0.0.0.255 destination 192.168.3.0 0.0.0.255
[RouterA-acl-adv-3002] quit
```

Configure an ACL on RouterB to define the data flows from 192.168.2.0/24 to 192.168.3.0/24.

```
[RouterB] acl number 3002
[RouterB-acl-adv-3002] rule permit ip source 192.168.2.0 0.0.0.255 destination 192.168.3.0 0.0.0.255
[RouterB-acl-adv-3002] quit
```

Step 3 Create an IPsec proposal on RouterA, RouterB, and RouterC respectively.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterC.

```
[RouterC] ipsec proposal tran1
[RouterC-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterC-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterC-ipsec-proposal-tran1] quit
```

Step 4 Create an IKE peer on RouterA, RouterB, and RouterC respectively.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Create an IKE peer on RouterA.

```
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterA-ike-peer-rut1] remote-address 60.1.3.1
[RouterA-ike-peer-rut1] quit
```

Configure an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Create an IKE peer on RouterB.

```
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] ike-proposal 5
```

```
[RouterB-ike-peer-rut1] pre-shared-key cipher huawei@124
[RouterB-ike-peer-rut1] remote-address 60.1.3.1
[RouterB-ike-peer-rut1] quit
```

Create an IKE proposal on RouterC.

```
[RouterC] ike proposal 5
[RouterC-ike-proposal-5] encryption-algorithm aes-128
[RouterC-ike-proposal-5] authentication-algorithm sha2-256
[RouterC-ike-proposal-5] dh group14
[RouterC-ike-proposal-5] quit
```

Configure an IKE user table on RouterC to allocate pre-shared keys for branches.

NOTE

When the IKEv1 main mode and pre-shared key authentication are used, **id-type** can only be set to **ip**, and in NAT traversal scenarios, the IP address must be set to the post-NAT address. If a branch dynamically obtains an IP address, you must use the IKEv1 aggressive mode or IKEv2 and advised to set **id-type** to **fqdn**.

```
[RouterC] ike user-table 10
[RouterC-ike-user-table-10] user routera
[RouterC-ike-user-table-10-routera] id-type ip 60.1.1.1
[RouterC-ike-user-table-10-routera] pre-shared-key huawei@123
[RouterC-ike-user-table-10-routera] quit
[RouterC-ike-user-table-10] user routerb
[RouterC-ike-user-table-10-routerb] id-type ip 60.1.2.1
[RouterC-ike-user-table-10-routerb] pre-shared-key huawei@124
[RouterC-ike-user-table-10-routerb] quit
```

Create an IKE peer on RouterC.

NOTE

As the responder to IKE negotiation requests, RouterC creates an IPsec policy through the IPsec policy template. You do not need to set **remote-address**.

```
[RouterC] ike peer rut1
[RouterC-ike-peer-rut1] ike-proposal 5
[RouterC-ike-peer-rut1] user-table 10
[RouterC-ike-peer-rut1] quit
```

Step 5 Create an IPsec policy on RouterA, RouterB, and RouterC respectively. RouterC creates an IPsec policy through the IPsec policy template.

Create an IPsec policy on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterA-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy on RouterB.

```
[RouterB] ipsec policy policy1 10 isakmp
[RouterB-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterB-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterB-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterB-ipsec-policy-isakmp-policy1-10] quit
```

Create a policy template on RouterC and apply the policy template to an IPsec policy.

```
[RouterC] ipsec policy-template use1 10
[RouterC-ipsec-policy-templet-use1-10] ike-peer rut1
[RouterC-ipsec-policy-templet-use1-10] proposal tran1
[RouterC-ipsec-policy-templet-use1-10] quit
[RouterC] ipsec policy policy1 10 isakmp template use1
```

Step 6 Apply an IPsec policy group to the interface of RouterA, RouterB, and RouterC.

Apply an IPsec policy group to an interface of RouterA.

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to an interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to an interface of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
```

Step 7 Verify the configuration.

After the configurations are complete, PC A and PC B can ping PC C successfully. The data transmitted among them is encrypted.

Run the **display ike sa** command on RouterA and RouterB to view the IKE configuration. The command output on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
24366 60.1.3.1:500 RD|ST v2:2 IP 60.1.3.1
24274 60.1.3.1:500 RD|ST v2:1 IP 60.1.3.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

Run the **display ike sa** command on RouterC. The command output is displayed as follows:

```
[RouterC] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
961 60.1.2.1:500 RD v2:2 IP 60.1.2.1
933 60.1.2.1:500 RD v2:1 IP 60.1.2.1
937 60.1.1.1:500 RD v2:2 IP 60.1.1.1
936 60.1.1.1:500 RD v2:1 IP 60.1.1.1

Number of IKE SA : 4
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

---End

Configuration Files

- RouterA configuration file

```
#
 sysname RouterA
#
acl number 3002
 rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.3.0
 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
#
ike peer rut1
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
 ike-proposal 5
 remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
#
interface GigabitEthernet0/0/1
 ip address 60.1.1.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.1.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 60.1.1.2
ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
#
return
```

● RouterB configuration file

```
#
 sysname RouterB
#
acl number 3002
 rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.3.0
 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
#
ike peer rut1
 pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%^%#
 ike-proposal 5
 remote-address 60.1.3.1
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
#
interface GigabitEthernet0/0/1
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
```

```
ip address 192.168.2.2 255.255.255.0
#
ip route-static 60.1.3.0 255.255.255.0 60.1.2.2
ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
#
return
```

- RouterC configuration file

```
#
sysname RouterC
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
#
ike user-table
10
 user routerb
  id-type ip 60.1.2.1
  pre-shared-key %^%#@q!5$RKXkQN'Sc&0D}$ .T}vBUy,=TYy]rBOZl|04%^
%#
 user routera
  id-type ip 60.1.1.1
  pre-shared-key %^%#C&&/)4psiA%=7T"/!J)B|CuiH4$us1x3muJpnTr&%^%#
#
ike peer rut1
 ike-proposal 5
 user-table 10
#
ipsec policy-template use1 10
 ike-peer rut1
 proposal tran1
#
ipsec policy policy1 10 isakmp template use1
#
interface GigabitEthernet0/0/1
 ip address 60.1.3.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.3.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 60.1.3.2
#
return
```

6.12.6 Example for Establishing an IPsec Tunnel Between the Branch and Headquarters with a Redundant Gateway

Networking Requirements

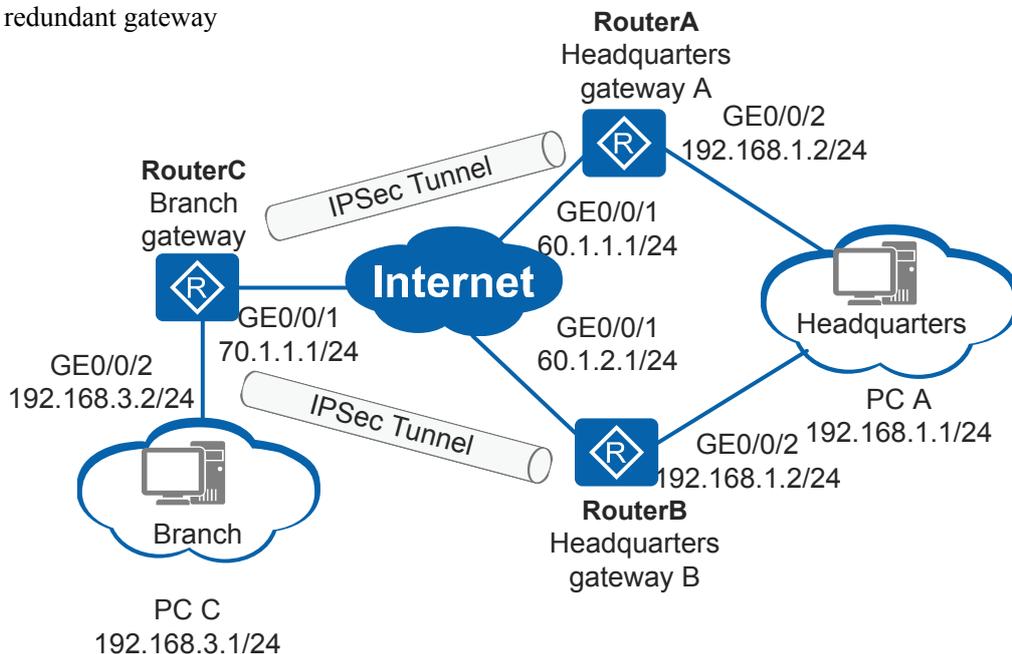
As shown in [Figure 6-47](#), two gateways RouterA and RouterB are deployed in the headquarters to improve security. RouterC in the branch communicates with the headquarters through the public network.

The enterprise requires to protect traffic transmitted over the public network between the enterprise branch and headquarters.

IPsec tunnels can be set up between the branch gateways and headquarters gateway because they communicate over the Internet. The branch gateway attempts to establish an IPsec tunnel

with the headquarters gateway RouterA. If the attempt fails, the branch gateway establishes an IPsec tunnel with the headquarters gateway RouterB.

Figure 6-47 Establishing an IPsec tunnel between the branch and headquarters with a redundant gateway



Configuration Roadmap

1. Configure the IP address and static route on each interface to implement communication between interfaces.
2. Configure an ACL to define the data flows to be protected by the IPsec tunnel.
3. Configure an IPsec proposal to define the traffic protection method.
4. Configure an IKE peer and define the attributes used for IKE negotiation.
5. Create an IPsec policy on RouterA, RouterB, and RouterC respectively to determine protection methods used for protecting different types of data flows. On RouterA and RouterB, IPsec policies are created through IPsec policy templates.
6. Apply an IPsec policy group to an interface so that the interface can protect traffic.

Procedure

Step 1 Configure an IP address and a static route for each interface on RouterA, RouterB, and RouterC to ensure that there are reachable routes among them.

Assign an IP address to each interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to the headquarters subnet is 60.1.1.2.

```
[RouterA] ip route-static 70.1.1.0 255.255.255.0 60.1.1.2  
[RouterA] ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
```

Assign an IP address to each interface on RouterB.

```
<Huawei> system-view  
[Huawei] sysname RouterB  
[RouterB] interface gigabitethernet 0/0/1  
[RouterB-GigabitEthernet0/0/1] ip address 60.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet0/0/1] quit  
[RouterB] interface gigabitethernet 0/0/2  
[RouterB-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0  
[RouterB-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to the headquarters subnet is 60.1.2.2.

```
[RouterB] ip route-static 70.1.1.0 255.255.255.0 60.1.2.2  
[RouterB] ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
```

Assign an IP address to each interface on RouterC.

```
<Huawei> system-view  
[Huawei] sysname RouterC  
[RouterC] interface gigabitethernet 0/0/1  
[RouterC-GigabitEthernet0/0/1] ip address 70.1.1.1 255.255.255.0  
[RouterC-GigabitEthernet0/0/1] quit  
[RouterC] interface gigabitethernet 0/0/2  
[RouterC-GigabitEthernet0/0/2] ip address 192.168.3.2 255.255.255.0  
[RouterC-GigabitEthernet0/0/2] quit
```

Configure a static route to the peer on RouterC. This example assumes that the next hop address in the route to the headquarters gateway A and B is 70.1.1.2.

```
[RouterC] ip route-static 0.0.0.0 0.0.0.0 70.1.1.2
```

Step 2 Configure an ACL on RouterA and RouterB to define the data flows to be protected.

NOTE

RouterA and RouterB create an IPsec policy through the IPsec policy template; therefore, this step is optional. If you configure an ACL on RouterA and RouterB, you must specify the destination address in the ACL rule.

Configure an ACL on RouterC to define the data flows from subnet 192.168.3.0/24 to subnet 192.168.1.0/24.

```
[RouterC] acl number 3002  
[RouterC-acl-adv-3002] rule permit ip source 192.168.3.0 0.0.0.255 destination  
192.168.1.0 0.0.0.255  
[RouterC-acl-adv-3002] quit
```

Step 3 Create an IPsec proposal on RouterA, RouterB, and RouterC respectively.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1  
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256  
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1  
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
```

```
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterB-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterC.

```
[RouterC] ipsec proposal tran1  
[RouterC-ipsec-proposal-tran1] esp authentication-algorithm sha2-256  
[RouterC-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterC-ipsec-proposal-tran1] quit
```

Run the **display ipsec proposal** command on RouterA, RouterB, and RouterC to view the configuration of the IPsec proposal. The command output on RouterA is used as an example.

```
[RouterA] display ipsec proposal name tran1  
  
IPsec proposal name: tran1  
Encapsulation mode: Tunnel  
Transform          : esp-new  
ESP protocol       : Authentication SHA2-HMAC-256  
                   Encryption     AES-128
```

Step 4 Create an IKE peer on RouterA, RouterB, and RouterC respectively.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5  
[RouterA-ike-proposal-5] encryption-algorithm aes-128  
[RouterA-ike-proposal-5] authentication-algorithm sha2-256  
[RouterA-ike-proposal-5] dh group14  
[RouterA-ike-proposal-5] quit
```

Create an IKE peer on RouterA.

```
[RouterA] ike peer rut1  
[RouterA-ike-peer-rut1] pre-shared-key cipher huawei@123  
[RouterA-ike-peer-rut1] ike-proposal 5  
[RouterA-ike-peer-rut1] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5  
[RouterB-ike-proposal-5] encryption-algorithm aes-128  
[RouterB-ike-proposal-5] authentication-algorithm sha2-256  
[RouterB-ike-proposal-5] dh group14  
[RouterB-ike-proposal-5] quit
```

Create an IKE peer on RouterB.

```
[RouterB] ike peer rut1  
[RouterB-ike-peer-rut1] pre-shared-key cipher huawei@123  
[RouterB-ike-peer-rut1] ike-proposal 5  
[RouterB-ike-peer-rut1] quit
```

NOTE

RouterA and RouterB function as responders to respond to an IKE negotiation request; therefore, they create IPsec policies through IPsec policy templates. You do not need to set **remote-address**.

Create an IKE peer on RouterC.

```
[RouterC] ike proposal 5  
[RouterC-ike-proposal-5] encryption-algorithm aes-128  
[RouterC-ike-proposal-5] authentication-algorithm sha2-256  
[RouterC-ike-proposal-5] dh group14  
[RouterC-ike-proposal-5] quit
```

Create an IKE peer on RouterC.

```
[RouterC] ike peer rut1  
[RouterC-ike-peer-rut1] ike-proposal 5
```

```
[RouterC-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterC-ike-peer-rut1] remote-address 60.1.1.1
[RouterC-ike-peer-rut1] remote-address 60.1.2.1
[RouterC-ike-peer-rut1] quit
```

Step 5 Create an IPsec policy on RouterA, RouterB, and RouterC respectively. On RouterA and RouterB, IPsec policies are created through IPsec policy templates.

Create an ipsec policy template on RouterA and apply the ipsec policy template to an IPsec policy.

```
[RouterA] ipsec policy-template use1 10
[RouterA-ipsec-policy-templet-use1-10] ike-peer rut1
[RouterA-ipsec-policy-templet-use1-10] proposal tran1
[RouterA-ipsec-policy-templet-use1-10] quit
[RouterA] ipsec policy policy1 10 isakmp template use1
```

Create an ipsec policy template on RouterB and apply the ipsec policy template to an IPsec policy.

```
[RouterB] ipsec policy-template use1 10
[RouterB-ipsec-policy-templet-use1-10] ike-peer rut1
[RouterB-ipsec-policy-templet-use1-10] proposal tran1
[RouterB-ipsec-policy-templet-use1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template use1
```

Create an IPsec policy on RouterC.

```
[RouterC] ipsec policy policy1 10 isakmp
[RouterC-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterC-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterC-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterC-ipsec-policy-isakmp-policy1-10] quit
```

Step 6 Apply an IPsec policy group to the interface of RouterA, RouterB, and RouterC.

Apply an IPsec policy group to the interface of RouterA.

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply an IPsec policy group to the interface of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
```

Step 7 Verify the configuration.

After the configurations are complete, PC C can ping PC A successfully. The data transmitted between PC C and PC A is encrypted.

Run the **display ike sa** command on RouterA and RouterB to view the IKE configuration. The command output on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer          VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  24366    70.1.1.1:500    RD|ST  v2:2    IP     70.1.1.1
  24274    70.1.1.1:500    RD|ST  v2:1    IP     70.1.1.1
```

```
Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE   RL--REPLACED   FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT   LKG--LAST KNOWN GOOD SEQ NO.   BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE   NEG--NEGOTIATING
```

Run the **display ike sa** command on RouterC. The command output is displayed as follows:

```
[RouterC] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
937 60.1.1.1:500 RD v2:2 IP 60.1.1.1
936 60.1.1.1:500 RD v2:1 IP 60.1.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE   RL--REPLACED   FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT   LKG--LAST KNOWN GOOD SEQ NO.   BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE   NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
ipsec proposal tran1
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
ike peer rut1
  pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n!$7`DEV&=G(=Et02P/%\*!%#
  ike-proposal 5
#
ipsec policy-template use1 10
  ike-peer rut1
  proposal tran1
#
ipsec policy policy1 10 isakmp template use1
#
interface GigabitEthernet0/0/1
  ip address 60.1.1.1 255.255.255.0
  ipsec policy policy1
#
interface GigabitEthernet0/0/2
  ip address 192.168.1.2 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
ip route-static 192.168.3.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
 ike-proposal 5
#
ipsec policy-template use1 10
 ike-peer rut1
 proposal tran1
#
ipsec policy policy1 10 isakmp template use1
#
interface GigabitEthernet0/0/1
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.1.2 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.2.2
ip route-static 192.168.3.0 255.255.255.0 60.1.2.2
#
return
```

- Configuration file of RouterC

```
#
sysname RouterC
#
acl number 3002
 rule 5 permit ip source 192.168.3.0 0.0.0.255 destination 192.168.1.0
 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%#IRFGEiFPJl$a'Qy,L*XQL_+*Grq-=yMb}ULZdS6%%#
 ike-proposal 5
 remote-address 60.1.1.1
 remote-address 60.1.2.1
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
#
```

```
interface GigabitEthernet0/0/1
ip address 70.1.1.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.3.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 70.1.1.2
#
return
```

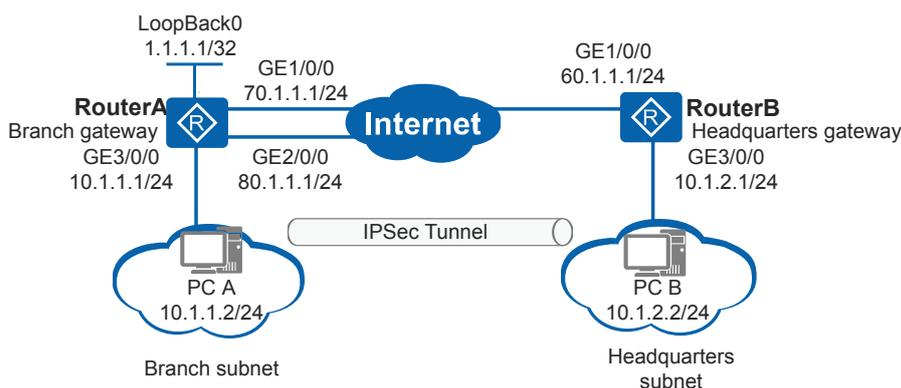
6.12.7 Example for Establishing an IPsec Tunnel Between the Enterprise Headquarters and Branch Using a Multi-Link Shared IPsec Policy Group

Networking Requirements

As shown in [Figure 6-48](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. RouterA uses two egress links in backup or load balancing mode. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The Enterprise wants to protect traffic between the branch subnet and headquarters subnet. If an active/standby switchover occurs or the egress link becomes faulty, IPsec services need to be smoothly switched. IPsec tunnels can be set up between the branch gateway and headquarters gateway because they communicate over the Internet. The two outbound interfaces negotiate with their peers to establish IPsec SAs respectively. When one interface alternates between Up and Down states and an active/standby switchover occurs, the two peers need to perform IKE negotiate again to generate IPsec SAs. The IKE re-negotiation causes IPsec service interruption in a short time. To ensure that IPsec services are smoothly switched, the two egress links on the branch gateway and the headquarters gateway only negotiate a shared IPsec SA.

Figure 6-48 Establishing an IPsec tunnel between the enterprise headquarters and branch using a multi-link shared IPsec policy group



Configuration Roadmap

The branch gateway uses a loopback interface to establish an IPsec tunnel with the headquarters gateway, and the two egress links and the headquarters gateway only negotiate a shared IPsec SA. The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure ACLs to define data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec policies and reference ACLs and IPsec proposals in the IPsec policies to determine the methods used to protect data flows.
6. Apply IPsec policy groups to interfaces. Configure a multi-link shared IPsec policy group on RouterA so that the IPsec policy group can be shared by multiple interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 70.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 80.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
[RouterA] interface gigabitethernet 3/0/0
[RouterA-GigabitEthernet3/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet3/0/0] quit
[RouterA] interface loopback 0
[RouterA-LoopBack0] ip address 1.1.1.1 255.255.255.255
[RouterA-LoopBack0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop addresses corresponding to the two outbound interfaces in the route to RouterB are 70.1.1.2 and 80.1.1.2.

```
[RouterA] ip route-static 10.1.2.0 255.255.255.0 70.1.1.2 preference 10
[RouterA] ip route-static 10.1.2.0 255.255.255.0 80.1.1.2 preference 20
[RouterA] ip route-static 60.1.1.0 255.255.255.0 70.1.1.2 preference 10
[RouterA] ip route-static 60.1.1.0 255.255.255.0 80.1.1.2 preference 20
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 3/0/0
[RouterB-GigabitEthernet3/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet3/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.1.2.

```
[RouterB] ip route-static 1.1.1.1 255.255.255.255 60.1.1.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.1.2
[RouterB] ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
[RouterB] ip route-static 80.1.1.0 255.255.255.0 60.1.1.2
```

Step 2 Configure ACLs on RouterA and RouterB to define data flows to be protected.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3101
[RouterA-acl-adv-3101] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[RouterA-acl-adv-3101] quit
```

Configure an ACL on RouterB to define data flows sent from 10.1.2.0/24 to 10.1.1.0/24.

```
[RouterB] acl number 3101
[RouterB-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[RouterB-acl-adv-3101] quit
```

Step 3 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal prop
[RouterA-ipsec-proposal-prop] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-prop] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-prop] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal prop
[RouterB-ipsec-proposal-prop] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop] quit
```

Step 4 Create IKE proposals on RouterA and RouterB.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Step 5 Configure IKE peers on RouterA and RouterB.

Configure an IKE peer on RouterA, reference the IKE proposal, and set the pre-shared key and remote ID.

```
[RouterA] ike peer rut
[RouterA-ike-peer-rut] undo version 2
[RouterA-ike-peer-rut] ike-proposal 5
[RouterA-ike-peer-rut] pre-shared-key cipher huawei
[RouterA-ike-peer-rut] remote-address 60.1.1.1
[RouterA-ike-peer-rut] quit
```

Configure an IKE peer on RouterB, reference the IKE proposal, and set the pre-shared key and remote ID.

```
[RouterB] ike peer rut
[RouterB-ike-peer-rut] undo version 2
[RouterB-ike-peer-rut] ike-proposal 5
[RouterB-ike-peer-rut] pre-shared-key cipher huawei
[RouterB-ike-peer-rut] remote-address 1.1.1.1
[RouterB-ike-peer-rut] quit
```

Step 6 Create IPsec policies on RouterA and RouterB.

Create an IPsec policy on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut
[RouterA-ipsec-policy-isakmp-policy1-10] proposal prop
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3101
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy on RouterB.

```
[RouterB] ipsec policy policy1 10 isakmp
[RouterB-ipsec-policy-isakmp-policy1-10] ike-peer rut
[RouterB-ipsec-policy-isakmp-policy1-10] proposal prop
[RouterB-ipsec-policy-isakmp-policy1-10] security acl 3101
[RouterB-ipsec-policy-isakmp-policy1-10] quit
```

Step 7 Apply IPsec policy groups to interfaces on RouterA and RouterB.

Configure a multi-link shared IPsec policy group on RouterA and apply the IPsec policy group to the two interfaces.

```
[RouterA] ipsec policy policy1 shared local-interface loopback 0
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec policy policy1
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ipsec policy policy1
[RouterA-GigabitEthernet2/0/0] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 8 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
937 60.1.1.1:500 RD|ST v1:2 IP 60.1.1.1
936 60.1.1.1:500 RD|ST v1:1 IP 60.1.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3101
rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
```

```
#
ipsec proposal prop
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
ike peer rut
  undo version 2
  pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
  ike-proposal 5
  remote-address 60.1.1.1
#
ipsec policy policy1 10 isakmp
  security acl 3101
  ike-peer rut
  proposal prop
#
ipsec policy policy1 shared local-interface LoopBack0
#
interface GigabitEthernet1/0/0
  ip address 70.1.1.1 255.255.255.0
  ipsec policy policy1
#
interface GigabitEthernet2/0/0
  ip address 80.1.1.1 255.255.255.0
  ipsec policy policy1
#
interface GigabitEthernet3/0/0
  ip address 10.1.1.1 255.255.255.0
#
interface LoopBack0
  ip address 1.1.1.1 255.255.255.255
#
ip route-static 10.1.2.0 255.255.255.0 70.1.1.2 preference 10
ip route-static 10.1.2.0 255.255.255.0 80.1.1.2 preference 20
ip route-static 60.1.1.0 255.255.255.0 70.1.1.2 preference 10
ip route-static 60.1.1.0 255.255.255.0 80.1.1.2 preference 20
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
acl number 3101
  rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
#
ipsec proposal prop
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
ike peer rut
  undo version 2
  pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%^%#
  ike-proposal 5
```

```
remote-address 1.1.1.1
#
ipsec policy policy1 10 isakmp
security acl 3101
ike-peer rut
proposal prop
#
interface GigabitEthernet1/0/0
ip address 60.1.1.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet3/0/0
ip address 10.1.2.1 255.255.255.0
#
ip route-static 1.1.1.1 255.255.255.255 60.1.1.2
ip route-static 10.1.1.0 255.255.255.0 60.1.1.2
ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
ip route-static 80.1.1.0 255.255.255.0 60.1.1.2
#
return
```

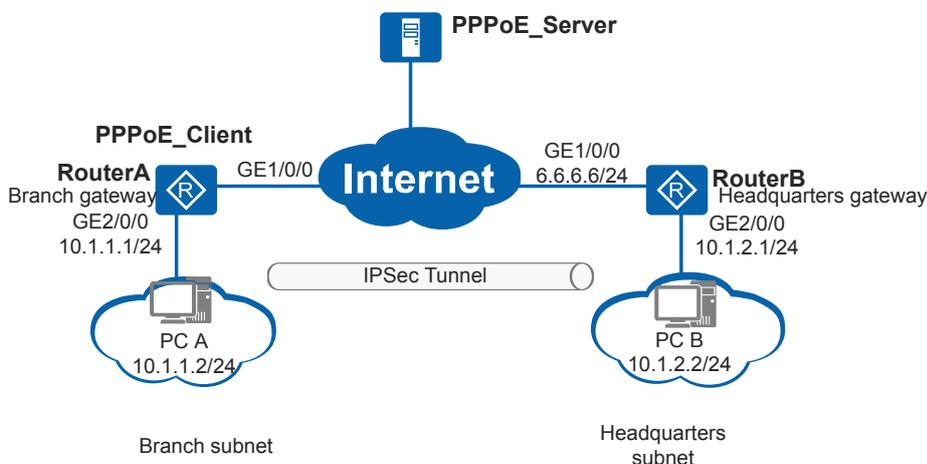
6.12.8 Example for Establishing an IPsec Tunnel Between the Enterprise Headquarters and Branch Through PPPoE

Networking Requirements

As shown in [Figure 6-49](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24. The branch gateway connects to the Internet using PPPoE, and obtains an IP address from the PPPoE server.

The enterprise wants to protect data flows between the branch subnets and the headquarters subnet. An IPsec tunnel can be set up between the branch gateway and headquarters gateway because they communicate over the Internet. The branch gateway functions as the PPPoE client to obtain an IP address, so the headquarters gateway cannot obtain the branch gateway's IP address and can only respond to IPsec negotiation requests initiated by the branch gateway.

Figure 6-49 Establishing an IPsec tunnel between the enterprise headquarters and branch through PPPoE



 NOTE

If both the branch gateway and headquarters gateway connect to the public network through PPPoE, the **remote-address host-name host-name** command must be run on them to specify the domain name for IPsec negotiation. Otherwise, the IPsec tunnel cannot be established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the PPPoE client on RouterA so that RouterA can obtain an IP address from the PPPoE server.
2. Configure the IKE negotiation mode in which an IPsec tunnel is set up. RouterB functions as the responder to receive IPsec negotiation requests initiated by RouterA.

Procedure

- Step 1** Configure the PPPoE client on RouterA so that RouterA can obtain an IP address from the PPPoE server.

Configure a dialer access group to permit all IPv4 packets to pass through.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] dialer-rule
[RouterA-dialer-rule] dialer-rule 1 ip permit
[RouterA-dialer-rule] quit
```

Create a dialer interface and set parameters of the dialer interface.

```
[RouterA] interface dialer 1
[RouterA-Dialer1] link-protocol ppp
[RouterA-Dialer1] ppp chap user user@huawei.com
[RouterA-Dialer1] ppp chap password cipher Huawei@1234
[RouterA-Dialer1] ip address ppp-negotiate
[RouterA-Dialer1] dialer user huawei
[RouterA-Dialer1] dialer bundle 1
[RouterA-Dialer1] dialer-group 1
[RouterA-Dialer1] quit
```

Bind the dialer interface to a physical interface and establish a PPPoE session.

```
[RouterA] interface gigabitethernet1/0/0
[RouterA-GigabitEthernet1/0/0] pppoe-client dial-bundle-number 1
[RouterA-GigabitEthernet1/0/0] quit
```

Assign IP addresses to interfaces.

```
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure LAN users to dial up using public network addresses translated through NAT.

```
[RouterA] acl number 3002
[RouterA-acl-adv-3002] rule 5 permit ip source 10.1.1.0 0.0.0.255
[RouterA-acl-adv-3002] quit
[RouterA] interface dialer 1
[RouterA-Dialer1] nat outbound 3002
[RouterA-Dialer1] quit
```

On RouterA, configure a static route to PC B. The route uses the IP address of the dialer interface as the next hop address.

```
[RouterA] ip route-static 6.6.6.0 24 dialer1  
[RouterA] ip route-static 10.1.2.0 24 dialer1
```

Step 2 On RouterA, set parameters for establishing an IPsec tunnel in IKE negotiation mode.

Configure an ACL to control data flows from the branch subnet 10.1.1.0/24 to the headquarters subnet 10.1.2.0/24. Assume that the NAT-translated branch subnet address is 1.1.1.0/24 (that is, public network address obtained through PPPoE dial-up).

```
[RouterA] acl number 3003  
[RouterA-acl-adv-3003] rule permit ip source 1.1.1.0 0.0.0.255 destination  
10.1.2.0 0.0.0.255  
[RouterA-acl-adv-3003] quit
```

Configure an IPsec proposal.

```
[RouterA] ipsec proposal prop1  
[RouterA-ipsec-proposal-prop1] esp authentication-algorithm sha2-256  
[RouterA-ipsec-proposal-prop1] esp encryption-algorithm aes-128  
[RouterA-ipsec-proposal-prop1] quit
```

Configure an IKE proposal.

```
[RouterA] ike proposal 5  
[RouterA-ike-proposal-5] encryption-algorithm aes-128  
[RouterA-ike-proposal-5] authentication-algorithm sha2-256  
[RouterA-ike-proposal-5] dh group14  
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer.

```
[RouterA] ike peer rut1  
[RouterA-ike-peer-rut1] undo version 2  
[RouterA-ike-peer-rut1] ike-proposal 5  
[RouterA-ike-peer-rut1] pre-shared-key cipher huawei@123  
[RouterA-ike-peer-rut1] remote-address 6.6.6.6  
[RouterA-ike-peer-rut1] quit
```

Configure an IPsec policy.

```
[RouterA] ipsec policy policy1 10 isakmp  
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut1  
[RouterA-ipsec-policy-isakmp-policy1-10] proposal prop1  
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3003  
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Run the **display ipsec policy** command to view the IPsec policy configuration.

Apply the IPsec policy group to the dialer interface.

```
[RouterA] interface dialer 1  
[RouterA-Dialer1] ipsec policy policy1  
[RouterA-Dialer1] quit
```

Step 3 On RouterB used as the responder, set parameters for establishing an IPsec tunnel in IKE negotiation mode.

Configure an IP address for an interface and a static route to the peer.

```
<Huawei> system-view  
[Huawei] sysname RouterB  
[RouterB] interface gigabitethernet 1/0/0  
[RouterB-GigabitEthernet1/0/0] ip address 6.6.6.6 255.255.255.0  
[RouterB-GigabitEthernet1/0/0] quit  
[RouterB] interface gigabitethernet 2/0/0  
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer. This example assumes that the next hop address in the route is 6.6.6.254.

```
[RouterB] ip route-static 0.0.0.0 0.0.0.0 6.6.6.254
```

Configure an IPsec proposal.

```
[RouterB] ipsec proposal prop1
[RouterB-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop1] quit
```

Configure an IKE proposal.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer.

Because RouterB as the responder uses an IPsec policy template to configure an IPsec policy, so you do not need to specify the remote IP address for the IKE peer.

```
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] undo version 2
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] pre-shared-key cipher huawei@123
[RouterB-ike-peer-rut1] quit
```

Configure an IPsec policy template.

```
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterB-ipsec-policy-templet-temp1-10] proposal prop1
[RouterB-ipsec-policy-templet-temp1-10] quit
```

Run the **display ipsec policy-template** command to view the IPsec policy template configuration.

Reference the IPsec policy template in the IPsec policy.

```
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Apply the IPsec policy group to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 4 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
Conn-ID  Peer          VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
 246     6.6.6.6:500        RD|ST  v1:2     IP     6.6.6.6
 245     6.6.6.6:500        RD|ST  v1:1     IP     6.6.6.6

Number of IKE SA : 2
-----
```

```
Flag Description:
RD--READY      ST--STAYALIVE    RL--REPLACED   FD--FADING     TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE      S--STANDBY      A--ALONE       NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3002
 rule 5 permit ip source 10.1.1.0 0.0.0.255
acl number 3003
 rule 5 permit ip source 1.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec proposal prop1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 undo version 2
 pre-shared-key cipher %%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%#
 ike-proposal 5
 remote-address 6.6.6.6
#
ipsec policy policy1 10 isakmp
 security acl 3003
 ike-peer rut1
 proposal prop1
#
interface Dialer1
 link-protocol ppp
 ppp chap user user@huawei.com
 ppp chap password cipher %%%^_PfANXK0(,Jr-(3p]"R,eOL%@@
 ip address ppp-negotiate
 dialer user huawei
 dialer bundle 1
 dialer-group 1
 nat outbound 3002
 ipsec policy policy1
#
interface GigabitEthernet1/0/0
 pppoe-client dial-bundle-number 1
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
dialer-rule
 dialer-rule 1 ip permit
#
ip route-static 6.6.6.0 255.255.255.0 dialer1
ip route-static 10.1.2.0 255.255.255.0 Dialer1
#
return
```

- Configuration file of RouterB

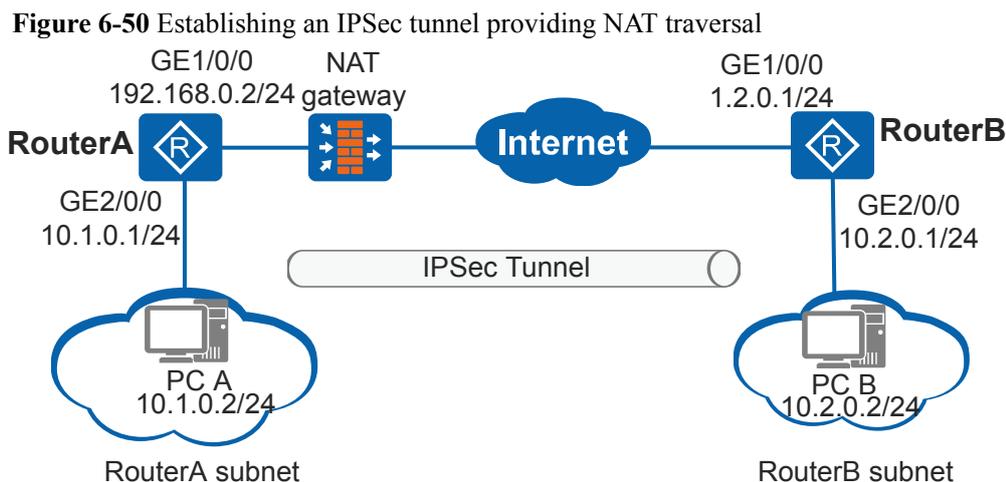
```
#
sysname RouterB
#
ipsec proposal prop1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 undo version 2
 pre-shared-key cipher %%K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
 ike-proposal 5
#
ipsec policy-template templ
 ike-peer rut1
 proposal prop1
#
ipsec policy policy1 10 isakmp template templ
#
interface GigabitEthernet1/0/0
 ip address 6.6.6.6 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 6.6.6.254
#
return
```

6.12.9 Example for Establishing an IPsec Tunnel Through NAT Traversal

Networking Requirements

As shown in [Figure 6-50](#), RouterA and RouterB communicate through the NAT gateway. RouterA is located on the subnet at 10.1.0.2/24, and RouterB is located on the subnet at 10.2.0.2/24.

The enterprise wants to protect traffic exchanged between RouterA and RouterB.



Configuration Roadmap

RouterA and RouterB communicate through the NAT gateway, so NAT traversal must be enabled for establishing an IPsec tunnel. The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure an ACL on RouterA to define data flows to be protected.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec policies on RouterA and RouterB. RouterB uses an IPsec policy template to create an IPsec policy.
6. Apply IPsec policy groups to interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 192.168.0.2 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.0.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 192.168.0.1.

```
[RouterA] ip route-static 0.0.0.0 0.0.0.0 192.168.0.1
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
```

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 1.2.0.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.2.0.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 1.2.0.2.

```
[RouterB] ip route-static 10.1.0.0 255.255.255.0 1.2.0.2
[RouterB] ip route-static 192.168.0.0 255.255.255.0 1.2.0.2
```

Step 2 # Configure an ACL on RouterA to define data flows sent from 10.1.0.0/24 to 10.2.0.0/24.

```
[RouterA] acl number 3101
[RouterA-acl-adv-3101] rule permit ip source 10.1.0.0 0.0.0.255 destination
10.2.0.0 0.0.0.255
[RouterA-acl-adv-3101] quit
```

Step 3 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
```

Step 4 Set the local ID type to **name** on RouterA and RouterB.

Set the local ID type to **name** on RouterA.

```
[RouterA] ike local-name rta
```

Set the local ID type to **name** on RouterB.

```
[RouterB] ike local-name rtb
```

Step 5 Configure IKE peers on RouterA and RouterB.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer rta
[RouterA-ike-peer-rta] undo version 2
[RouterA-ike-peer-rta] exchange-mode aggressive
[RouterA-ike-peer-rta] ike-proposal 5
[RouterA-ike-peer-rta] pre-shared-key cipher huawei@123
[RouterA-ike-peer-rta] local-id-type fqdn
[RouterA-ike-peer-rta] remote-address 1.2.0.1
[RouterA-ike-peer-rta] remote-id rtb
[RouterA-ike-peer-rta] nat traversal
[RouterA-ike-peer-rta] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

```
[RouterB] ike peer rtb
[RouterB-ike-peer-rtb] undo version 2
[RouterB-ike-peer-rtb] exchange-mode aggressive
[RouterB-ike-peer-rtb] ike-proposal 5
[RouterB-ike-peer-rtb] pre-shared-key cipher huawei@123
[RouterB-ike-peer-rtb] local-id-type fqdn
[RouterB-ike-peer-rtb] remote-id rta
[RouterA-ike-peer-rta] nat traversal
[RouterB-ike-peer-rtb] quit
```

Step 6 Create IPsec policies on RouterA and RouterB.

Create an IPsec policy in IKE negotiation mode on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3101
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rta
[RouterA-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy in IKE negotiation mode on RouterB.

```
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rtb
[RouterB-ipsec-policy-templet-temp1-10] proposal tran1
[RouterB-ipsec-policy-templet-temp1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Run the **display ipsec policy** command on RouterA and RouterB to view the configurations of the IPsec policies.

Step 7 Apply IPsec policy groups to interfaces on RouterA and RouterB.

Apply the IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec policy policy1
[RouterA-GigabitEthernet1/0/0] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 8 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
15 1.2.0.1:4500 RD|ST v1:2 FQDN rtb
14 1.2.0.1:4500 RD|ST v1:1 FQDN rtb
```

```
Number of IKE SA : 2
```

```
-----  
Flag Description:  
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT  
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP  
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

---End

Configuration Files

- Configuration file of RouterA

```
#  
sysname RouterA  
#  
ike local-name rta  
#  
acl number 3101  
rule 5 permit ip source 10.1.0.0 0.0.0.255 destination 10.2.0.0 0.0.0.255  
#  
ipsec proposal tran1  
esp authentication-algorithm sha2-256  
esp encryption-algorithm aes-128  
#  
ike proposal 5  
encryption-algorithm aes-128  
dh group14  
authentication-algorithm sha2-256  
authentication-method pre-share  
integrity-algorithm hmac-sha2-256  
prf hmac-sha2-256  
#  
ike peer rta  
undo version 2  
exchange-mode aggressive  
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#  
ike-proposal 5  
local-id-type fqdn  
remote-id rtb  
remote-address 1.2.0.1  
#  
ipsec policy policy1 10 isakmp  
security acl 3101  
ike-peer rta  
proposal tran1  
#  
interface GigabitEthernet1/0/0  
ip address 192.168.0.2 255.255.255.0  
ipsec policy policy1  
#  
interface GigabitEthernet2/0/0  
ip address 10.1.0.1 255.255.255.0  
#  
ip route-static 0.0.0.0 0.0.0.0 192.168.0.1  
#  
return
```

- Configuration file of RouterB

```
#  
sysname RouterB  
#  
ike local-name rtb  
#  
ipsec proposal tran1  
esp authentication-algorithm sha2-256  
esp encryption-algorithm aes-128
```

```
#
ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
ike peer rtb
  undo version 2
  exchange-mode aggressive
  pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
  ike-proposal 5
  local-id-type fqdn
  remote-id rta
#
ipsec policy-template temp1 10
  ike-peer rtb
  proposal tran1
#
ipsec policy policy1 10 isakmp template temp1
#
interface GigabitEthernet1/0/0
  ip address 1.2.0.1 255.255.255.0
  ipsec policy policy1
#
interface GigabitEthernet2/0/0
  ip address 10.2.0.1 255.255.255.0
#
ip route-static 10.1.0.0 255.255.255.0 1.2.0.2
ip route-static 192.168.0.0 255.255.255.0 1.2.0.2
#
return
```

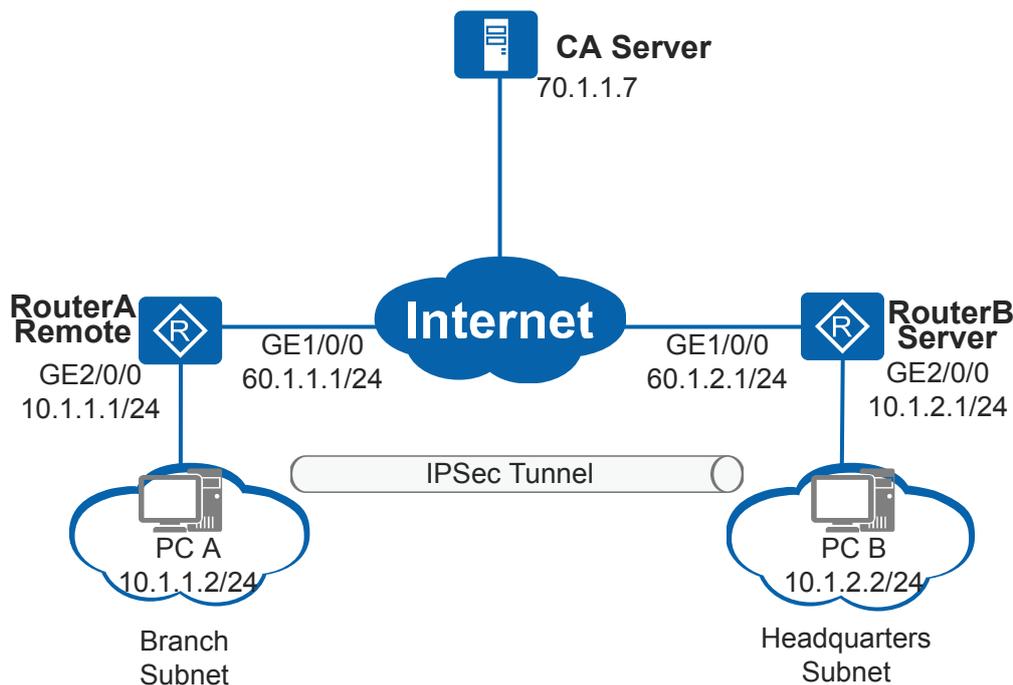
6.12.10 Example for Establishing an IPsec Tunnel in IKE Negotiation Mode by Specifying DNs

Networking Requirements

As shown in [Figure 6-51](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The headquarters gateway and branch gateway apply for digital certificates from the CA server. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise wants to use distinguished names (DNs) to identify identities and create SAs to protect data flows between the branch subnet and the headquarters subnet.

Figure 6-51 Establishing an IPsec tunnel in IKE negotiation mode by specifying DNs



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Apply for digital certificates from the CA server. The digital certificates are used for RSA signature authentication.
3. Configure ACLs to define data flows to be protected.
4. Configure IPsec proposals to define the method used to protect IPsec traffic.
5. Configure IKE peers to define IKE negotiation attributes.
6. Configure IPsec policies and reference ACLs, IPsec proposals, and IKE peers in the IPsec policies to define protection methods for data flows between RouterA and RouterB.
7. Apply IPsec policy groups to interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 60.1.1.2.

```
[RouterA] ip route-static 60.1.2.0 255.255.255.0 60.1.1.2  
[RouterA] ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view  
[Huawei] sysname RouterB  
[RouterB] interface gigabitethernet 1/0/0  
[RouterB-GigabitEthernet1/0/0] ip address 60.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet1/0/0] quit  
[RouterB] interface gigabitethernet 2/0/0  
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.2.2.

```
[RouterB] ip route-static 60.1.1.0 255.255.255.0 60.1.2.2  
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
```

Step 2 Configure ACLs on RouterA and RouterB to define data flows to be protected.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3001  
[RouterA-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255 destination  
10.1.2.0 0.0.0.255  
[RouterA-acl-adv-3001] quit
```

Configure an ACL on RouterB to define data flows sent from 10.1.2.0/24 to 10.1.1.0/24.

```
[RouterB] acl number 3001  
[RouterB-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255 destination  
10.1.1.0 0.0.0.255  
[RouterB-acl-adv-3001] quit
```

Step 3 Configure PKI entities and PKI domains on RouterA and RouterB, which are used to apply for digital certificates from the CA server.

Create an RSA key pair on RouterA.

```
[RouterA] pki rsa local-key-pair create rsa_scep exportable  
Info: The name of the new key-pair will be: rsa_scep  
The size of the public key ranges from 512 to  
4096.  
Input the bits in the modules:2048  
Generating key-  
pairs...  
.....+++  
.....+++
```

Configure a PKI entity on RouterA.

```
[RouterA] pki entity rta  
[RouterA-pki-entity-rta] country CN  
[RouterA-pki-entity-rta] state jiangsu  
[RouterA-pki-entity-rta] locality nanjing  
[RouterA-pki-entity-rta] organization huawei  
[RouterA-pki-entity-rta] organization-unit VPN  
[RouterA-pki-entity-rta] common-name ipsec  
[RouterA-pki-entity-rta] quit
```

Configure a PKI domain on RouterA.

```
[RouterA] pki realm rta  
[RouterA-pki-realm-rta] ca id ca_root
```

```
[RouterA-pki-realm-rta] enrollment-url http://70.1.1.7:8080/certsrv/mscep/  
mscep.dll times 4 ra  
[RouterA-pki-realm-rta] entity rta  
[RouterA-pki-realm-rta] rsa local-key-pair rsa_scep  
[RouterA-pki-realm-rta] fingerprint sha256  
e71add0744360e91186b828412d279e06dcc15a4ab4bb3d13842820396b526a0  
[RouterA-pki-realm-rta] password cipher 6AE73F21E6D3571D
```

Create an RSA key pair on RouterB.

```
[RouterB] pki rsa local-key-pair create rsa_scep exportable  
Info: The name of the new key-pair will be: rsa_scep  
The size of the public key ranges from 512 to  
4096.  
Input the bits in the modules:2048  
Generating key-  
pairs...  
.....+++  
.....+++
```

Configure a PKI entity on RouterB.

```
[RouterB] pki entity rtb  
[RouterB-pki-entity-rtb] country CN  
[RouterB-pki-entity-rtb] state jiangsu  
[RouterB-pki-entity-rtb] locality nanjing  
[RouterB-pki-entity-rtb] organization huawei  
[RouterB-pki-entity-rtb] organization-unit VPN  
[RouterB-pki-entity-rtb] common-name ipsec  
[RouterB-pki-entity-rtb] quit
```

Configure a PKI domain on RouterB.

```
[RouterB] pki realm rtb  
[RouterB-pki-realm-rtb] ca id ca_root  
[RouterB-pki-realm-rtb] enrollment-url http://70.1.1.7:8080/certsrv/mscep/  
mscep.dll times 4 ra  
[RouterB-pki-realm-rtb] entity rtb  
[RouterB-pki-realm-rtb] rsa local-key-pair rsa_scep  
[RouterB-pki-realm-rtb] fingerprint sha256  
e71add0744360e91186b828412d279e06dcc15a4ab4bb3d13842820396b526a0  
[RouterB-pki-realm-rtb] password cipher 6AE73F21E6D3571D
```

Step 4 Request digital certificates on RouterA and RouterB.

Request a local certificate on RouterA.

```
[RouterA-pki-realm-rta] auto-enroll 60 regenerate 2048  
[RouterA-pki-realm-rta] quit
```

Before obtaining and installing a local certificate, the device obtains and installs a CA certificate first. The CA and local certificates are named **rta_ca.cer** and **rta_local.cer**.

Request a local certificate on RouterB.

```
[RouterB-pki-realm-rtb] auto-enroll 60 regenerate 2048  
[RouterB-pki-realm-rtb] quit
```

Before obtaining and installing a local certificate, the device obtains and installs a CA certificate first. The CA and local certificates are named **rtb_ca.cer** and **rtb_local.cer**.

Step 5 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal prop  
[RouterA-ipsec-proposal-prop] esp authentication-algorithm sha2-256  
[RouterA-ipsec-proposal-prop] esp encryption-algorithm aes-128  
[RouterA-ipsec-proposal-prop] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal prop
[RouterB-ipsec-proposal-prop] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop] quit
```

Step 6 Configure IKE peers on RouterA and RouterB.

Configure an IKE proposal that defines RSA signature authentication on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] authentication-method rsa-signature
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer rta
[RouterA-ike-peer-rta] undo version 2
[RouterA-ike-peer-rta] ike-proposal 5
[RouterA-ike-peer-rta] local-id-type dn
[RouterA-ike-peer-rta] pki realm rta
[RouterA-ike-peer-rta] remote-address 60.1.2.1
[RouterA-ike-peer-rta] quit
```

Configure an IKE proposal that defines RSA signature authentication on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] authentication-method rsa-signature
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

```
[RouterB] ike peer rtb
[RouterB-ike-peer-rtb] undo version 2
[RouterB-ike-peer-rtb] ike-proposal 5
[RouterB-ike-peer-rtb] local-id-type dn
[RouterB-ike-peer-rtb] pki realm rtb
[RouterB-ike-peer-rtb] remote-address 60.1.1.1
[RouterB-ike-peer-rtb] quit
```

Step 7 Create IPsec policies on RouterA and RouterB.

Create an IPsec policy in IKE negotiation mode on RouterA.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rta
[RouterA-ipsec-policy-isakmp-policy1-10] proposal prop
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3001
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Create an IPsec policy in IKE negotiation mode on RouterB.

```
[RouterB] ipsec policy policy1 10 isakmp
[RouterB-ipsec-policy-isakmp-policy1-10] ike-peer rtb
[RouterB-ipsec-policy-isakmp-policy1-10] proposal prop
[RouterB-ipsec-policy-isakmp-policy1-10] security acl 3001
[RouterB-ipsec-policy-isakmp-policy1-10] quit
```

Step 8 Apply IPsec policy groups to interfaces on RouterA and RouterB.

Apply the IPsec policy group to the interface of RouterA

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec policy policy1
[RouterA-GigabitEthernet1/0/0] quit
```

Apply the IPsec policy group to the interface of RouterB.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 9 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
  -----  -
      4    60.1.2.1:500              RD|ST    v1:2    DN      C=CN, ST=jiangsu,
L=nanjing, O=huawei, OU=VPN, CN=ipsec
      3    60.1.2.1:500              RD|ST    v1:1    DN      C=CN, ST=jiangsu,
L=nanjing, O=huawei, OU=VPN, CN=ipsec

Number of IKE SA : 2
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
pki entity rta
country CN
state jiangsu
locality nanjing
organization huawei
organization-unit VPN
common-name ipsec
#
pki realm rta
ca id ca_root
enrollment-url http://70.1.1.7:8080/certsrv/mscep/mscep.dll times 4 ra
entity rta
fingerprint sha256
e71add0744360e91186b828412d279e06dcc15a4ab4bb3d13842820396b526a0
rsa local-key-pair
rsa_scep
password cipher %$%$1HN-bn(k;^|0850AtYF3(M4%$%
$
auto-enroll 60 regenerate
#
acl number 3001
rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
```

```
ipsec proposal prop
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method rsa-signature
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
ike peer rta
  undo version 2
  ike-proposal 5
  local-id-type dn
  remote-address 60.1.2.1
  pki realm rta
#
ipsec policy policy1 10 isakmp
  security acl 3001
  ike-peer rta
  proposal prop
#
interface GigabitEthernet1/0/0
  ip address 60.1.1.1 255.255.255.0
  ipsec policy policy1
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.1 255.255.255.0
#
ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
pki entity rtb
  country CN
  state jiangsu
  locality nanjing
  organization huawei
  organization-unit VPN
  common-name ipsec
#
pki realm rtb
  ca id ca_root
  enrollment-url http://70.1.1.7:8080/certsrv/mscep/mscep.dll times 4 ra
  entity rtb
  fingerprint sha256
e71add0744360e91186b828412d279e06dcc15a4ab4bb3d13842820396b526a0
  rsa local-key-pair
  rsa scep
  password cipher %$%$\1HN-bn(k;^|0850AtYF3(M4%$%
$
  auto-enroll 60 regenerate
#
acl number 3001
  rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
#
ipsec proposal prop
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
ike proposal 5
  encryption-algorithm aes-128
```

```
dh group14
authentication-algorithm sha2-256
authentication-method rsa-signature
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rtb
undo version 2
ike-proposal 5
local-id-type dn
remote-address 60.1.1.1
pki realm rtb
#
ipsec policy policy1 10 isakmp
security acl 3001
ike-peer rtb
proposal prop
#
interface GigabitEthernet1/0/0
ip address 60.1.2.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
#
return
```

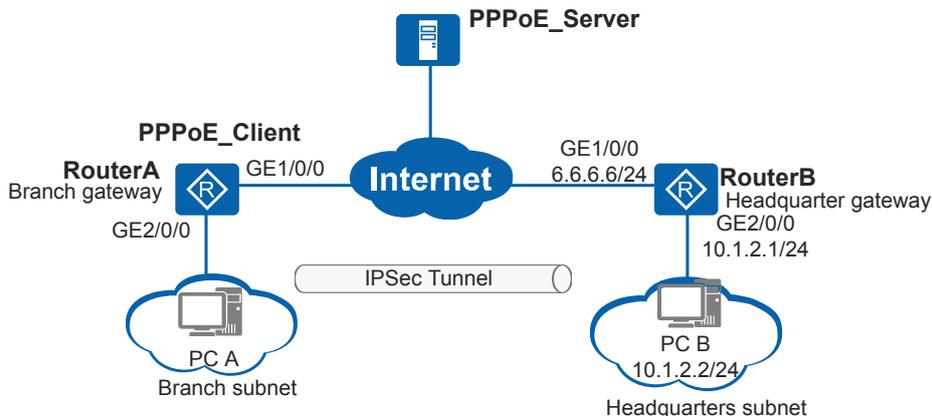
6.12.11 Example for Establishing an IPsec Tunnel Through Negotiation Initiated by the Branch User That Dynamically Obtains an IP Address

Networking Requirements

As shown in [Figure 6-52](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet IP addresses are allocated by the branch gateway through DHCP, and the headquarters subnet is located on 10.1.2.0/24. The branch gateway connects to the Internet using PPPoE, and obtains an IP address from the PPPoE_server.

The enterprise wants to protect data flows between the branch subnet and the headquarters subnet. An IPsec tunnel can be set up between the branch gateway and headquarters gateway because they communicate over the Internet. The branch gateway functions as the PPPoE client to obtain an IP address, so the headquarters gateway cannot obtain the branch gateway's IP address and can only respond to IPsec negotiation request sent by the branch gateway.

Figure 6-52 Establishing an IPsec tunnel through negotiation initiated by the branch user that dynamically obtains an IP address



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the PPPoE client on RouterA so that RouterA can obtain an IP address from the PPPoE server.
2. Enable DHCP on RouterA so that IP addresses can be dynamically allocated through DHCP.
3. Configure the IKE negotiation mode in which an IPsec tunnel is set up. RouterB functions as the responder to receive IPsec negotiation requests initiated by RouterA.

Procedure

- Step 1** Configure the PPPoE client on RouterA so that RouterA can obtain an IP address from the PPPoE server.

Configure a dialer access group to permit all IPv4 packets to pass through.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] dialer-rule
[RouterA-dialer-rule] dialer-rule 1 ip permit
[RouterA-dialer-rule] quit
```

Create a dialer interface and set parameters of the dialer interface.

```
[RouterA] interface dialer 1
[RouterA-Dialer1] link-protocol ppp
[RouterA-Dialer1] ppp chap user user@huawei.com
[RouterA-Dialer1] ppp chap password cipher Huawei@1234
[RouterA-Dialer1] ip address ppp-negotiate
[RouterA-Dialer1] dialer user huawei
[RouterA-Dialer1] dialer bundle 1
[RouterA-Dialer1] dialer-group 1
[RouterA-Dialer1] quit
```

Bind the dialer interface to a physical interface and establish a PPPoE session.

```
[RouterA] interface gigabitethernet1/0/0
[RouterA-GigabitEthernet1/0/0] pppoe-client dial-bundle-number 1
[RouterA-GigabitEthernet1/0/0] quit
```

On RouterA, configure a static route to PC B. The route uses the IP address of Dialer1 as the next hop address.

```
[RouterA] ip route-static 6.6.6.0 24 dialer1
[RouterA] ip route-static 10.1.2.0 24 dialer1
```

Step 2 Enable DHCP on RouterA so that IP addresses can be dynamically allocated through DHCP.

Enable DHCP and configure a global address pool.

```
[RouterA] dhcp enable
[RouterA] ip pool pooltest
[RouterA-ip-pool-pooltest] network 10.1.1.0 mask 255.255.255.0
[RouterA-ip-pool-pooltest] quit
```

Configure RouterA to assign IP addresses from a global address pool on an interface.

```
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] dhcp select global
[RouterA-GigabitEthernet2/0/0] quit
```

Step 3 On RouterA, set parameters for establishing an IPsec tunnel in IKE negotiation mode.

Configure an ACL to define data flows destined for 10.1.2.0/24.

```
[RouterA] acl number 3003
[RouterA-acl-adv-3003] rule permit ip destination 10.1.2.0 0.0.0.255
[RouterA-acl-adv-3003] quit
```

Configure an IPsec proposal.

```
[RouterA] ipsec proposal prop1
[RouterA-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-prop1] quit
```

Configure an IKE proposal.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer.

```
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] pre-shared-key cipher Huawei@1234
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] remote-address 6.6.6.6
[RouterA-ike-peer-rut1] quit
```

Configure an IPsec policy.

```
[RouterA] ipsec policy policy1 10 isakmp
[RouterA-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterA-ipsec-policy-isakmp-policy1-10] proposal prop1
[RouterA-ipsec-policy-isakmp-policy1-10] security acl 3003 dynamic-source
[RouterA-ipsec-policy-isakmp-policy1-10] quit
```

Apply the IPsec policy group to the dialer interface.

```
[RouterA] interface dialer 1
[RouterA-Dialer1] ipsec policy policy1
[RouterA-Dialer1] quit
```

Step 4 On RouterB used as the responder, set parameters for establishing an IPsec tunnel in IKE negotiation mode.

Configure an IP address for an interface and a static route to the peer.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 6.6.6.6 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to peer. This example assumes that the next hop address in the route is 6.6.6.1.

```
[RouterB] ip route-static 10.1.1.0 255.255.255.0 6.6.6.1
```

Configure an IPsec proposal.

```
[RouterB] ipsec proposal prop1
[RouterB-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop1] quit
```

Configure an IKE proposal.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer.

Because RouterB as the responder uses an IPsec policy template to configure an IPsec policy, so you do not need to specify the remote IP address for the IKE peer.

```
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] pre-shared-key cipher Huawei@1234
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] quit
```

Configure an IPsec policy template.

```
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterB-ipsec-policy-templet-temp1-10] proposal prop1
[RouterB-ipsec-policy-templet-temp1-10] quit
```

Reference the IPsec policy template in the IPsec policy.

```
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Apply the IPsec policy group to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 5 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. Data exchanged between PC A and PC B is encrypted. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
246 6.6.6.6:500 RD|ST v2:2 IP 6.6.6.6
245 6.6.6.6:500 RD|ST v2:1 IP 6.6.6.6

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3003
rule 5 permit ip destination 10.1.2.0 0.0.0.255
#
ipsec proposal prop1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&G(=Et02P/%\*!%^%#
ike-proposal 5
remote-address 6.6.6.6
#
ipsec policy policy1 10 isakmp
security acl 3003 dynamic-source
ike-peer rut1
proposal prop1
#
interface Dialer1
link-protocol ppp
ppp chap user user@huawei.com
ppp chap password cipher %%%^_PfANXK0(, Jr-(3p]"R, eOL%%%%
policy policy1
#
interface GigabitEthernet1/0/0
pppoe-client dial-bundle-number 1
#
interface GigabitEthernet2/0/0
dhcp select global
#
dialer-rule
dialer-rule 1 ip permit
#
ip route-static 6.6.6.0 255.255.255.0 dialer1
ip route-static 10.1.2.0 255.255.255.0 Dialer1
#
return
```

- Configuration file of RouterB

```
#
 sysname RouterB
#
 ipsec proposal prop1
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
 ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
 ike peer rut1
  pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
  ike-proposal 5
#
 ipsec policy-template templ
  ike-peer rut1
  proposal prop1
#
 ipsec policy policy1 10 isakmp template templ
#
 interface Ethernet1/0/0
  ip address 6.6.6.6 255.255.255.0
  ipsec policy policy1
#
 interface Ethernet2/0/0
  ip address 10.1.2.1 255.255.255.0
#
 ip route-static 10.1.1.0 255.255.255.0 6.6.6.1
#
 return
```

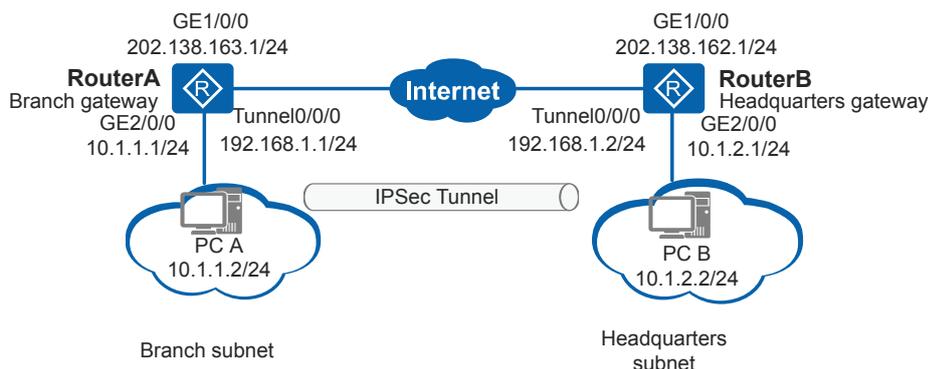
6.12.12 Example for Establishing an IPsec Tunnel Using a Tunnel Interface

Networking Requirements

As shown in [Figure 6-53](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise wants to protect data flows between the branch subnet and the headquarters subnet. An IPsec tunnel can be set up between the branch gateway and headquarters gateway because they communicate over the Internet. There are many branch subnets and many data flows need to be protected by IPsec. You can use a tunnel interface to create an IPsec tunnel to protect IPsec packets. You do not need to configure ACLs to define traffic characteristics.

Figure 6-53 Establishing an IPsec tunnel using a tunnel interface



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure IPsec proposals to define the method used to protect IPsec traffic.
3. Configure IKE peers to define IKE negotiation attributes.
4. Configure IPsec profiles and reference IPsec proposals and IKE peers in the IPsec profiles to determine the methods used to protect data flows.
5. Apply IPsec profiles to IPsec tunnel interfaces.
6. Configure static routes on IPsec tunnel interfaces and import data flows to be protected by IPsec to the tunnel interfaces.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 202.138.163.2.

```
[RouterA] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
```

```
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 202.138.162.2.

```
[RouterB] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
```

Step 2 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1  
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256  
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1  
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256  
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterB-ipsec-proposal-tran1] quit
```

Run the **display ipsec proposal** command on RouterA and RouterB to view the IPsec proposal configuration.

Step 3 Configure IKE peers on RouterA and RouterB.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5  
[RouterA-ike-proposal-5] authentication-algorithm sha2-256  
[RouterA-ike-proposal-5] encryption-algorithm aes-128  
[RouterA-ike-proposal-5] dh group14  
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer spub  
[RouterA-ike-peer-spub] ike-proposal 5  
[RouterA-ike-peer-spub] pre-shared-key cipher Huawei@1234  
[RouterA-ike-peer-spub] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ike proposal 5  
[RouterB-ike-proposal-5] authentication-algorithm sha2-256  
[RouterB-ike-proposal-5] encryption-algorithm aes-128  
[RouterB-ike-proposal-5] dh group14  
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

```
[RouterB] ike peer spua  
[RouterB-ike-peer-spua] ike-proposal 5  
[RouterB-ike-peer-spua] pre-shared-key cipher Huawei@1234  
[RouterB-ike-peer-spua] quit
```

Step 4 Create IPsec profiles on RouterA and RouterB.

Create an IPsec profile on RouterA.

```
[RouterA] ipsec profile profile1  
[RouterA-ipsec-profile-profile1] proposal tran1  
[RouterA-ipsec-profile-profile1] ike-peer spub  
[RouterA-ipsec-profile-profile1] quit
```

Create an IPsec profile on RouterB.

```
[RouterB] ipsec profile profile1
[RouterB-ipsec-profile-profile1] proposal tran1
[RouterB-ipsec-profile-profile1] ike-peer spua
[RouterB-ipsec-profile-profile1] quit
```

Step 5 Apply the IPsec profiles to IPsec tunnel interfaces on RouterA and RouterB.

Apply the IPsec profile to the interface of RouterA.

```
[RouterA] interface tunnel 0/0/0
[RouterA-Tunnel0/0/0] ip address 192.168.1.1 255.255.255.0
[RouterA-Tunnel0/0/0] tunnel-protocol ipsec
[RouterA-Tunnel0/0/0] source 202.138.163.1
[RouterA-Tunnel0/0/0] destination 202.138.162.1
[RouterA-Tunnel0/0/0] ipsec profile profile1
[RouterA-Tunnel0/0/0] quit
```

Apply the IPsec policy to the interface of RouterB.

```
[RouterB] interface tunnel 0/0/0
[RouterB-Tunnel0/0/0] ip address 192.168.1.2 255.255.255.0
[RouterB-Tunnel0/0/0] tunnel-protocol ipsec
[RouterB-Tunnel0/0/0] source 202.138.162.1
[RouterB-Tunnel0/0/0] destination 202.138.163.1
[RouterB-Tunnel0/0/0] ipsec profile profile1
[RouterB-Tunnel0/0/0] quit
```

Run the **display ipsec profile** command on RouterA and RouterB to view the IPsec profile configuration.

Step 6 Configure static routes on IPsec tunnel interfaces and import data flows to be protected by IPsec to the tunnel interfaces.

Configure a static route on the tunnel interface of RouterA.

```
[RouterA] ip route-static 10.1.2.0 255.255.255.0 tunnel 0/0/0
```

Configure a static route on the tunnel interface of RouterB.

```
[RouterB] ip route-static 10.1.1.0 255.255.255.0 tunnel 0/0/0
```

Step 7 Verify the configuration.

After the configurations are complete, run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID   Peer                               VPN   Flag(s)   Phase   RemoteType   RemoteID
-----
  16        202.138.162.1:500                 RD|ST v2:2      IP      202.138.162.1
  14        202.138.162.1:500                 RD|ST v2:1      IP      202.138.162.1

Number of IKE SA : 2

-----
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY    A--ALONE     NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
```

```
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer spub
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&G(=Et02P/%\!*%#
 ike-proposal 5
#
ipsec profile profile1
 ike-peer spub
 proposal tran1
#
interface Tunnel0/0/0
 ip address 192.168.1.1 255.255.255.0
 tunnel-protocol ipsec
 source 202.138.163.1
 destination 202.138.162.1
 ipsec profile profile1
#
interface GigabitEthernet1/0/0
 ip address 202.138.163.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
ip route-static 10.1.2.0 255.255.255.0 tunnel0/0/0
#
return
```

● Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer spua
 pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%^%#
 ike-proposal 5
#
ipsec profile profile1
 ike-peer spua
 proposal tran1
#
interface Tunnel0/0/0
 ip address 192.168.1.2 255.255.255.0
 tunnel-protocol ipsec
 source 202.138.162.1
 destination 202.138.163.1
 ipsec profile profile1
#
interface GigabitEthernet1/0/0
```

```
ip address 202.138.162.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
ip route-static 10.1.1.0 255.255.255.0 tunnel10/0/0
#
return
```

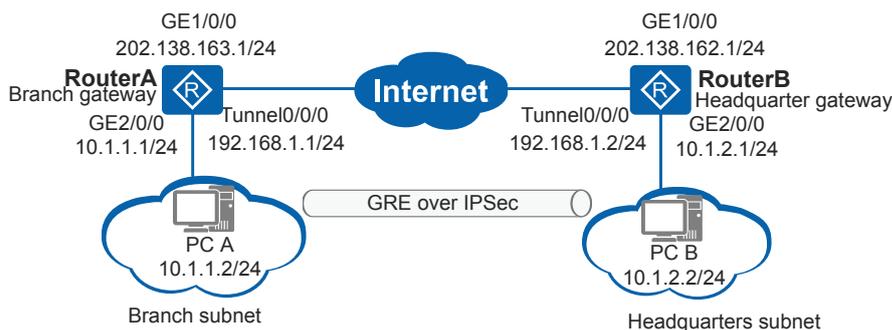
6.12.13 Example for Establishing GRE over IPsec Using a Tunnel Interface

Networking Requirements

As shown in [Figure 6-54](#), RouterA (branch gateway) and RouterB (headquarters gateway) communicate through the Internet.

The enterprise wants to protect traffic including multicast data between the headquarters and branch. As IPsec cannot be applied to multicast data directly, GRE over IPsec can be established between virtual tunnel interfaces to protect traffic on tunnel interfaces.

Figure 6-54 Establishing GRE over IPsec using a tunnel interface



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for physical interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure a GRE tunnel interface.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec profiles and reference IPsec proposals and IKE peers in the IPsec profiles.
6. Apply IPsec profiles to IPsec tunnel interfaces.
7. Configure static routes on IPsec tunnel interfaces and import data flows to be protected by IPsec to the tunnel interfaces.

Procedure

Step 1 Configure IP addresses and static routes for physical interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 202.138.163.2.

```
[RouterA] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 202.138.162.2.

```
[RouterB] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
```

Step 2 Configure a GRE tunnel interface.

Configure RouterA.

```
[RouterA] interface tunnel 0/0/0
[RouterA-Tunnel0/0/0] ip address 192.168.1.1 255.255.255.0
[RouterA-Tunnel0/0/0] tunnel-protocol gre
[RouterA-Tunnel0/0/0] source 202.138.163.1
[RouterA-Tunnel0/0/0] destination 202.138.162.1
[RouterA-Tunnel0/0/0] quit
```

Configure RouterB.

```
[RouterB] interface tunnel 0/0/0
[RouterB-Tunnel0/0/0] ip address 192.168.1.2 255.255.255.0
[RouterB-Tunnel0/0/0] tunnel-protocol gre
[RouterB-Tunnel0/0/0] source 202.138.162.1
[RouterB-Tunnel0/0/0] destination 202.138.163.1
[RouterB-Tunnel0/0/0] quit
```

Step 3 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
```

```
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128  
[RouterB-ipsec-proposal-tran1] quit
```

Step 4 Configure IKE peers on RouterA and RouterB.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5  
[RouterA-ike-proposal-5] authentication-algorithm sha2-256  
[RouterA-ike-proposal-5] encryption-algorithm aes-128  
[RouterA-ike-proposal-5] dh group14  
[RouterA-ike-proposal-5] quit
```

Configure an IKE peer on RouterA.

```
[RouterA] ike peer spub  
[RouterA-ike-peer-spub] ike-proposal 5  
[RouterA-ike-peer-spub] pre-shared-key cipher Huawei@1234  
[RouterA-ike-peer-spub] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ike proposal 5  
[RouterB-ike-proposal-5] authentication-algorithm sha2-256  
[RouterB-ike-proposal-5] encryption-algorithm aes-128  
[RouterB-ike-proposal-5] dh group14  
[RouterB-ike-proposal-5] quit
```

Configure an IKE peer on RouterB.

```
[RouterB] ike peer spua  
[RouterB-ike-peer-spua] ike-proposal 5  
[RouterB-ike-peer-spua] pre-shared-key cipher Huawei@1234  
[RouterB-ike-peer-spua] quit
```

Step 5 Create IPsec profiles on RouterA and RouterB.

Create an IPsec profile on RouterA.

```
[RouterA] ipsec profile profile1  
[RouterA-ipsec-profile-profile1] proposal tran1  
[RouterA-ipsec-profile-profile1] ike-peer spub  
[RouterA-ipsec-profile-profile1] quit
```

Create an IPsec profile on RouterB.

```
[RouterB] ipsec profile profile1  
[RouterB-ipsec-profile-profile1] proposal tran1  
[RouterB-ipsec-profile-profile1] ike-peer spua  
[RouterB-ipsec-profile-profile1] quit
```

Step 6 Apply the IPsec profiles to IPsec tunnel interfaces on RouterA and RouterB.

Apply the IPsec profile to the interface of RouterA.

```
[RouterA] interface tunnel 0/0/0  
[RouterA-Tunnel0/0/0] ipsec profile profile1  
[RouterA-Tunnel0/0/0] quit
```

Apply the IPsec policy to the interface of RouterB.

```
[RouterB] interface tunnel 0/0/0  
[RouterB-Tunnel0/0/0] ipsec profile profile1  
[RouterB-Tunnel0/0/0] quit
```

Run the **display ipsec profile** command on RouterA and RouterB to view the IPsec profile configuration.

Step 7 Configure static routes on IPsec tunnel interfaces and import data flows to be protected by IPsec to the tunnel interfaces.

Configure a static route on the tunnel interface of RouterA.

```
[RouterA] ip route-static 10.1.2.0 255.255.255.0 tunnel 0/0/0
```

Configure a static route on the tunnel interface of RouterB.

```
[RouterB] ip route-static 10.1.1.0 255.255.255.0 tunnel 0/0/0
```

Step 8 Verify the configuration.

After the configurations are complete, run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
16       202.138.162.1:500          RD|ST  v2:2     IP     202.138.162.1
14       202.138.162.1:500          RD|ST  v2:1     IP     202.138.162.1

Number of IKE SA : 2
-----
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer spub
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&G(=Et02P/%\*!%^%#
 ike-proposal 5
#
ipsec profile profile1
 ike-peer spub
 proposal tran1
#
interface Tunnel0/0/0
 ip address 192.168.1.1 255.255.255.0
 tunnel-protocol gre
 source 202.138.163.1
 destination 202.138.162.1
 ipsec profile profile1
#
interface GigabitEthernet1/0/0
 ip address 202.138.163.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
```

```
ip route-static 10.1.2.0 255.255.255.0 tunnel10/0/0
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer spua
 pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%^%#
 ike-proposal 5
#
ipsec profile profile1
 ike-peer spua
 proposal tran1
#
interface Tunnel0/0/0
 ip address 192.168.1.2 255.255.255.0
 tunnel-protocol gre
 source 202.138.162.1
 destination 202.138.163.1
 ipsec profile profile1
#
interface GigabitEthernet1/0/0
 ip address 202.138.162.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
ip route-static 10.1.1.0 255.255.255.0 tunnel10/0/0
#
return
```

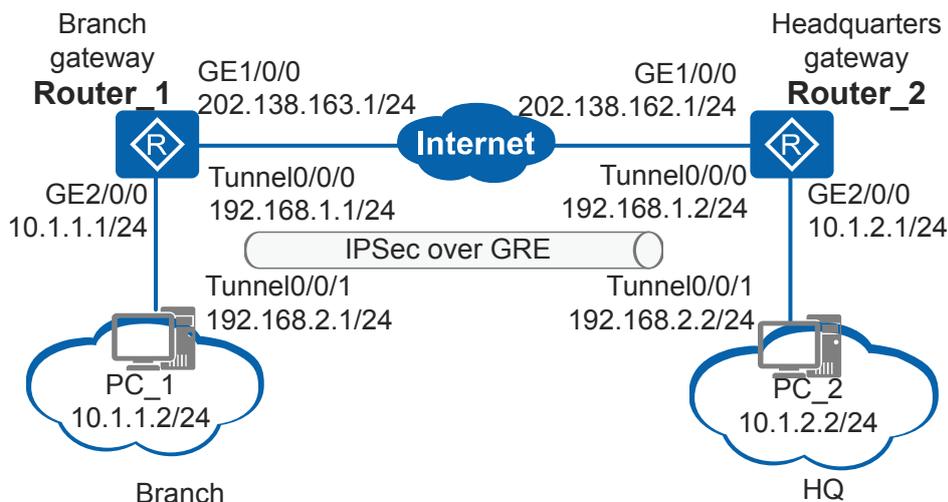
6.12.14 Example for Establishing IPsec over GRE Using a Tunnel Interface

Networking Requirements

As shown in [Figure 6-55](#), Router_1 (branch gateway) and Router_2 (headquarters gateway) communicate through the Internet.

The branch communicates with the headquarters through a GRE tunnel. The enterprise wants to protect traffic excluding multicast data between the headquarters and branch. IPsec over GRE can be established between virtual tunnel interfaces to protect traffic between the headquarters and branch.

Figure 6-55 Establishing IPsec over GRE using a tunnel interface



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for physical interfaces on Router_1 and Router_2 so that routes between Router_1 and Router_2 are reachable.
2. Configure a GRE tunnel interface.
3. Configure IPsec proposals to define the method used to protect IPsec traffic.
4. Configure IKE peers to define IKE negotiation attributes.
5. Configure IPsec profiles and reference IPsec policies and IKE peers in the IPsec profiles.
6. Configure IPsec tunnel interfaces and specify a GRE tunnel interface as the source interface of the IPsec tunnel and the other GRE tunnel interface as the outbound interface for routes to the destination address of the IPsec tunnel.
7. Apply IPsec profiles to the IPsec tunnel interfaces to enable IPsec on the interfaces.
8. Configure static routes for the IPsec tunnel interfaces to import data flows to be protected by IPsec to the interfaces.

Procedure

Step 1 Configure IP addresses and static routes for physical interfaces on Router_1 and Router_2.

Assign an IP address to an interface on Router_1.

```
<Huawei> system-view
[Huawei] sysname Router_1
[Router_1] interface gigabitethernet 1/0/0
[Router_1-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[Router_1-GigabitEthernet1/0/0] quit
[Router_1] interface gigabitethernet 2/0/0
[Router_1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[Router_1-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on Router_1. This example assumes that the next hop address in the route to Router_2 is 202.138.163.2.

```
[Router_1] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
```

Assign an IP address to an interface on Router_2.

```
<Huawei> system-view
[Huawei] sysname Router_2
[Router_2] interface gigabitethernet 1/0/0
[Router_2-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[Router_2-GigabitEthernet1/0/0] quit
[Router_2] interface gigabitethernet 2/0/0
[Router_2-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[Router_2-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on Router_2. This example assumes that the next hop address in the route to Router_1 is 202.138.162.2.

```
[Router_2] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
```

Step 2 Configure a GRE tunnel interface.

Configure Router_1.

```
[Router_1] interface tunnel 0/0/0
[Router_1-Tunnel0/0/0] ip address 192.168.1.1 255.255.255.0
[Router_1-Tunnel0/0/0] tunnel-protocol gre
[Router_1-Tunnel0/0/0] source 202.138.163.1
[Router_1-Tunnel0/0/0] destination 202.138.162.1
[Router_1-Tunnel0/0/0] quit
```

Configure Router_2.

```
[Router_2] interface tunnel 0/0/0
[Router_2-Tunnel0/0/0] ip address 192.168.1.2 255.255.255.0
[Router_2-Tunnel0/0/0] tunnel-protocol gre
[Router_2-Tunnel0/0/0] source 202.138.162.1
[Router_2-Tunnel0/0/0] destination 202.138.163.1
[Router_2-Tunnel0/0/0] quit
```

Step 3 Create IPsec proposals on Router_1 and Router_2.

Create an IPsec proposal on Router_1.

```
[Router_1] ipsec proposal tran1
[Router_1-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[Router_1-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[Router_1-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on Router_2.

```
[Router_2] ipsec proposal tran1
[Router_2-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[Router_2-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[Router_2-ipsec-proposal-tran1] quit
```

Step 4 Configure IKE peers on Router_1 and Router_2.

Create an IKE proposal on Router_1.

```
[Router_1] ike proposal 5
[Router_1-ike-proposal-5] authentication-algorithm sha2-256
[Router_1-ike-proposal-5] encryption-algorithm aes-128
[Router_1-ike-proposal-5] dh group14
[Router_1-ike-proposal-5] quit
```

Configure an IKE peer on Router_1.

```
[Router_1] ike peer spub
[Router_1-ike-peer-spub] ike-proposal 5
```

```
[Router_1-ike-peer-spua] pre-shared-key cipher Huawei@1234  
[Router_1-ike-peer-spua] quit
```

Create an IPsec proposal on Router_2.

```
[Router_2] ike proposal 5  
[Router_2-ike-proposal-5] authentication-algorithm sha2-256  
[Router_2-ike-proposal-5] encryption-algorithm aes-128  
[Router_2-ike-proposal-5] dh group14  
[Router_2-ike-proposal-5] quit
```

Configure an IKE peer on Router_2.

```
[Router_2] ike peer spua  
[Router_2-ike-peer-spua] ike-proposal 5  
[Router_2-ike-peer-spua] pre-shared-key cipher Huawei@1234  
[Router_2-ike-peer-spua] quit
```

Step 5 Create IPsec profiles on Router_1 and Router_2.

Create an IPsec profile on Router_1.

```
[Router_1] ipsec profile profile1  
[Router_1-ipsec-profile-profile1] proposal tran1  
[Router_1-ipsec-profile-profile1] ike-peer spua  
[Router_1-ipsec-profile-profile1] quit
```

Create an IPsec profile on Router_2.

```
[Router_2] ipsec profile profile1  
[Router_2-ipsec-profile-profile1] proposal tran1  
[Router_2-ipsec-profile-profile1] ike-peer spua  
[Router_2-ipsec-profile-profile1] quit
```

Step 6 Configure an IPsec tunnel interface on Router_1 and Router_2 respectively. Specify a GRE tunnel interface as the source interface of the IPsec tunnel and the other GRE tunnel interface as the outbound interface for routes to the destination address of the IPsec tunnel.

Configure Router_1.

```
[Router_1] interface tunnel 0/0/1  
[Router_1-Tunnel0/0/1] ip address 192.168.2.1 255.255.255.0  
[Router_1-Tunnel0/0/1] tunnel-protocol ipsec  
[Router_1-Tunnel0/0/1] source tunnel 0/0/0  
[Router_1-Tunnel0/0/1] destination 192.168.1.2  
[Router_1-Tunnel0/0/1] quit
```

Configure Router_2.

```
[Router_2] interface tunnel 0/0/1  
[Router_2-Tunnel0/0/1] ip address 192.168.2.2 255.255.255.0  
[Router_2-Tunnel0/0/1] tunnel-protocol ipsec  
[Router_2-Tunnel0/0/1] source tunnel 0/0/0  
[Router_2-Tunnel0/0/1] destination 192.168.1.1  
[Router_2-Tunnel0/0/1] quit
```

Step 7 Apply IPsec profiles to the IPsec tunnel interfaces.

Apply the IPsec profile to the interface of Router_1.

```
[Router_1] interface tunnel 0/0/1  
[Router_1-Tunnel0/0/1] ipsec profile profile1  
[Router_1-Tunnel0/0/1] quit
```

Apply the IPsec policy to the interface of Router_2.

```
[Router_2] interface tunnel 0/0/1  
[Router_2-Tunnel0/0/1] ipsec profile profile1  
[Router_2-Tunnel0/0/1] quit
```

Run the **display ipsec profile** command on Router_1 and Router_2 to view the IPsec profile configuration.

Step 8 Configure static routes on IPsec tunnel interfaces and import data flows to be protected by IPsec to the tunnel interfaces.

Configure a static route on the tunnel interface of Router_1.

```
[Router_1] ip route-static 10.1.2.0 255.255.255.0 tunnel 0/0/1
```

Configure a static route on the tunnel interface of Router_2.

```
[Router_2] ip route-static 10.1.1.0 255.255.255.0 tunnel 0/0/1
```

Step 9 Verify the configuration.

After the configurations are complete, run the **display ike sa** command on Router_1 and Router_2 to view the IKE SA configuration. The display on Router_1 is used as an example.

```
[Router_1] display ike sa
IKE SA information :
 Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
 16      202.138.162.1:500          RD|ST  v2:2     IP     202.138.162.1
 14      202.138.162.1:500          RD|ST  v2:1     IP     202.138.162.1

Number of IKE SA : 2

-----
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of Router_1

```
#
 sysname Router_1
#
 ipsec proposal tran1
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
 ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
 ike peer spub
  pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
  ike-proposal 5
#
 ipsec profile profile1
  ike-peer spub
  proposal tran1
#
 interface Tunnel0/0/0
  ip address 192.168.1.1 255.255.255.0
  tunnel-protocol gre
  source 202.138.163.1
  destination 202.138.162.1
#
 interface Tunnel0/0/1
```

```
ip address 192.168.2.1 255.255.255.0
tunnel-protocol ipsec
source Tunnel0/0/0
destination 192.168.1.2
ipsec profile profile1
#
interface GigabitEthernet1/0/0
ip address 202.138.163.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 10.1.2.0 255.255.255.0 tunnel0/0/1
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
#
return
```

● Configuration file of Router_2

```
#
sysname Router_2
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer spua
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%%#
ike-proposal 5
#
ipsec profile profile1
ike-peer spua
proposal tran1
#
interface Tunnel0/0/0
ip address 192.168.1.2 255.255.255.0
tunnel-protocol gre
source 202.138.163.2
destination 202.138.163.1
#
interface Tunnel0/0/1
ip address 192.168.2.2 255.255.255.0
tunnel-protocol ipsec
source Tunnel0/0/0
destination 192.168.1.1
ipsec profile profile1
#
interface GigabitEthernet1/0/0
ip address 202.138.162.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
ip route-static 10.1.1.0 255.255.255.0 tunnel0/0/1
ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
#
return
```

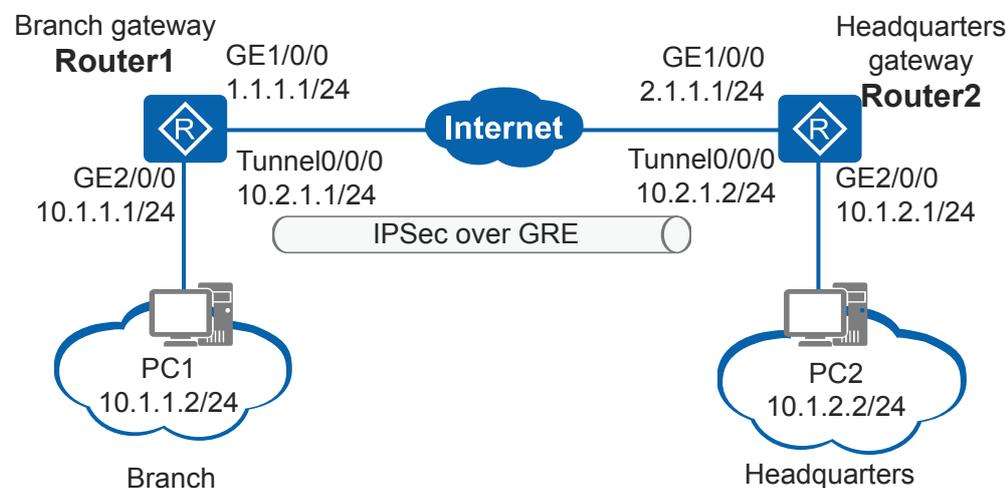
6.12.15 Example for Establishing an IPsec over GRE Tunnel Between the Headquarters and Branch (Based on ACL)

Networking Requirements

In **Figure 6-56**, Router1 is the gateway of an enterprise branch, and Router2 is the gateway of the headquarters. Router1 and Router2 communicate through the public network.

On the live network, the enterprise branch communicates with the headquarters through a GRE tunnel. The enterprise wants to protect traffic excluding multicast data between the headquarters and branch. An IPsec over GRE tunnel can be established based on ACL to protect traffic between the headquarters and branch.

Figure 6-56 Establishing an IPsec over GRE tunnel between the headquarters and branch



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the IP address and static route on each physical interface to ensure reachable routes between the interfaces.
2. Configure GRE tunnel interfaces.
3. Configure static routes for tunnel interfaces so that data flows are imported to the tunnel interfaces.
4. Configure an ACL to define the data flows to be protected by IPsec.
5. Configure an IPsec proposal to define the traffic protection method.
6. Configure an IKE peer and IKE proposal to define the attributes used for IKE negotiation.
7. Configure a security policy and apply the IKE proposal, IKE peer, and ACL.
8. Apply the security policy to the tunnel interfaces to enable IPsec protection.

Procedure

Step 1 Configure an IP address and a static route for each physical interface on the routers.

Configure Router1.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface gigabitethernet 1/0/0
[Router1-GigabitEthernet1/0/0] ip address 1.1.1.1 255.255.255.0
[Router1-GigabitEthernet1/0/0] quit
[Router1] interface gigabitethernet 2/0/0
[Router1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[Router1-GigabitEthernet2/0/0] quit
```

Configure a static route to the remote end on Router1. This example assumes that the next hop address of the route is 1.1.1.2.

```
[Router1] ip route-static 2.1.1.0 255.255.255.0 1.1.1.2
```

Configure Router2.

```
<Huawei> system-view
[Huawei] sysname Router2
[Router2] interface gigabitethernet 1/0/0
[Router2-GigabitEthernet1/0/0] ip address 2.1.1.1 255.255.255.0
[Router2-GigabitEthernet1/0/0] quit
[Router2] interface gigabitethernet 2/0/0
[Router2-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[Router2-GigabitEthernet2/0/0] quit
```

Configure a static route to the remote end on Router2. This example assumes that the next hop address of the route is 2.1.1.2.

```
[Router2] ip route-static 1.1.1.0 255.255.255.0 2.1.1.2
```

Step 2 Configure GRE tunnel interfaces.

Configure Router1.

```
[Router1] interface tunnel 0/0/0
[Router1-Tunnel0/0/0] ip address 10.2.1.1 255.255.255.0
[Router1-Tunnel0/0/0] tunnel-protocol gre
[Router1-Tunnel0/0/0] source 1.1.1.1
[Router1-Tunnel0/0/0] destination 2.1.1.1
[Router1-Tunnel0/0/0] quit
```

Configure Router2.

```
[Router2] interface tunnel 0/0/0
[Router2-Tunnel0/0/0] ip address 10.2.1.2 255.255.255.0
[Router2-Tunnel0/0/0] tunnel-protocol gre
[Router2-Tunnel0/0/0] source 2.1.1.1
[Router2-Tunnel0/0/0] destination 1.1.1.1
[Router2-Tunnel0/0/0] quit
```

Step 3 Configure static routes for tunnel interfaces so that data flows are imported to the tunnel interfaces.

Configure Router1.

```
[Router1] ip route-static 10.1.2.0 255.255.255.0 tunnel 0/0/0
```

Configure Router2.

```
[Router2] ip route-static 10.1.1.0 255.255.255.0 tunnel 0/0/0
```

After the configuration is complete, run the **display tunnel-info all** command on the routers to view GRE tunnel establishment information. The command output on Router1 is used as an example.

```
[Router1] display tunnel-info all
* -> Allocated VC Token
Tunnel ID          Type          Destination          Token
-----
0x1                gre          2.1.1.1             1
```

Step 4 Create an ACL on the routers to define the data flows to be protected.

Configure Router1.

```
[Router1] acl number 3101
[Router1-acl-adv-3101] rule permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[Router1-acl-adv-3101] quit
```

Configure Router2.

```
[Router2] acl number 3101
[Router2-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Router2-acl-adv-3101] quit
```

Step 5 Create an IPsec proposal on the routers.

Configure Router1.

```
[Router1] ipsec proposal tran1
[Router1-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[Router1-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[Router1-ipsec-proposal-tran1] quit
```

Configure Router2.

```
[Router2] ipsec proposal tran1
[Router2-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[Router2-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[Router2-ipsec-proposal-tran1] quit
```

Step 6 Configure an IKE proposal and an IKE peer on the routers.

Configure Router1.

```
[Router1] ike proposal 5
[Router1-ike-proposal-5] authentication-algorithm sha2-256
[Router1-ike-proposal-5] encryption-algorithm aes-128
[Router1-ike-proposal-5] dh group14
[Router1-ike-proposal-5] quit
[Router1] ike peer spub
[Router1-ike-peer-spub] ike-proposal 5
[Router1-ike-peer-spub] pre-shared-key cipher Huawei@1234
[Router1-ike-peer-spub] remote-address 10.2.1.2
[Router1-ike-peer-spub] quit
```

Configure Router2.

```
[Router2] ike proposal 5
[Router2-ike-proposal-5] authentication-algorithm sha2-256
[Router2-ike-proposal-5] encryption-algorithm aes-128
[Router2-ike-proposal-5] dh group14
[Router2-ike-proposal-5] quit
[Router2] ike peer spua
[Router2-ike-peer-spua] ike-proposal 5
[Router2-ike-peer-spua] pre-shared-key cipher Huawei@1234
[Router2-ike-peer-spua] remote-address 10.2.1.1
[Router2-ike-peer-spua] quit
```

Step 7 Create a security policy on the routers.

Configure Router1.

```
[Router1] ipsec policy map1 10 isakmp
[Router1-ipsec-policy-isakmp-map1-10] proposal tran1
[Router1-ipsec-policy-isakmp-map1-10] ike-peer spub
[Router1-ipsec-policy-isakmp-map1-10] security acl 3101
[Router1-ipsec-policy-isakmp-map1-10] quit
```

Configure Router2.

```
[Router2] ipsec policy use1 10 isakmp
[Router2-ipsec-policy-isakmp-use1-10] proposal tran1
[Router2-ipsec-policy-isakmp-use1-10] ike-peer spua
[Router2-ipsec-policy-isakmp-use1-10] security acl 3101
[Router2-ipsec-policy-isakmp-use1-10] quit
```

Step 8 Apply the security policy to the router interfaces.

Configure Router1.

```
[Router1] interface tunnel 0/0/0
[Router1-Tunnel0/0/0] ipsec policy map1
[Router1-Tunnel0/0/0] quit
```

Configure Router2.

```
[Router2] interface tunnel 0/0/0
[Router2-Tunnel0/0/0] ipsec policy use1
[Router2-Tunnel0/0/0] quit
```

Step 9 Verify the configuration.

After the configuration is complete, run the **display ike sa** command on the routers to view the SA establishment information. The command output on Router1 is used as an example.

```
[Router1] display ike sa
IKE SA information :
Conn-ID   Peer                VPN   Flag(s)   Phase   RemoteType  RemoteID
-----
20        10.2.1.2:500        RD|A  v2:2     IP      10.2.1.2
19        10.2.1.2:500        RD|A  v2:1     IP      10.2.1.2

Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY    A--ALONE     NEG--NEGOTIATING
```

After an SA is successfully established, data transmitted between the headquarters and branch is encrypted.

---End

Configuration Files

- Router1 configuration file

```
#
sysname Router1
#
acl number 3101
rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec proposal tran1
```

```
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
#
ike peer spub
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&G(=Et02P/%\*!%^%#
 ike-proposal 5
 remote-address 10.2.1.2
#
ipsec policy map1 10 isakmp
 security acl 3101
 ike-peer spub
 proposal tran1
#
interface GigabitEthernet1/0/0
 ip address 1.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 10.2.1.1 255.255.255.0
 tunnel-protocol gre
 source 1.1.1.1
 destination 2.1.1.1
 ipsec policy map1
#
ip route-static 2.1.1.0 255.255.255.0 1.1.1.2
ip route-static 10.1.2.0 255.255.255.0 Tunnel0/0/0
#
return
```

● Router2 configuration file

```
#
 sysname Router2
#
acl number 3101
 rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
#
ike peer spua
 pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%^%#
 ike-proposal 5
 remote-address 10.2.1.1
#
ipsec policy use1 10 isakmp
 security acl 3101
 ike-peer spua
 proposal tran1
#
interface GigabitEthernet1/0/0
 ip address 2.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 10.2.1.2 255.255.255.0
```

```
tunnel-protocol gre
source 2.1.1.1
destination 1.1.1.1
ipsec policy use1
#
ip route-static 1.1.1.0 255.255.255.0 2.1.1.2
ip route-static 10.1.1.0 255.255.255.0 Tunnel10/0/0
#
return
```

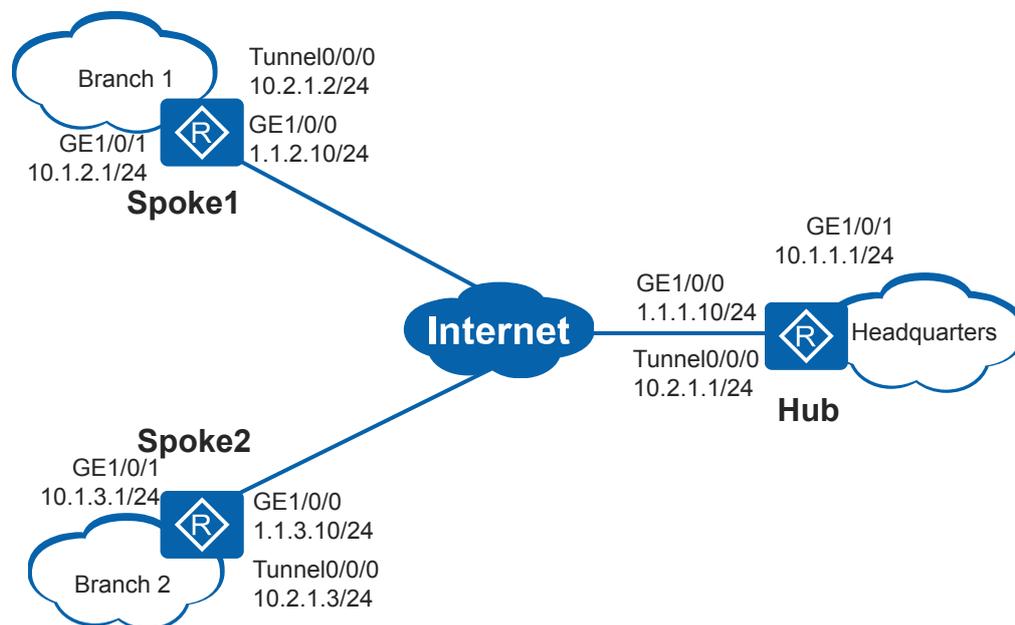
6.12.16 Example for Establishing IPsec over DSVPN Tunnels Between Hub and Spokes (Based on ACL)

Networking Requirements

In **Figure 6-57**, a large-sized enterprise has the headquarters (Hub) and multiple branches (Spoke1 and Spoke2 in this example) located in different areas, and the Spokes connect to public networks using dynamic IP addresses obtained through DHCP. DSVPN is deployed to enable communication between Spokes as well as between Spoke and Hub.

The enterprise requires that data transmitted between Spokes as well as between Spoke and Hub be encrypted. IPsec over DSVPN can be configured on Hub and Spokes to provide traffic protection.

Figure 6-57 Establishing IPsec over DSVPN tunnels between Hub and Spokes



NOTE

Assume that the dynamic addresses obtained by Spoke1 and Spoke2 are 1.1.2.10 and 1.1.3.10, respectively.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure DSVPN to implement VPN interconnection between the Spokes because the Spokes connect to the public network using dynamic IP addresses and the Spokes do not know the public IP addresses of each other.
2. Deploy DSVPN in shortcut mode because there are a large number of Spokes.
3. Configure OSPF to simplify maintenance because subnets of the Hub and Spokes frequently change.
4. Configure IPsec over DSVPN to encrypt data transmitted between the Hub and Spokes using IPsec before transmitting the data using DSVPN.

Procedure

Step 1 Configure IP addresses for interfaces.

Configure the Hub.

```
<Huawei> system-view
[Huawei] sysname Hub
[Hub] interface gigabitethernet 1/0/0
[Hub-GigabitEthernet1/0/0] ip address 1.1.1.10 255.255.255.0
[Hub-GigabitEthernet1/0/0] quit
[Hub] interface gigabitethernet 1/0/1
[Hub-GigabitEthernet1/0/1] ip address 10.1.1.1 255.255.255.0
[Hub-GigabitEthernet1/0/1] quit
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ip address 10.2.1.1 255.255.255.0
[Hub-Tunnel0/0/0] quit
```

Configure Spoke1.

```
<Huawei> system-view
[Huawei] sysname Spoke1
[Spoke1] interface gigabitethernet 1/0/0
[Spoke1-GigabitEthernet1/0/0] ip address dhcp-alloc
[Spoke1-GigabitEthernet1/0/0] quit
[Spoke1] interface gigabitethernet 1/0/1
[Spoke1-GigabitEthernet1/0/1] ip address 10.1.2.1 255.255.255.0
[Spoke1-GigabitEthernet1/0/1] quit
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] ip address 10.2.1.2 255.255.255.0
[Spoke1-Tunnel0/0/0] quit
```

Configure Spoke2.

```
<Huawei> system-view
[Huawei] sysname Spoke2
[Spoke2] interface gigabitethernet 1/0/0
[Spoke2-GigabitEthernet1/0/0] ip address dhcp-alloc
[Spoke2-GigabitEthernet1/0/0] quit
[Spoke2] interface gigabitethernet 1/0/1
[Spoke2-GigabitEthernet1/0/1] ip address 10.1.3.1 255.255.255.0
[Spoke2-GigabitEthernet1/0/1] quit
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] ip address 10.2.1.3 255.255.255.0
[Spoke2-Tunnel0/0/0] quit
```

Step 2 Configure OSPF to ensure reachable routes over the public network.

Configure the Hub.

```
[Hub] ospf 2
[Hub-ospf-2] area 0.0.0.1
[Hub-ospf-2-area-0.0.0.1] network 1.1.1.0 0.0.0.255
[Hub-ospf-2-area-0.0.0.1] quit
[Hub-ospf-2] quit
```

Configure Spoke1.

```
[Spoke1] ospf 2
[Spoke1-ospf-2] area 0.0.0.1
[Spoke1-ospf-2-area-0.0.0.1] network 1.1.2.0 0.0.0.255
[Spoke1-ospf-2-area-0.0.0.1] quit
[Spoke1-ospf-2] quit
```

Configure Spoke2.

```
[Spoke2] ospf 2
[Spoke2-ospf-2] area 0.0.0.1
[Spoke2-ospf-2-area-0.0.0.1] network 1.1.3.0 0.0.0.255
[Spoke2-ospf-2-area-0.0.0.1] quit
[Spoke2-ospf-2] quit
```

Step 3 Configure OSPF to ensure reachable routes between private networks.

Configure the Hub.

```
[Hub] ospf 1 router-id 10.2.1.1
[Hub-ospf-1] area 0.0.0.0
[Hub-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Hub-ospf-1-area-0.0.0.0] quit
[Hub-ospf-1] quit
```

Configure Spoke1.

```
[Spoke1] ospf 1 router-id 10.2.1.2
[Spoke1-ospf-1] area 0.0.0.0
[Spoke1-ospf-1-area-0.0.0.0] network 10.1.2.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Spoke1-ospf-1-area-0.0.0.0] quit
[Spoke1-ospf-1] quit
```

Configure Spoke2.

```
[Spoke2] ospf 1 router-id 10.2.1.3
[Spoke2-ospf-1] area 0.0.0.0
[Spoke2-ospf-1-area-0.0.0.0] network 10.1.3.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[Spoke2-ospf-1-area-0.0.0.0] quit
[Spoke2-ospf-1] quit
```

Step 4 Configure tunnel interfaces.

On the Hub and Spokes, set the OSPF network type to broadcast to enable the Spokes to learn routes from each other. On Spoke1 and Spoke2, configure static NHRP peer entries of the Hub.

Configure the Hub.

```
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] tunnel-protocol gre p2mp
[Hub-Tunnel0/0/0] source gigabitethernet 1/0/0
[Hub-Tunnel0/0/0] nhrp entry multicast dynamic
[Hub-Tunnel0/0/0] ospf network-type p2mp
[Hub-Tunnel0/0/0] quit
```

Configure Spoke1.

```
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke1-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke1-Tunnel0/0/0] nhrp entry 10.2.1.1 1.1.1.10 register
[Spoke1-Tunnel0/0/0] ospf network-type p2mp
[Spoke1-Tunnel0/0/0] quit
```

Configure Spoke2.

```
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] tunnel-protocol gre p2mp
[Spoke2-Tunnel0/0/0] source gigabitethernet 1/0/0
[Spoke2-Tunnel0/0/0] nhrp entry 10.2.1.1 1.1.1.10 register
[Spoke2-Tunnel0/0/0] ospf network-type p2mp
[Spoke2-Tunnel0/0/0] quit
```

Step 5 Configure an ACL to define the data flows to be protected by IPsec.

Configure Spoke1.

```
[Spoke1] acl number 3101
[Spoke1-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Spoke1-acl-adv-3101] rule permit ip source 10.1.2.0 0.0.0.255 destination
10.1.3.0 0.0.0.255
[Spoke1-acl-adv-3101] quit
```

Configure Spoke2.

```
[Spoke2] acl number 3101
[Spoke2-acl-adv-3101] rule permit ip source 10.1.3.0 0.0.0.255 destination
10.1.1.0 0.0.0.255
[Spoke2-acl-adv-3101] rule permit ip source 10.1.3.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[Spoke2-acl-adv-3101] quit
```

Step 6 Configure an IKE proposal.

Configure the Hub.

```
[Hub] ike proposal 1
[Hub-ike-proposal-1] dh group14
[Hub-ike-proposal-1] encryption-algorithm aes-256
[Hub-ike-proposal-1] authentication-algorithm sha2-256
[Hub-ike-proposal-1] prf aes-xcbc-128
[Hub-ike-proposal-1] quit
```

Configure Spoke1.

```
[Spoke1] ike proposal 1
[Spoke1-ike-proposal-1] dh group14
[Spoke1-ike-proposal-1] encryption-algorithm aes-256
[Spoke1-ike-proposal-1] authentication-algorithm sha2-256
[Spoke1-ike-proposal-1] prf aes-xcbc-128
[Spoke1-ike-proposal-1] quit
```

Configure Spoke2.

```
[Spoke2] ike proposal 1
[Spoke2-ike-proposal-1] dh group14
[Spoke2-ike-proposal-1] encryption-algorithm aes-256
[Spoke2-ike-proposal-1] authentication-algorithm sha2-256
[Spoke2-ike-proposal-1] prf aes-xcbc-128
[Spoke2-ike-proposal-1] quit
```

Step 7 Configure an IKE peer.

Configure the Hub.

```
[Hub] ike peer hub
[Hub-ike-peer-hub] ike-proposal 1
[Hub-ike-peer-hub] pre-shared-key cipher Huawei@1234
[Hub-ike-peer-hub] dpd type periodic
[Hub-ike-peer-hub] dpd idle-time 40
[Hub-ike-peer-hub] quit
```

Configure Spoke1.

```
[Spoke1] ike peer spokel
[Spoke1-ike-peer-spokel] ike-proposal 1
```

```
[Spoke1-ike-peer-spoke1] pre-shared-key cipher Huawei@1234
[Spoke1-ike-peer-spoke1] remote-address 10.2.1.1
[Spoke1-ike-peer-spoke1] dpd type periodic
[Spoke1-ike-peer-spoke1] dpd idle-time 40
[Spoke1-ike-peer-spoke1] quit
```

Configure Spoke2.

```
[Spoke2] ike peer spoke2
[Spoke2-ike-peer-spoke2] ike-proposal 1
[Spoke2-ike-peer-spoke2] pre-shared-key cipher Huawei@1234
[Spoke2-ike-peer-spoke2] remote-address 10.2.1.1
[Spoke2-ike-peer-spoke2] dpd type periodic
[Spoke2-ike-peer-spoke2] dpd idle-time 40
[Spoke2-ike-peer-spoke2] quit
```

Step 8 Create an IPsec proposal.

On the Hub and Spokes, create an IPsec proposal.

Configure the Hub.

```
[Hub] ipsec proposal pro1
[Hub-ipsec-proposal-pro1] transform esp
[Hub-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Hub-ipsec-proposal-pro1] esp encryption-algorithm aes-256
[Hub-ipsec-proposal-pro1] quit
```

Configure Spoke1.

```
[Spoke1] ipsec proposal pro1
[Spoke1-ipsec-proposal-pro1] transform esp
[Spoke1-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke1-ipsec-proposal-pro1] esp encryption-algorithm aes-256
[Spoke1-ipsec-proposal-pro1] quit
```

Configure Spoke2.

```
[Spoke2] ipsec proposal pro1
[Spoke2-ipsec-proposal-pro1] transform esp
[Spoke2-ipsec-proposal-pro1] esp authentication-algorithm sha2-256
[Spoke2-ipsec-proposal-pro1] esp encryption-algorithm aes-256
[Spoke2-ipsec-proposal-pro1] quit
```

Step 9 Configure a security policy.

Configure the Hub.

```
[Hub] ipsec policy-template use1 10
[Hub-ipsec-policy-templet-use1-10] ike-peer hub
[Hub-ipsec-policy-templet-use1-10] proposal pro1
[Hub-ipsec-policy-templet-use1-10] quit
[Hub] ipsec policy policy1 10 isakmp template use1
```

Configure Spoke1.

```
[Spoke1] ipsec policy policy1 10 isakmp
[Spoke1-ipsec-policy-isakmp-policy1-10] ike-peer spoke1
[Spoke1-ipsec-policy-isakmp-policy1-10] proposal pro1
[Spoke1-ipsec-policy-isakmp-policy1-10] security acl 3101
[Spoke1-ipsec-policy-isakmp-policy1-10] quit
```

Configure Spoke2.

```
[Spoke2] ipsec policy policy1 10 isakmp
[Spoke2-ipsec-policy-isakmp-policy1-10] ike-peer spoke2
[Spoke2-ipsec-policy-isakmp-policy1-10] proposal pro1
[Spoke2-ipsec-policy-isakmp-policy1-10] security acl 3101
[Spoke2-ipsec-policy-isakmp-policy1-10] quit
```

Step 10 Apply the security policy to the tunnel interfaces to enable IPsec protection.

```
# Configure the Hub.
[Hub] interface tunnel 0/0/0
[Hub-Tunnel0/0/0] ipsec policy policy1
[Hub-Tunnel0/0/0] quit

# Configure Spoke1.
[Spoke1] interface tunnel 0/0/0
[Spoke1-Tunnel0/0/0] ipsec policy policy1
[Spoke1-Tunnel0/0/0] quit

# Configure Spoke2.
[Spoke2] interface tunnel 0/0/0
[Spoke2-Tunnel0/0/0] ipsec policy policy1
[Spoke2-Tunnel0/0/0] quit
```

Step 11 Verify the configuration.

After the configuration is complete, run the **display ike sa** command on the Spokes to view the SA establishment information. The command output on Spoke1 is used as an example.

```
[Spoke1] display ike sa
IKE SA information :
  Conn-ID   Peer                VPN   Flag(s)   Phase   RemoteType  RemoteID
-----
  20        10.2.1.1:500        RD|A  v2:2      IP      10.2.1.1
  19        10.2.1.1:500        RD|A  v2:1      IP      10.2.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING    TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY    A--ALONE     NEG--NEGOTIATING
```

The command output shows that an SA is successfully established between Spoke1 and the Hub to encrypt data transmitted between them.

After pinging 10.1.3.1 from Spoke1, run the **display ike sa** command to view the SA establishment information. The command output on Spoke1 is used as an example.

```
[Spoke1] display ike sa
IKE SA information :
  Conn-ID   Peer                VPN   Flag(s)   Phase   RemoteType  RemoteID
-----
  22        10.2.1.3:500        RD|A  v2:2      IP      10.2.1.3
  21        10.2.1.3:500        RD|A  v2:1      IP      10.2.1.3
  20        10.2.1.1:500        RD|A  v2:2      IP      10.2.1.1
  19        10.2.1.1:500        RD|A  v2:1      IP      10.2.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING    TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY    A--ALONE     NEG--NEGOTIATING
```

The command output shows that an SA is successfully established between Spoke1 and Spoke2 to encrypt data transmitted between them.

----End

Configuration Files

- Hub configuration file

```
#
sysname Hub
#
ipsec proposal pro1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-256
#
ike proposal 1
 encryption-algorithm
 aes-256
 dh
 group14
 authentication-algorithm sha2-256
 prf aes-xcbc-128
#
ike peer hub
 pre-shared-key cipher %%#O3uIP\YNF+`AcJhbZ&C7y*iVl0OU@DraF58J4=;%^%#
 ike-proposal 1
 dpd type periodic
 dpd idle-time 40
#
ipsec policy-template use1 10
 ike-peer hub
 proposal pro1
#
ipsec policy policy1 10 isakmp template use1
#
interface GigabitEthernet1/0/0
 ip address 1.1.1.10 255.255.255.0
#
interface GigabitEthernet1/0/1
 ip address 10.1.1.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 10.2.1.1 255.255.255.0
 tunnel-protocol gre p2mp
 source GigabitEthernet1/0/0
 ospf network-type p2mp
 ipsec policy policy1
 nhrp entry multicast dynamic
#
ospf 1 router-id 10.2.1.1
 area 0.0.0.0
 network 10.1.1.0 0.0.0.255
 network 10.2.1.0 0.0.0.255
#
ospf 2
 area 0.0.0.1
 network 1.1.1.0 0.0.0.255
#
return
```

- Spoke1 configuration file

```
#
sysname Spoke1
#
acl number 3101
 rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
 rule 10 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.3.0 0.0.0.255
#
ipsec proposal pro1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-256
#
ike proposal 1
 encryption-algorithm aes-256
```

```
dh group14
authentication-algorithm sha2-256
prf aes-xcbc-128
#
ike peer spoke1
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iV100U@DraF58J4=;%^%#
ike-proposal 1
dpd type periodic
dpd idle-time 40
remote-address 10.2.1.1
#
ipsec policy policy1 10 isakmp
security acl 3101
ike-peer spoke1
proposal pro1
#
interface GigabitEthernet1/0/0
ip address dhcp-alloc
#
interface GigabitEthernet1/0/1
ip address 10.1.2.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 10.2.1.2 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type p2mp
ipsec policy policy1
nhp entry 10.2.1.1 1.1.1.10 register
#
ospf 1 router-id 10.2.1.2
area 0.0.0.0
network 10.1.2.0 0.0.0.255
network 10.2.1.0 0.0.0.255
#
ospf 2
area 0.0.0.1
network 1.1.2.0 0.0.0.255
#
return
```

● Spoke2 configuration file

```
#
sysname Spoke2
#
acl number 3101
rule 5 permit ip source 10.1.3.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
rule 10 permit ip source 10.1.3.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec proposal pro1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-256
#
ike proposal 1
encryption-algorithm
aes-256
dh
group14
authentication-algorithm sha2-256
prf aes-xcbc-128
#
ike peer spoke2
pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iV100U@DraF58J4=;%^%#
ike-proposal 1
dpd type periodic
dpd idle-time 40
remote-address 10.2.1.1
#
ipsec policy policy1 10 isakmp
security acl 3101
```

```

ike-peer spoke2
proposal prol
#
interface GigabitEthernet1/0/0
ip address dhcp-alloc
#
interface GigabitEthernet1/0/1
ip address 10.1.3.1 255.255.255.0
#
interface Tunnel0/0/0
ip address 10.2.1.3 255.255.255.0
tunnel-protocol gre p2mp
source GigabitEthernet1/0/0
ospf network-type p2mp
ipsec policy policy1
nhrp entry 10.2.1.1 1.1.1.10 register
#
ospf 1 router-id 10.2.1.3
area 0.0.0.0
network 10.1.3.0 0.0.0.255
network 10.2.1.0 0.0.0.255
#
ospf 2
area 0.0.0.1
network 1.1.3.0 0.0.0.255
#
return
    
```

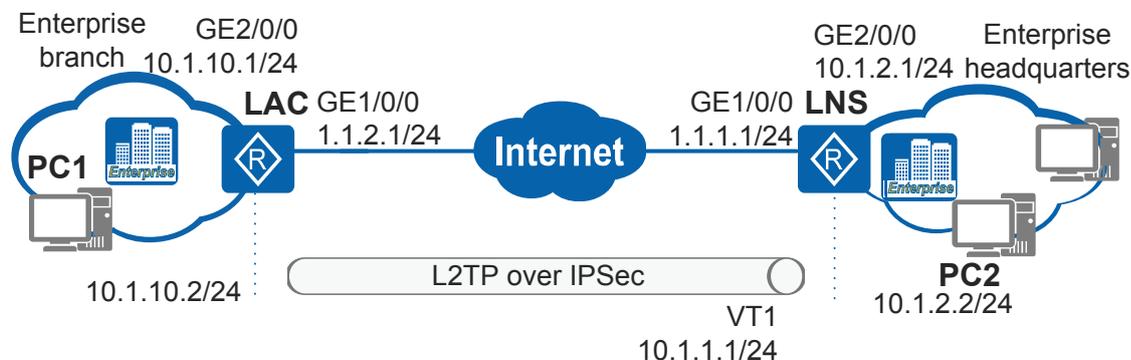
6.12.17 Example for Configuring L2TP Over IPsec to Implement Secure Communication Between the Headquarters and Branch

Networking Requirements

As shown in [Figure 6-58](#), the LAC is the enterprise branch gateway and the LNS is the enterprise headquarters gateway. The LAC automatically dials up to establish L2TP connections between the LNS for secure communication.

The enterprise requires that service packets transmitted over the L2TP tunnel be protected from being intercepted and tampered. L2TP over IPsec can be configured to encrypt service packets transmitted between the enterprise headquarters and branch.

Figure 6-58 Configuring L2TP over IPsec to implement secure communication between the headquarters and branch



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the IP address and static route on each interface to implement communication between interfaces.
2. Enable L2TP on the LAC. The PPP user sends a connection request to the LNS in the headquarters through an L2TP tunnel. After the PPP user is authenticated, the tunnel is set up.
3. On the LAC, configure a reachable route to the LNS and the enable the auto dial-up function.
4. On the LNS, configure L2TP, create a PPP user, and configure a route to the public network segment.
5. Configure an ACL to define the data flows to be protected by the IPsec tunnel.
6. Configure an IPsec proposal to define the traffic protection method.
7. Configure an IKE peer and define the attributes used for IKE negotiation.
8. Configure an IPsec policy, and apply the ACL, IPsec proposal, and IKE peers to the IPsec policy to define the data flows to be protected and protection method.
9. Apply the IPsec policy group to an interface so that the interface can protect traffic.

Procedure

Step 1 Assign IP addresses to interfaces and configure a static route to the remote device.

Assign an IP address to each interface on the LAC.

```
<Huawei> system-view
[Huawei] sysname LAC
[LAC] interface gigabitEthernet 1/0/0
[LAC-GigabitEthernet1/0/0] ip address 1.1.2.1 255.255.255.0
[LAC-GigabitEthernet1/0/0] quit
[LAC] interface gigabitEthernet 2/0/0
[LAC-GigabitEthernet2/0/0] ip address 10.1.10.1 255.255.255.0
[LAC-GigabitEthernet2/0/0] quit
```

Configure a public network route on the LAC to implement a reachable route to the LNS. A static route is used in this example, and the next-hop IP address is **1.1.2.2**.

```
[LAC] ip route-static 1.1.1.1 255.255.255.0 1.1.2.2
```

Assign an IP address to each interface on the LNS.

```
<Huawei> system-view
[Huawei] sysname LNS
[LNS] interface gigabitEthernet 1/0/0
[LNS-GigabitEthernet1/0/0] ip address 1.1.1.1 255.255.255.0
[LNS-GigabitEthernet1/0/0] quit
[LNS] interface gigabitEthernet 2/0/0
[LNS-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[LNS-GigabitEthernet2/0/0] quit
```

Configure a public network route on the LNS to implement a reachable route to the LAC. A static route is used in this example, and the next-hop IP address is **1.1.1.2**.

```
[LNS] ip route-static 1.1.2.1 255.255.255.0 1.1.1.2
```

Step 2 Configure L2TP.

On the LAC, enable L2TP globally, create an L2TP group, and configure the user **huawei** to establish an L2TP connection to the LNS.

```
[LAC] l2tp enable
[LAC] l2tp-group 1
[LAC-l2tp1] tunnel name lac
[LAC-l2tp1] start l2tp ip 1.1.1.1 fullusername huawei
```

Enable tunnel authentication and set the tunnel password on the LAC.

```
[LAC-l2tp1] tunnel authentication
[LAC-l2tp1] tunnel password cipher huawei
[LAC-l2tp1] quit
```

Configure the user name and password, PPP authentication, and IP address for the virtual PPP user on the LAC.

```
[LAC] interface virtual-template 1
[LAC-Virtual-Template1] ppp chap user huawei
[LAC-Virtual-Template1] ppp chap password cipher Huawei@1234
[LAC-Virtual-Template1] ip address ppp-negotiate
[LAC-Virtual-Template1] quit
```

Enable the LAC to dial up and establish an L2TP tunnel.

```
[LAC] interface virtual-template 1
[LAC-Virtual-Template1] l2tp-auto-client enable
[LAC-Virtual-Template1] quit
```

Configure a private network route on the LAC, so users in the enterprise branch can communicate with users in the headquarters.

```
[LAC] ip route-static 10.1.2.0 255.255.255.0 virtual-template 1
```

Configure AAA authentication, user name **huawei**, and password **Huawei@1234** on the LNS.

```
[LNS] aaa
[LNS-aaa] local-user huawei password
Please configure the login password (8-128)
It is recommended that the password consist of at least 2 types of characters, including lowercase letters, uppercase letters, numerals and special characters.
Please enter password:
Please confirm password:
Info: Add a new user.
Warning: The new user supports all access modes. The management user access modes such as Telnet, SSH, FTP, HTTP, and Terminal have security risks. You are advised to configure the required access modes only.
[LNS-aaa] local-user huawei service-type ppp
[LNS-aaa] quit
```

Configure an IP address pool for the LNS and assign an IP address to the dial-up interface of the LAC.

```
[LNS] ip pool 1
[LNS-ip-pool-1] network 10.1.1.0 mask 24
[LNS-ip-pool-1] gateway-list 10.1.1.1
[LNS-ip-pool-1] quit
```

Create a virtual interface template and configure PPP negotiation parameters on the LNS.

```
[LNS] interface virtual-template 1
[LNS-Virtual-Template1] ppp authentication-mode chap
[LNS-Virtual-Template1] remote address pool 1
[LNS-Virtual-Template1] ip address 10.1.1.1 255.255.255.0
[LNS-Virtual-Template1] quit
```

Enable L2TP and configure an L2TP group on the LNS.

```
[LNS] l2tp enable
[LNS] l2tp-group 1
```

Configure the LNS tunnel name and specify the LAC tunnel name.

```
[LNS-12tp1] tunnel name lns
[LNS-12tp1] allow 12tp virtual-template 1 remote lac
```

Enable the tunnel authentication function, and configure an authentication password.

```
[LNS-12tp1] tunnel authentication
[LNS-12tp1] tunnel password cipher huawei
[LNS-12tp1] quit
```

Configure a private network route on the LNS, so users in the headquarters can communicate with users in the enterprise branch.

```
[LNS] ip route-static 10.1.10.0 255.255.255.0 virtual-template 1
```

Step 3 Configure an ACL to define the traffic to be protected.

Configure an ACL on the LAC.

```
[LAC] acl number 3101
[LAC-acl-adv-3101] rule permit ip source 1.1.2.0 0.0.0.255 destination 1.1.1.0
0.0.0.255
[LAC-acl-adv-3101] quit
```

Configure an ACL on the LNS.

```
[LNS] acl number 3101
[LNS-acl-adv-3101] rule permit ip source 1.1.1.0 0.0.0.255 destination 1.1.2.0
0.0.0.255
[LNS-acl-adv-3101] quit
```

Step 4 Create an IPsec proposal.

Create an IPsec proposal on the LAC.

```
[LAC] ipsec proposal tran1
[LAC-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[LAC-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[LAC-ipsec-proposal-tran1] quit
```

Create an IPsec proposal on the LNS.

```
[LNS] ipsec proposal tran1
[LNS-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[LNS-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[LNS-ipsec-proposal-tran1] quit
```

Step 5 Configure IKE peers.

Configure an IKE proposal on the LAC.

```
[LAC] ike proposal 5
[LAC-ike-proposal-5] encryption-algorithm aes-128
[LAC-ike-proposal-5] authentication-algorithm sha2-256
[LAC-ike-proposal-5] dh group14
[LAC-ike-proposal-5] quit
```

Create an IKE peer on the LAC and set the pre-shared key and remote ID based on the default configuration.

```
[LAC] ike peer spub
[LAC-ike-peer-spub] ike-proposal 5
[LAC-ike-peer-spub] pre-shared-key cipher Huawei@1234
[LAC-ike-peer-spub] remote-address 1.1.1.1
[LAC-ike-peer-spub] quit
```

Configure an IKE proposal on the LNS.

```
[LNS] ike proposal 5
[LNS-ike-proposal-5] encryption-algorithm aes-128
[LNS-ike-proposal-5] authentication-algorithm sha2-256
```

```
[LNS-ike-proposal-5] dh group14
[LNS-ike-proposal-5] quit
```

Create an IKE peer on the LNS and set the pre-shared key and remote ID based on the default configuration.

```
[LNS] ike peer spua
[LNS-ike-peer-spua] ike-proposal 5
[LNS-ike-peer-spua] pre-shared-key cipher Huawei@1234
[LNS-ike-peer-spua] remote-address 1.1.2.1
[LNS-ike-peer-spua] quit
```

Step 6 Create an IPsec policy.

Configure an IPsec policy in IKE negotiation mode on the LAC.

```
[LAC] ipsec policy map1 10 isakmp
[LAC-ipsec-policy-isakmp-map1-10] ike-peer spub
[LAC-ipsec-policy-isakmp-map1-10] proposal tran1
[LAC-ipsec-policy-isakmp-map1-10] security acl 3101
[LAC-ipsec-policy-isakmp-map1-10] quit
```

Configure an IPsec policy in IKE negotiation mode on the LNS.

```
[LNS] ipsec policy use1 10 isakmp
[LNS-ipsec-policy-isakmp-use1-10] ike-peer spua
[LNS-ipsec-policy-isakmp-use1-10] proposal tran1
[LNS-ipsec-policy-isakmp-use1-10] security acl 3101
[LNS-ipsec-policy-isakmp-use1-10] quit
```

Step 7 Apply the IPsec policy group to an interface so that the interface can protect traffic.

Apply an IPsec policy group to the interface of the LAC.

```
[LAC] interface gigabitethernet 1/0/0
[LAC-GigabitEthernet1/0/0] ipsec policy map1
[LAC-GigabitEthernet1/0/0] quit
```

Apply an IPsec policy group to the interface of the LNS.

```
[LNS] interface gigabitethernet 1/0/0
[LNS-GigabitEthernet1/0/0] ipsec policy use1
[LNS-GigabitEthernet1/0/0] quit
```

Step 8 Verify the configuration.

After the configurations are complete, PC1 can ping PC2 successfully. The data transmitted between PC1 and PC2 is encrypted. Run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on the LAC to view the SAs established through IKE negotiation.

```
[LAC] display ike sa
IKE SA information :
Conn-ID      Peer          VPN    Flag(s)    Phase    RemoteType  RemoteID
-----
16           1.1.1.1:500  VPN1   RD|ST      v2:2     IP          1.1.1.1
14           1.1.1.1:500  VPN1   RD|ST      v2:1     IP          1.1.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

Run the **display l2tp tunnel** command on the LAC or LNS to view L2TP tunnel and session information. The command output for the LAC is shown as an example.

```
[LAC] display l2tp tunnel

Total tunnel : 1
LocalTID RemoteTID RemoteAddress      Port    Sessions RemoteName
1         1         1.1.1.1          1701    1         lns
```

---End

Configuration Files

- Configuration file of the LAC

```
#
sysname LAC
#
l2tp enable
#
acl number 3101
rule 5 permit ip source 1.1.2.0 0.0.0.255 destination 1.1.1.0 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer spub
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
 ike-proposal 5
 remote-address 1.1.1.1
#
ipsec policy map1 10 isakmp
 security acl 3101
 ike-peer spub
 proposal tran1
#
interface Virtual-Template1
 ppp chap user huawei
 ppp chap password cipher %@@@U>upTZ}mQM:rhRL:4;s$, (xf%@@@
 ip address ppp-negotiate
 l2tp-auto-client enable
#
interface GigabitEthernet1/0/0
 ip address 1.1.2.1 255.255.255.0
 ipsec policy map1
#
interface GigabitEthernet2/0/0
 ip address 10.1.10.1 255.255.255.0
#
l2tp-group 1
 tunnel password cipher %@@@/-#)Lg[S4F:#2~ZNvqa$}\DL%@@@
 tunnel name lac
 start l2tp ip 1.1.1.1 fullusername huawei
#
ip route-static 1.1.1.1 255.255.255.0 1.1.2.2
ip route-static 10.1.2.0 255.255.255.0 Virtual-Template1
#
return
```

- Configuration file of the LNS

```
#
sysname LNS
#
l2tp enable
#
acl number 3101
rule 5 permit ip source 1.1.1.0 0.0.0.255 destination 1.1.2.0 0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer spua
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%^%#
ike-proposal 5
remote-address 1.1.2.1
#
ipsec policy use1 10 isakmp
security acl 3101
ike-peer spua
proposal tran1
#
ip pool 1
network 10.1.1.0 mask 255.255.255.0
gateway-list 10.1.1.1
#
aaa
local-user huawei password cipher $1a$_<`.CO&(:LeS/$#F
\H0Qv8B]KAZja3)3q'RNx;VI$
local-user huawei service-type ppp
#
interface Virtual-Templat1
ppp authentication-mode chap
remote address pool 1
ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet1/0/0
ip address 1.1.1.1 255.255.255.0
ipsec policy use1
#
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
l2tp-group 1
allow l2tp virtual-template 1 remote lac
tunnel password cipher %%@EB~j7Je>;@>uNr'D=J<]WL%@@@
tunnel name lns
#
ip route-static 1.1.2.1 255.255.255.0 1.1.1.2
ip route-static 10.1.10.0 255.255.255.0 Virtual-Templat1
#
return
```

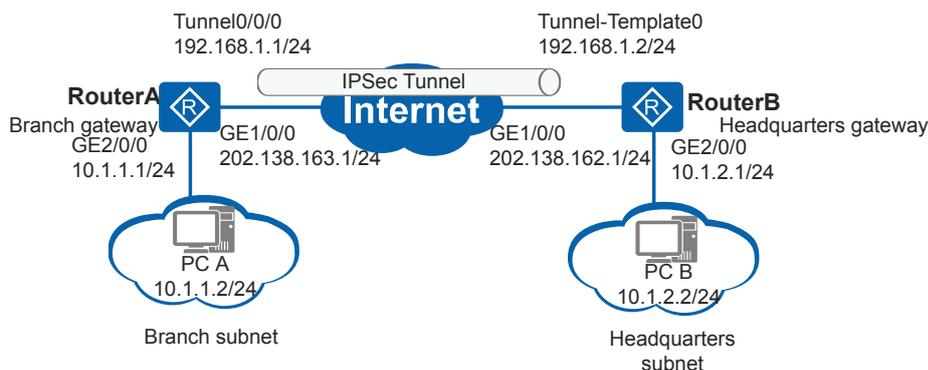
6.12.18 Example for Configuring a Tunnel Template Interface for IPsec Tunnel Setup

Networking Requirements

As shown in [Figure 6-59](#), enterprise's branch and headquarters communicate through the public network. However, the topologies of headquarters and branch networks change frequently. The enterprise requires to protect traffic transmitted over the public network between the branch and headquarters, and the enterprise hopes that the IPsec configuration does not change when the network topologies change.

1. The branch gateway RouterA and headquarters gateway RouterB can set up an IPsec tunnel over the public network to protect traffic between them.
2. The topologies of headquarters and branch networks change frequently, the IPsec tunnel needs to be set up using tunnel interfaces, and information about the subnet and interface to be protected by IPsec needs to be configured locally.

Figure 6-59 Configuring a virtual tunnel template interface for IPsec Tunnel setup



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes on the interfaces to implement communication between them.
2. Configure ACLs to define the subnet that the local device needs to protect.
3. Configure AAA service schemes to define the subnet route information and the ip-address interface that the local device needs to send.
4. Configure IPsec proposals to define the data flow protection method.
5. Configure IKE peers and define the attributes used for IKE negotiation.
6. Configure IPsec profiles, and apply the IPsec proposal and IKE peers to the IPsec profile to define the data flows to be protected and protection method.
7. Apply the IPsec profiles to the tunnel template interface and tunnel interface respectively to enable IPsec protection on the interfaces.

Procedure

Step 1 Configure IP addresses and static routes on the interfaces of RouterA and RouterB.

Configure an IP address for each interface of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route from RouterA to RouterB. This example assumes that the next hop address of the route is 202.138.163.2.

```
[RouterA] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
```

Configure an IP address for each interface of RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route from RouterB to RouterA. This example assumes that the next hop address of the route is 202.138.162.2.

```
[RouterB] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
```

Step 2 Configure ACLs to define the subnet that the local device needs to protect.

Configure an ACL on RouterA to permit data flows with the source address 10.1.1.0/24 to pass through.

```
[RouterA] acl number 3001
[RouterA-acl-adv-3001] rule permit ip source 10.1.1.0 0.0.0.255
[RouterA-acl-adv-3001] quit
```

Configure an ACL on RouterB to permit data flows with the source address 10.1.2.0/24 to pass through.

```
[RouterB] acl number 3001
[RouterB-acl-adv-3001] rule permit ip source 10.1.2.0 0.0.0.255
[RouterB-acl-adv-3001] quit
```

Step 3 Configure AAA service schemes to define the subnet route information that the local device needs to send.

Configure an AAA service scheme on RouterA.

```
[RouterA] aaa
[RouterA-aaa] service-scheme schemetest
[RouterA-aaa-service-schemetest] route set acl 3001
[RouterA-aaa-service-schemetest] route set interface
[RouterA-aaa-service-schemetest] quit
[RouterA-aaa] quit
```

Configure an AAA service scheme on RouterB.

```
[RouterB] aaa
[RouterB-aaa] service-scheme schemetest
```

```
[RouterB-aaa-service-schemetest] route set acl 3001
[RouterB-aaa-service-schemetest] route set interface
[RouterB-aaa-service-schemetest] quit
[RouterB-aaa] quit
```

Step 4 Create IPsec proposals on RouterA and RouterB.

Create an IPsec proposal on RouterA.

```
[RouterA] ipsec proposal prop1
[RouterA-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-prop1] quit
```

Create an IPsec proposal on RouterB.

```
[RouterB] ipsec proposal prop1
[RouterB-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop1] quit
```

Run the **display ipsec proposal** command on RouterA and RouterB to view the configuration of the IPsec proposal.

Step 5 Create IKE peers on RouterA and RouterB.

Create an IKE proposal on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
```

Create an IKE peer on RouterA.

```
[RouterA] ike peer peer2
[RouterA-ike-peer-peer2] ike-proposal 5
[RouterA-ike-peer-peer2] pre-shared-key cipher Huawei@1234
[RouterA-ike-peer-peer2] service-scheme schemetest
[RouterA-ike-peer-peer2] config-exchange request
[RouterA-ike-peer-peer2] config-exchange set accept
[RouterA-ike-peer-peer2] config-exchange set send
[RouterA-ike-peer-peer2] route accept
[RouterA-ike-peer-peer2] quit
```

Create an IKE proposal on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
```

Create an IKE peer on RouterB.

```
[RouterB] ike peer peer2
[RouterB-ike-peer-peer2] ike-proposal 5
[RouterB-ike-peer-peer2] pre-shared-key cipher Huawei@1234
[RouterB-ike-peer-peer2] service-scheme schemetest
[RouterB-ike-peer-peer2] config-exchange set accept
[RouterB-ike-peer-peer2] config-exchange set send
[RouterB-ike-peer-peer2] route accept
[RouterB-ike-peer-peer2] quit
```

Step 6 Create IPsec profiles on RouterA and RouterB respectively.

Create an IPsec profile on RouterA.

```
[RouterA] ipsec profile profile1
[RouterA-ipsec-profile-profile1] proposal prop1
```

```
[RouterA-ipsec-profile-profile1] ike-peer peer2
[RouterA-ipsec-profile-profile1] quit
```

Create an IPsec profile on RouterB.

```
[RouterB] ipsec profile profile1
[RouterB-ipsec-profile-profile1] proposal prop1
[RouterB-ipsec-profile-profile1] ike-peer peer2
[RouterB-ipsec-profile-profile1] quit
```

Step 7 Apply the IPsec profiles to the interfaces of RouterA and RouterB.

Apply the IPsec profile to the interface of RouterA.

```
[RouterA] interface tunnel 0/0/0
[RouterA-Tunnel0/0/0] ip address 192.168.1.1 255.255.255.0
[RouterA-Tunnel0/0/0] tunnel-protocol ipsec
[RouterA-Tunnel0/0/0] source gigabitethernet1/0/0
[RouterA-Tunnel0/0/0] destination 202.138.162.1
[RouterA-Tunnel0/0/0] ipsec profile profile1
[RouterA-Tunnel0/0/0] quit
```

Apply the IPsec profile to the interface of RouterB.

```
[RouterB] interface loopback0
[RouterB-LoopBack0] ip address 192.168.1.2 255.255.255.255
[RouterB-LoopBack0] quit
[RouterB] interface tunnel-template 0
[RouterB-Tunnel-Template0] ip address unnumbered interface loopback0
[RouterB-Tunnel-Template0] tunnel-protocol ipsec
[RouterB-Tunnel-Template0] source gigabitethernet1/0/0
[RouterB-Tunnel-Template0] ipsec profile profile1
[RouterB-Tunnel-Template0] quit
```

Run the **display ipsec profile** command on RouterA and RouterB to view the IPsec profile configuration.

Step 8 Verify the configuration.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  16      202.138.162.1:500          RD|ST  v2:2     IP     202.138.162.1
  14      202.138.162.1:500          RD|ST  v2:1     IP     202.138.162.1

Number of IKE SA : 2
-----
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

Run the **display ip routing-table** command on RouterA and RouterB to view route information. This example only shows information about subnet routes that are successfully sent.

```
[RouterA] display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 16      Routes : 16

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
  10.1.2.0/24      Unr    0    0        D   192.168.1.2      Tunnel0/0/0
```

```
[RouterB] display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
  Destinations : 16          Routes : 16

Destination/Mask    Proto   Pre  Cost           Flags NextHop         Interface
-----
 10.1.1.0/24      Unr     62   0             RD    192.168.1.1       Tunnel-Template0
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3001
 rule 5 permit ip source 10.1.1.0 0.0.0.255
#
ipsec proposal prop1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer peer2
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
 ike-proposal 5
 service-scheme schemetest
 route accept
 config-exchange request
 config-exchange set accept
 config-exchange set send
#
ipsec profile profile1
 ike-peer peer2
 proposal prop1
#
aaa
 service-scheme schemetest
 route set acl 3001
 route set interface
#
interface GigabitEthernet1/0/0
 ip address 202.138.163.1 255.255.255.0
#
interface Tunnel0/0/0
 ip address 192.168.1.1 255.255.255.0
 tunnel-protocol ipsec
 source GigabitEthernet1/0/0
 destination 202.138.162.1
 ipsec profile profile1
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2
#
return
```

- Configuration file of RouterB

```
#
 sysname RouterB
#
 acl number 3001
  rule 5 permit ip source 10.1.2.0 0.0.0.255
#
 ipsec proposal propl
  esp authentication-algorithm sha2-256
  esp encryption-algorithm aes-128
#
 ike proposal 5
  encryption-algorithm aes-128
  dh group14
  authentication-algorithm sha2-256
  authentication-method pre-share
  integrity-algorithm hmac-sha2-256
  prf hmac-sha2-256
#
 ike peer peer2
  pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%^%#
  ike-proposal 5
  service-scheme schemetest
  route accept
  config-exchange set accept
  config-exchange set send
#
 ipsec profile profile1
  ike-peer peer2
  proposal propl
#
 aaa
  service-scheme schemetest
  route set acl 3001
  route set interface
#
 interface GigabitEthernet1/0/0
  ip address 202.138.162.1 255.255.255.0
#
 interface GigabitEthernet2/0/0
  ip address 10.1.2.1 255.255.255.0
#
 interface Tunnel-Template0
  ip address unnumbered interface LoopBack0
  tunnel-protocol ipsec
  source GigabitEthernet1/0/0
  ipsec profile profile1
#
 interface LoopBack0
  ip address 192.168.1.2 255.255.255.255
#
 ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
#
return
```

6.12.19 Example for Establishing an IPsec Tunnel Using an Efficient VPN Policy in Client Mode

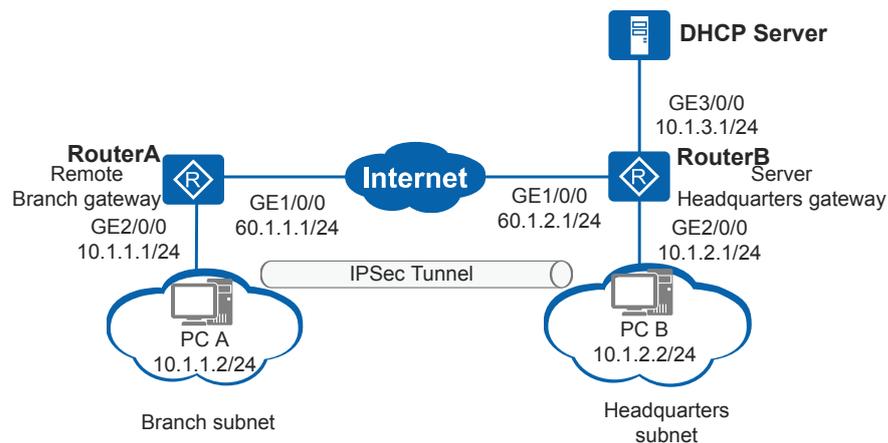
Networking Requirements

As shown in [Figure 6-60](#), RouterA (remote small-scale branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The headquarters and branch networks are not planned uniformly. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24. The DHCP server is located on the headquarters network and allocates an IP address to the branch gateway.

The enterprise requires that traffic between headquarters and branch networks should be securely transmitted and the headquarters gateway should manage the branch gateway with simplified configuration in centralized manner. An Efficient VPN policy in client mode can be used to establish an IPsec tunnel to protect traffic. This method facilitates IPsec tunnel establishment and maintenance.

In client mode, RouterA requests an IP address from RouterB to establish an IPsec tunnel, and requests the DNS domain name, DNS server IP addresses, and WINS server IP addresses for the branch subnet.

Figure 6-60 Establishing an IPsec tunnel using an Efficient VPN policy in client mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Configure the DHCP server address on RouterB so that IP addresses can be dynamically allocated through DHCP.
3. Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.
4. Configure an Efficient VPN policy in client mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
```

```
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0  
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 60.1.1.2.

```
[RouterA] ip route-static 60.1.2.0 255.255.255.0 60.1.1.2  
[RouterA] ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB. The IP address of GigabitEthernet4/0/0 must be on the same network segment as the IP address assigned by the DHCP server.

```
<Huawei> system-view  
[Huawei] sysname RouterB  
[RouterB] interface gigabitethernet 1/0/0  
[RouterB-GigabitEthernet1/0/0] ip address 60.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet1/0/0] quit  
[RouterB] interface gigabitethernet 2/0/0  
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0  
[RouterB-GigabitEthernet2/0/0] quit  
[RouterB] interface gigabitethernet 3/0/0  
[RouterB-GigabitEthernet3/0/0] ip address 10.1.3.1 255.255.255.0  
[RouterB-GigabitEthernet3/0/0] quit  
[RouterB] interface gigabitethernet 4/0/0  
[RouterB-GigabitEthernet4/0/0] ip address 100.1.1.3 255.255.255.0  
[RouterB-GigabitEthernet4/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.2.2.

```
[RouterB] ip route-static 60.1.1.0 255.255.255.0 60.1.2.2  
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.2.2  
[RouterB] ip route-static 100.1.1.0 255.255.255.0 60.1.2.2
```

Step 2 Configure the DHCP server address on RouterB so that IP addresses can be dynamically allocated through DHCP.

Enable DHCP, create a DHCP server group, and add DHCP servers to the DHCP server group.

```
[RouterB] dhcp enable  
[RouterB] dhcp server group dhcp-ser1  
[RouterB-dhcp-server-group-dhcp-ser1] dhcp-server 10.1.3.2  
[RouterB-dhcp-server-group-dhcp-ser1] gateway 100.1.1.3  
[RouterB-dhcp-server-group-dhcp-ser1] quit
```

Step 3 Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

In the service scheme view, configure the resources to be allocated, including the IP address, DNS domain name, DNS server IP addresses, and WINS server IP addresses.

```
[RouterB] aaa  
[RouterB-aaa] service-scheme schemetest  
[RouterB-aaa-service-schemetest] dhcp-server group dhcp-ser1  
[RouterB-aaa-service-schemetest] dns-name mydomain.com.cn  
[RouterB-aaa-service-schemetest] dns 2.2.2.2  
[RouterB-aaa-service-schemetest] dns 2.2.2.3 secondary  
[RouterB-aaa-service-schemetest] wins 3.3.3.2  
[RouterB-aaa-service-schemetest] wins 3.3.3.3 secondary  
[RouterB-aaa-service-schemetest] quit  
[RouterB-aaa] quit
```

Configure an IKE proposal and an IKE peer, and bind the service scheme to the IKE peer.

```
[RouterB] ike proposal 5  
[RouterB-ike-proposal-5] dh group14  
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
```

```
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut3
[RouterB-ike-peer-rut3] pre-shared-key cipher huawei
[RouterB-ike-peer-rut3] ike-proposal 5
[RouterB-ike-peer-rut3] service-scheme schemetest
[RouterB-ike-peer-rut3] quit
```

Configure an IPsec proposal and establish an IPsec policy using an IPsec policy template.

```
[RouterB] ipsec proposal prop1
[RouterB-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop1] quit
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut3
[RouterB-ipsec-policy-templet-temp1-10] proposal prop1
[RouterB-ipsec-policy-templet-temp1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Apply the IPsec policy to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 4 Configure an Efficient VPN policy in client mode on RouterA to establish an IPsec tunnel.

Configure an Efficient VPN policy in client mode and specify the remote address and pre-shared key.

```
[RouterA] ipsec efficient-vpn evpn mode client
[RouterA-ipsec-efficient-vpn-evpn] remote-address 60.1.2.1 v2
[RouterA-ipsec-efficient-vpn-evpn] pre-shared-key cipher huawei
[RouterA-ipsec-efficient-vpn-evpn] dh group14
[RouterA-ipsec-efficient-vpn-evpn] quit
```

Apply the Efficient VPN policy to the interface.

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec efficient-vpn evpn
[RouterA-GigabitEthernet1/0/0] quit
```

Step 5 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA. The following information is displayed:

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
   26     60.1.2.1:500             RD|ST   v2:2     IP     60.1.2.1
   25     60.1.2.1:500             RD|ST   v2:1     IP     60.1.2.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
```

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
ipsec efficient-vpn evpn mode client
remote-address 60.1.2.1 v2
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
dh group14
#
interface GigabitEthernet1/0/0
ip address 60.1.1.1 255.255.255.0
ipsec efficient-vpn evpn
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
dhcp enable
#
ipsec proposal prop1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut3
pre-shared-key cipher %^%#K{JG:rWVHPMnf;5\||,GW(Luq'qi8BT4nOj%5W5=)%^%#
ike-proposal 5
service-scheme schemetest
#
ipsec policy-template templ 10
ike-peer rut3
proposal prop1
#
ipsec policy policy1 10 isakmp template templ
#
dhcp server group dhcp-ser1
dhcp-server 10.1.3.2 0
gateway 100.1.1.3
#
aaa
service-scheme schemetest
dns 2.2.2.2
dns 2.2.2.3 secondary
dhcp-server group dhcp-ser1
wins 3.3.3.2
wins 3.3.3.3 secondary
dns-name mydomain.com.cn
#
interface GigabitEthernet1/0/0
ip address 60.1.2.1 255.255.255.0
ipsec policy policy1
#
```

```
interface GigabitEthernet2/0/0
ip address 10.1.2.1 255.255.255.0
#
interface GigabitEthernet3/0/0
ip address 10.1.3.1 255.255.255.0
#
interface GigabitEthernet4/0/0
ip address 100.1.1.3 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
ip route-static 100.1.1.0 255.255.255.0 60.1.2.2
#
return
```

6.12.20 Example for Configuring an IPsec Tunnel Using an Efficient VPN Policy in Network Mode

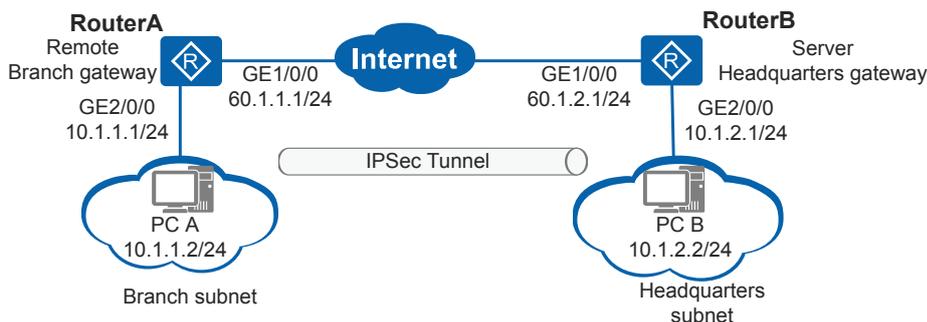
Networking Requirements

As shown in **Figure 6-61**, RouterA (remote branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The headquarters and branch networks are planned uniformly. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise requires that traffic between headquarters and branch networks should be securely transmitted and the headquarters gateway should manage the branch gateway with simplified configuration in centralized manner. An Efficient VPN policy in network mode can be used to establish an IPsec tunnel to protect traffic. This method facilitates IPsec tunnel establishment and maintenance.

In network mode, RouterA does not request an IP address from RouterB, and uses the original IP address to establish an IPsec tunnel with RouterB. RouterA applies to RouterB for the DNS domain name, DNS server IP addresses, and WINS server IP addresses for the branch subnet.

Figure 6-61 Establishing an IPsec tunnel using an Efficient VPN policy in network mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.

2. Configure an Efficient VPN policy in network mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.
3. On RouterB, configure the resources to be allocated, including the DNS server IP addresses, and WINS server IP addresses.
4. Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 60.1.1.2.

```
[RouterA] ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.2.2.

```
[RouterB] ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
```

Step 2 Configure an Efficient VPN policy in network mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3001
[RouterA-acl-adv-3001] rule 1 permit ip source 10.1.1.2 0.0.0.255 destination
10.1.2.2 0.0.0.255
```

Configure an Efficient VPN policy in network mode and specify the ACL, remote address, and pre-shared key.

```
[RouterA] ipsec efficient-vpn evpn mode network
[RouterA-ipsec-efficient-vpn-evpn] security acl 3001
[RouterA-ipsec-efficient-vpn-evpn] remote-address 60.1.2.1 v2
[RouterA-ipsec-efficient-vpn-evpn] pre-shared-key cipher Huawei@1234
[RouterA-ipsec-efficient-vpn-evpn] dh group14
[RouterA-ipsec-efficient-vpn-evpn] quit
```

Apply the Efficient VPN policy to the interface.

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec efficient-vpn evpn
```

Step 3 Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

In the service scheme view, configure the resources to be allocated, including the DNS domain name, DNS server IP addresses, and WINS server IP addresses.

```
[RouterB] aaa
[RouterB-aaa] service-scheme schemetest
[RouterB-aaa-service-schemetest] dns-name mydomain.com.cn
[RouterB-aaa-service-schemetest] dns 2.2.2.2
[RouterB-aaa-service-schemetest] dns 2.2.2.3 secondary
[RouterB-aaa-service-schemetest] wins 3.3.3.2
[RouterB-aaa-service-schemetest] wins 3.3.3.3 secondary
[RouterB-aaa-service-schemetest] quit
[RouterB-aaa] quit
```

Configure an IKE proposal and an IKE peer.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut3
[RouterB-ike-peer-rut3] pre-shared-key cipher Huawei@1234
[RouterB-ike-peer-rut3] ike-proposal 5
[RouterB-ike-peer-rut3] service-scheme schemetest
[RouterB-ike-peer-rut3] quit
```

Configure an IPsec proposal and establish an IPsec policy using an IPsec policy template.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
[RouterB] ipsec policy-template use1 10
[RouterB-ipsec-policy-templet-use1-10] ike-peer rut3
[RouterB-ipsec-policy-templet-use1-10] proposal tran1
[RouterB-ipsec-policy-templet-use1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template use1
```

Apply the IPsec policy to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 4 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer                VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  26       60.1.2.1:500                RD|ST   v2:2    IP     60.1.2.1
  25       60.1.2.1:500                RD|ST   v2:1    IP     60.1.2.1

Number of IKE SA : 2
-----
```

```
Flag Description:
RD--READY      ST--STAYALIVE  RL--REPLACED  FD--FADING    TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE      S--STANDBY    A--ALONE      NEG--NEGOTIATING
```

---End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3001
rule 1 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255
#
ipsec efficient-vpn evpn mode network
remote-address 60.1.2.1 v2
pre-shared-key cipher %%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\!*!%^%#
security acl 3001
dh group14
#
interface GigabitEthernet1/0/0
ip address 60.1.1.1 255.255.255.0
ipsec efficient-vpn evpn
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut3
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%%#
ike-proposal 5
service-scheme schemetest
#
ipsec policy-template use1 10
ike-peer rut3
proposal tran1
#
ipsec policy policy1 10 isakmp template use1
#
aaa
service-scheme schemetest
dns 2.2.2.2
dns 2.2.2.3 secondary
wins 3.3.3.2
wins 3.3.3.3 secondary
dns-name mydomain.com.cn
```

```
#
interface GigabitEthernet1/0/0
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
#
return
```

6.12.21 Example for Configuring an IPsec Tunnel Using an Efficient VPN Policy in Network-Plus Mode

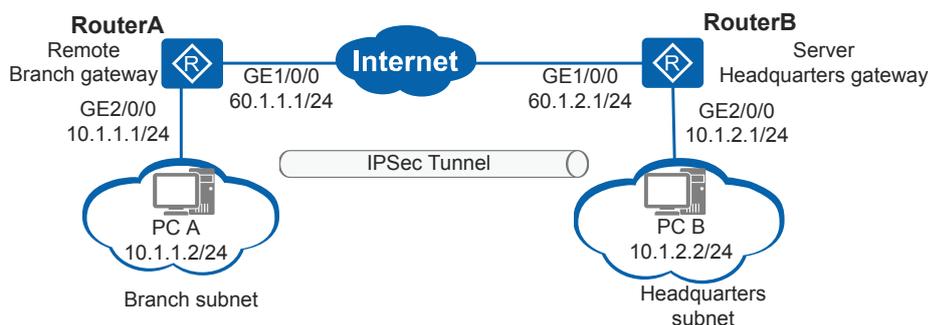
Networking Requirements

As shown in [Figure 6-62](#), RouterA (remote branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The headquarters and branch networks are planned uniformly. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24.

The enterprise requires that traffic between headquarters and branch networks should be securely transmitted and the headquarters gateway should manage the branch gateway with simplified configuration in centralized manner. Ping and Telnet techniques are used for management and maintenance. An Efficient VPN policy in network-plus mode can be used to establish an IPsec tunnel to protect traffic. This method facilitates IPsec tunnel establishment and maintenance.

In network-plus mode, RouterA requests an IP address from RouterB to establish an IPsec tunnel, and requests the DNS domain name, DNS server IP addresses, and WINS server IP addresses for the branch subnet. The obtained IP address is used for the headquarters to perform ping and Telnet operations for the branch.

Figure 6-62 Establishing an IPsec tunnel using an Efficient VPN policy in network-plus mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.

2. Configure an Efficient VPN policy in network-plus mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.
3. On RouterB, configure the resources to be allocated, including the IP address, DNS domain name, DNS server IP addresses, and WINS server IP addresses.
4. Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

Procedure

Step 1 Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 60.1.1.2.

```
[RouterA] ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.2.2.

```
[RouterB] ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 100.1.1.0 255.255.255.0 60.1.2.2
```

Step 2 Configure an Efficient VPN policy in network-plus mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.

Configure an ACL on RouterA to define data flows sent from 10.1.1.0/24 to 10.1.2.0/24.

```
[RouterA] acl number 3001
[RouterA-acl-adv-3001] rule 1 permit ip source 10.1.1.0 0.0.0.255 destination
10.1.2.0 0.0.0.255
[RouterA-acl-adv-3001] quit
```

Configure an Efficient VPN policy in network-plus mode and specify the ACL, remote address, and pre-shared key.

```
[RouterA] ipsec efficient-vpn evpn mode network-plus
[RouterA-ipsec-efficient-vpn-evpn] security acl 3001
[RouterA-ipsec-efficient-vpn-evpn] remote-address 60.1.2.1 v1
[RouterA-ipsec-efficient-vpn-evpn] pre-shared-key cipher Huawei@1234
[RouterA-ipsec-efficient-vpn-evpn] dh group14
[RouterA-ipsec-efficient-vpn-evpn] quit
```

Apply the Efficient VPN policy to the interface.

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec efficient-vpn evpn
```

Step 3 Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

Configure the resources to be allocated, including the IP address, DNS domain name, DNS server IP addresses, and WINS server IP addresses.

```
[RouterB] ip pool pol
[RouterB-ip-pool-pol] network 100.1.1.0 mask 255.255.255.128
[RouterB-ip-pool-pol] gateway-list 100.1.1.1
[RouterB-ip-pool-pol] quit
[RouterB] aaa
[RouterB-aaa] service-scheme schemetest
[RouterB-aaa-service-schemetest] ip-pool pol
[RouterB-aaa-service-schemetest] dns-name mydomain.com.cn
[RouterB-aaa-service-schemetest] dns 2.2.2.2
[RouterB-aaa-service-schemetest] dns 2.2.2.3 secondary
[RouterB-aaa-service-schemetest] wins 3.3.3.2
[RouterB-aaa-service-schemetest] wins 3.3.3.3 secondary
[RouterB-aaa-service-schemetest] quit
[RouterB-aaa] quit
```

Configure an IKE proposal and an IKE peer.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] encryption-algorithm aes-256
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut3
[RouterB-ike-peer-rut3] undo version 2
[RouterB-ike-peer-rut3] exchange-mode aggressive
[RouterB-ike-peer-rut3] pre-shared-key cipher Huawei@1234
[RouterB-ike-peer-rut3] ike-proposal 5
[RouterB-ike-peer-rut3] service-scheme schemetest
[RouterB-ike-peer-rut3] quit
```

Configure an IPsec proposal and establish an IPsec policy using an IPsec policy template.

```
[RouterB] ipsec proposal tran1
[RouterB-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-tran1] quit
[RouterB] ipsec policy-template use1 10
[RouterB-ipsec-policy-templet-use1-10] ike-peer rut3
[RouterB-ipsec-policy-templet-use1-10] proposal tran1
[RouterB-ipsec-policy-templet-use1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template use1
```

Apply the IPsec policy to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
```

Step 4 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID Peer           VPN   Flag(s)  Phase  RemoteType  RemoteID
```

```
-----  
118      60.1.2.1:500      RD|ST      v1:2      IP      60.1.2.1  
117      60.1.2.1:500      RD|ST      v1:2      IP      60.1.2.1  
116      60.1.2.1:500      RD|ST      v1:1      IP      60.1.2.1  
-----
```

Number of IKE SA : 3

Flag Description:

```
RD--READY      ST--STAYALIVE      RL--REPLACED      FD--FADING      TO--TIMEOUT  
HRT--HEARTBEAT      LKG--LAST KNOWN GOOD SEQ NO.      BCK--BACKED UP  
M--ACTIVE      S--STANDBY      A--ALONE      NEG--NEGOTIATING
```

---End

Configuration Files

- Configuration file of RouterA

```
#  
sysname RouterA  
#  
acl number 3001  
rule 1 permit ip source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255  
#  
ipsec efficient-vpn evpn mode network-plus  
remote-address 60.1.2.1 v1  
pre-shared-key cipher %%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%%#  
security acl 3001  
dh group14  
#  
interface GigabitEthernet1/0/0  
ip address 60.1.1.1 255.255.255.0  
ipsec efficient-vpn evpn  
#  
interface GigabitEthernet2/0/0  
ip address 10.1.1.1 255.255.255.0  
#  
ip route-static 60.1.2.0 255.255.255.0 60.1.1.2  
ip route-static 10.1.2.0 255.255.255.0 60.1.1.2  
#  
return
```

- Configuration file of RouterB

```
#  
sysname RouterB  
#  
ipsec proposal tran1  
esp authentication-algorithm sha2-256  
esp encryption-algorithm aes-128  
#  
ike proposal 5  
encryption-algorithm aes-256  
dh group14  
authentication-algorithm sha2-256  
authentication-method pre-share  
integrity-algorithm hmac-sha2-256  
prf hmac-sha2-256  
#  
ike peer rut3  
undo version 2  
exchange-mode aggressive  
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%%#  
ike-proposal 5  
service-scheme schemetest  
#  
ipsec policy-template usel 10  
ike-peer rut3  
proposal tran1
```

```
#
ipsec policy policy1 10 isakmp template use1
#
ip pool po1
 gateway-list 100.1.1.1
 network 100.1.1.0 mask 255.255.255.128
#
aaa
 service-scheme schemetest
  dns 2.2.2.2
  dns 2.2.2.3 secondary
  ip-pool po1
  wins 3.3.3.2
  wins 3.3.3.3 secondary
  dns-name mydomain.com.cn
#
interface GigabitEthernet1/0/0
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
ip route-static 100.1.1.0 255.255.255.0 60.1.2.2
#
return
```

6.12.22 Example for Configuring Efficient VPN in Network-auto-cfg Mode to Establish an IPsec Tunnel

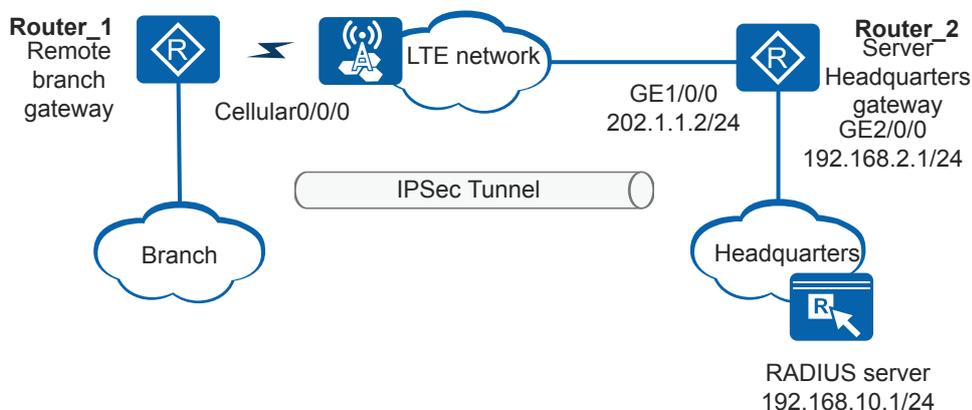
Networking Requirements

As shown in [Figure 6-63](#), Router_1 is a remote enterprise branch gateway and Router_2 is the enterprise headquarters gateway. The branch connects to the headquarters over the LTE network. The headquarters and branch networks are planned beforehand.

The enterprise requires to protect traffic transmitted between the enterprise branch and headquarters. The headquarters gateway is required to uniformly manage and maintain the branch gateway with simple configuration using the ping and Telnet technologies. The enterprise branch and headquarters communicate over the LTE network. An IPsec tunnel can be established between them using Efficient VPN in Network-auto-cfg mode to provide security protection. This mode facilitates establishment, management, and maintenance of the IPsec tunnel.

In Efficient VPN Network-auto-cfg mode, Router_1 applies for authorization information from the RADIUS server located in the headquarters. The authorization information includes an IP address for establishing the IPsec tunnel, an IP address pool for allocating addresses to users, a DNS domain name, DNS server addresses, and WINS server addresses used by the branch users. The headquarters uses the IP address to perform ping, Telnet, or other management and maintenance operations.

Figure 6-63 Configuring Efficient VPN in Network-auto-cfg mode to establish an IPsec tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a Cellular interface on Router_1 to connect it to the LTE network.
2. Configure static routes to ensure that the devices are reachable to each other over the LTE network.
3. Configure a RADIUS server template on Router_2.
4. Configure Router_2 as the responder to establish an IPsec tunnel with Router_1 in policy template mode.
5. Configure Efficient VPN in Network-auto-cfg mode on Router_1 to initiate an IPsec tunnel establishment request to Router_2.

Procedure

Step 1 Configure a Cellular interface and an APN profile.

Configure Router_1.

```
<Huawei> system-view
[Huawei] sysname Router_1
[Router_1] dialer-rule
[Router_1-dialer-rule] dialer-rule 1 ip permit
[Router_1-dialer-rule] quit
[Router_1] interface cellular 0/0/0
[Router_1-Cellular0/0/0] dialer enable-circular
[Router_1-Cellular0/0/0] ip address negotiate
[Router_1-Cellular0/0/0] dialer-group 1
[Router_1-Cellular0/0/0] dialer number *99#
[Router_1-Cellular0/0/0] mode lte auto
[Router_1-Cellular0/0/0] quit
[Router_1] apn profile lteprofile
[Router_1-apn-profile-lteprofile] apn LTENET
[Router_1-apn-profile-lteprofile] quit
[Router_1] interface cellular 0/0/0
[Router_1-Cellular0/0/0] apn-profile lteprofile
[Router_1-Cellular0/0/0] shutdown
[Router_1-Cellular0/0/0] undo shutdown
[Router_1-Cellular0/0/0] quit
```

After Cellular0/0/0 dials up successfully, it obtains the IP address 202.1.10.1/24.

Step 2 Configure IP addresses for the interfaces of Router_2.

```
<Huawei> system-view
[Huawei] sysname Router_2
[Router_2] interface gigabitethernet 1/0/0
[Router_2-GigabitEthernet1/0/0] ip address 202.1.1.2 255.255.255.0
[Router_2-GigabitEthernet1/0/0] quit
[Router_2] interface gigabitethernet 2/0/0
[Router_2-GigabitEthernet2/0/0] ip address 192.168.2.1 255.255.255.0
[Router_2-GigabitEthernet2/0/0] quit
```

Step 3 Configure static routes to ensure that the devices are reachable to each other over the LTE network.

Configure Router_1.

```
[Router_1] ip route-static 0.0.0.0 0 cellular 0/0/0
```

Configure Router_2.

```
[Router_2] ip route-static 0.0.0.0 0 202.1.1.1
```

Step 4 Configure a RADIUS server template on Router_2.

```
[Router_2] radius-server template shiva
[Router_2-radius-shiva] radius-server authentication 192.168.10.1 1812
[Router_2-radius-shiva] radius-server accounting 192.168.10.1 1813
[Router_2-radius-shiva] radius-server shared-key cipher hello
[Router_2-radius-shiva] quit
[Router_2] aaa
[Router_2-aaa] authentication-scheme rds
[Router_2-aaa-authen-rds] authentication-mode radius
[Router_2-aaa-authen-rds] quit
[Router_2-aaa] domain rds
[Router_2-aaa-domain-rds] authentication-scheme rds
[Router_2-aaa-domain-rds] radius-server shiva
[Router_2-aaa-domain-rds] quit
[Router_2-aaa] quit
```

Step 5 Configure Router_2 as the responder to establish an IPsec tunnel with Router_1 in policy template mode.

Configure an IKE proposal and an IKE peer.

```
[Router_2] ike proposal 1
[Router_2-ike-proposal-1] encryption-algorithm aes-256
[Router_2-ike-proposal-1] dh group14
[Router_2-ike-proposal-1] quit
[Router_2] ike peer rut1
[Router_2-ike-peer-rut1] undo version 2
[Router_2-ike-peer-rut1] exchange-mode aggressive
[Router_2-ike-peer-rut1] pre-shared-key cipher Huawei@1234
[Router_2-ike-peer-rut1] ike-proposal 1
[Router_2-ike-peer-rut1] aaa authorization domain rds
[Router_2-ike-peer-rut1] quit
```

Configure an IPsec proposal and an IPsec policy using the policy template.

```
[Router_2] ipsec proposal prop1
[Router_2-ipsec-proposal-prop1] undo esp authentication-algorithm
[Router_2-ipsec-proposal-prop1] undo esp encryption-algorithm
[Router_2-ipsec-proposal-prop1] quit
[Router_2] ipsec policy-template temp1 10
[Router_2-ipsec-policy-templet-temp1-10] ike-peer rut1
[Router_2-ipsec-policy-templet-temp1-10] proposal prop1
[Router_2-ipsec-policy-templet-temp1-10] quit
[Router_2] ipsec policy policy1 10 isakmp template temp1
```

Apply the IPsec policy group to the interface.

```
[Router_2] interface gigabitethernet 1/0/0
[Router_2-GigabitEthernet1/0/0] ipsec policy policy1
[Router_2-GigabitEthernet1/0/0] quit
```

Step 6 Configure Efficient VPN in Network-auto-cfg mode on Router_1 to establish an IPsec tunnel.

Set the Efficient VPN mode to Network-auto-cfg, and specify the remote address and pre-shared key for IKE negotiation in the Network-auto-cfg mode view.

```
[Router_1] ipsec efficient-vpn evpn mode network-auto-cfg
[Router_1-ipsec-efficient-vpn-evpn] remote-address 202.1.1.2 v1
[Router_1-ipsec-efficient-vpn-evpn] pre-shared-key cipher Huawei@1234
[Router_1-ipsec-efficient-vpn-evpn] sim-based-username type imsi password
huawei@123
[Router_1-ipsec-efficient-vpn-evpn] dh group14
[Router_1-ipsec-efficient-vpn-evpn] quit
```

Apply Efficient VPN to the interface.

```
[Router_1] interface cellular 0/0/0
[Router_1-Cellular0/0/0] ipsec efficient-vpn evpn
[Router_1-Cellular0/0/0] quit
```

Step 7 Verify the configuration.

After the configurations are complete, perform the **ping** operation on Router_1. Router_1 can still ping Router_2 successfully.

```
[Router_1] ping 202.1.1.2
PING 202.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 202.1.1.2: bytes=56 Sequence=1 ttl=255 time=4 ms
  Reply from 202.1.1.2: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 202.1.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
  Reply from 202.1.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 202.1.1.2: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 202.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/2/4 ms
```

Run the **display ipsec sa brief** command on Router_1 and Router_2 respectively to check whether an IPsec SA is successfully negotiated. The command output on Router_1 is used as an example.

```
[Router_1] display ipsec sa brief
```

Src address	Dst address	SPI	VPN	Protocol	Algorithm
202.1.10.1	202.1.1.2	2622706230	0	ESP	
202.1.10.1	202.1.1.2	3375760671	0	ESP	
202.1.1.2	202.1.10.1	645145990	0	ESP	
202.1.1.2	202.1.10.1	3429582856	0	ESP	

----End

Configuration Files

- Configuration file of Router_1

```
#
sysname Router_1
#
ipsec efficient-vpn evpn mode network-auto-cfg
remote-address 202.1.1.2 v1
pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
```

```
sim-based-username type imsi password %^
%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02F/%\*!%^%#
dh group14
#
interface Cellular0/0/0
dialer enable-circular
dialer-group 1
apn-profile lteprofile
dialer timer autodial 10
dialer number *99#
ipsec efficient-vpn evpn
ip address negotiate
mode lte auto
#
ip route-static 0.0.0.0 0.0.0.0 Cellular0/0/0
#
dialer-rule
dialer-rule 1 ip permit
#
apn profile lteprofile
apn LTENET
#
return
```

● Configuration file of Router_2

```
#
sysname Router_2
#
radius-server template shiva
radius-server shared-key cipher %%#R,T#A1Imu&K9;*-wF=/
2x{Ib*(^v><;=s*)mBup9%%#
radius-server authentication 192.168.10.1 1812 weight 80
radius-server accounting 192.168.10.1 1813 weight 80
#
ipsec proposal prop1
undo esp authentication-algorithm
undo esp encryption-algorithm
#
ike proposal 1
encryption-algorithm
aes-256
dh
group14
authentication-algorithm
sha2-256
authentication-method pre-
share
integrity-algorithm hmac-
sha2-256
prf hmac-sha2-256
#
ike peer rut1
undo version 2
exchange-mode aggressive
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
ike-proposal 1
aaa authorization domain rds
#
ipsec policy-template temp1 10
ike-peer rut1
proposal prop1
#
ipsec policy policy1 10 isakmp template temp1
#
aaa
authentication-scheme rds
authentication-mode radius
domain rds
authentication-scheme rds
radius-server shiva
```

```
#
interface GigabitEthernet1/0/0
ip address 202.1.1.2 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet2/0/0
ip address 192.168.2.1 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 202.1.1.1
#
return
```

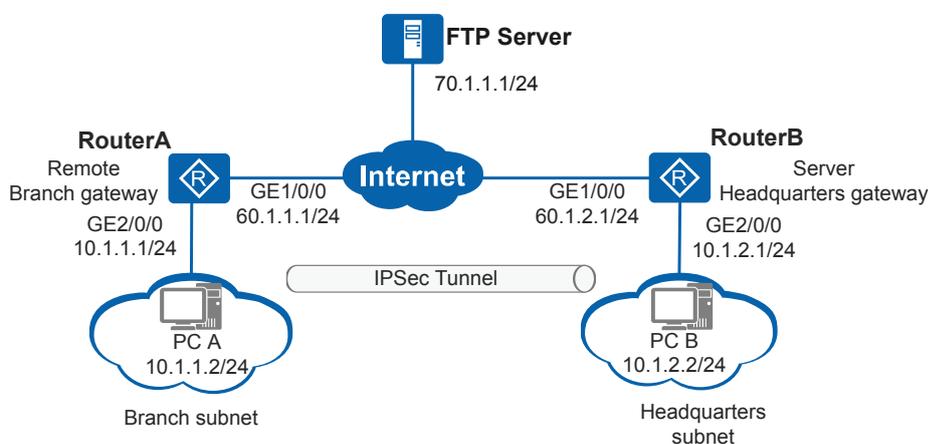
6.12.23 Example for Configuring Automatic Upgrade of the Efficient VPN Remote Device

Networking Requirements

As shown in **Figure 6-64**, RouterA (remote small-sized branch gateway) and RouterB (headquarters gateway) communicate through the Internet. The branch subnet is 10.1.1.0/24 and the headquarters subnet is 10.1.2.0/24. The branch gateway can download upgrade files from the FTP server. The branch gateway and headquarters gateway establish an IPsec tunnel to protect data flows using an Efficient VPN policy in client mode. Efficient VPN facilitates IPsec tunnel establishment and maintenance.

The enterprise requires that RouterA should obtain the FTP server IP address and network resources from RouterB through the IPsec tunnel to implement automatic upgrade so that the headquarters can manage the branch in centralized manner. Network deployment and maintenance can be also improved.

Figure 6-64 Configuring automatic upgrade of the Efficient VPN remote device



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses and static routes for interfaces on RouterA and RouterB so that routes between RouterA and RouterB are reachable.
2. Prepare the *.ini file on the FTP server. This file provides guidance for the upgrade of the branch gateway.

3. Configure an Efficient VPN policy in client mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.
4. Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

 **NOTICE**

After the device is upgraded, the original configuration is deleted. You can save the original configuration before the upgrade or write the required configuration in the configuration file for upgrade.

Procedure

- Step 1** Prepare the **huawei.ini** file on the FTP server. This file provides guidance for the upgrade of the branch gateway.

Format of the **huawei.ini** file:

```
MAC=5489-9874-  
ce3b;vrpfile=devicesoft_nocounter.cc;vrpver=V200R003C00B160;patchfile=patch.pat;cfgfile  
=cfg_1.cfg;restartflag=Y;location=nanjing;
```

The parameters are described as follows:

- **MAC**: MAC address of the branch device interface to which an Efficient VPN policy is applied. The branch device downloads the version file, patch file, and configuration file in which the MAC address is matched.
- **vrpfile**: path and file name of the version file. The file name extension of the version file is *.cc.
- **vrpver**: version number of the version file on the server.
- **patchfile**: path and file name of the patch file. The file name extension of the patch file is *.pat.
- **cfgfile**: path and file name of the configuration file on the server. The file name extension of the configuration file can be *.cfg or *.zip.
- **restartflag**: whether the device needs to restart. Y indicates that the device needs to restart. If hot patch files are used, the device does not need to restart. If cold patch files or files of other types are used, the device needs to restart; otherwise, the device upgrade fails.
- **location**: device location.

- Step 2** Configure IP addresses and static routes for interfaces on RouterA and RouterB.

Assign an IP address to an interface on RouterA.

```
<Huawei> system-view  
[Huawei] sysname RouterA  
[RouterA] interface gigabitethernet 1/0/0  
[RouterA-GigabitEthernet1/0/0] ip address 60.1.1.1 255.255.255.0  
[RouterA-GigabitEthernet1/0/0] quit  
[RouterA] interface gigabitethernet 2/0/0  
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0  
[RouterA-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterA. This example assumes that the next hop address in the route to RouterB is 60.1.1.2.

```
[RouterA] ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
```

Assign an IP address to an interface on RouterB.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[RouterB-GigabitEthernet2/0/0] quit
```

Configure a static route to the peer on RouterB. This example assumes that the next hop address in the route to RouterA is 60.1.2.2.

```
[RouterB] ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
```

Step 3 Configure an Efficient VPN policy in client mode on RouterA. RouterA as the initiator establishes an IPsec tunnel with RouterB.

Configure an Efficient VPN policy in client mode and specify the remote address and pre-shared key.

```
[RouterA] ipsec efficient-vpn evpn mode client
[RouterA-ipsec-efficient-vpn-evpn] remote-address 60.1.2.1 v1
[RouterA-ipsec-efficient-vpn-evpn] pre-shared-key cipher Huawei@1234
[RouterA-ipsec-efficient-vpn-evpn] dh group14
[RouterA-ipsec-efficient-vpn-evpn] quit
```

Apply the Efficient VPN policy to the interface.

```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipsec efficient-vpn evpn
[RouterA-GigabitEthernet1/0/0] quit
```

Step 4 Configure RouterB as the responder to use an IPsec policy template to establish an IPsec tunnel with RouterA.

Configure the URL and version number to be delivered in the service scheme view so that the IKE peer can reference them.

```
[RouterB] ip pool pooltest
[RouterB-ip-pool-pooltest] network 100.1.1.0 mask 255.255.255.0
[RouterB-ip-pool-pooltest] gateway-list 100.1.1.2
[RouterB-ip-pool-pooltest] quit
[RouterB] aaa
[RouterB-aaa] service-scheme schemetest
[RouterB-aaa-service-schemetest] ip-pool pooltest
[RouterB-aaa-service-schemetest] auto-update url ftp://
username:userpassword@70.1.1.1/huawei.ini version 1
[RouterB-aaa-service-schemetest] quit
[RouterB-aaa] quit
```

NOTE

The branch gateway determines whether to perform the upgrade according to the version number in the **auto-update url** command. The upgrade is performed only when the version to be delivered is later than the current version of the branch gateway.

Configure an IKE proposal and an IKE peer.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] encryption-algorithm aes-256
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut
[RouterB-ike-peer-rut] undo version 2
[RouterB-ike-peer-rut] exchange-mode aggressive
[RouterB-ike-peer-rut] pre-shared-key cipher Huawei@1234
[RouterB-ike-peer-rut] ike-proposal 5
[RouterB-ike-peer-rut] service-scheme schemetest
[RouterB-ike-peer-rut] quit
```

Configure an IPsec proposal and establish an IPsec policy using an IPsec policy template.

```
[RouterB] ipsec proposal prop1
[RouterB-ipsec-proposal-prop1] esp authentication-algorithm sha2-256
[RouterB-ipsec-proposal-prop1] esp encryption-algorithm aes-128
[RouterB-ipsec-proposal-prop1] quit
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut
[RouterB-ipsec-policy-templet-temp1-10] proposal prop1
[RouterB-ipsec-policy-templet-temp1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Apply the IPsec policy to an interface.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] ipsec policy policy1
[RouterB-GigabitEthernet1/0/0] quit
```

Step 5 Verify the configuration.

After the configurations are complete, PC A can ping PC B successfully. You can run the **display ipsec statistics** command to view packet statistics.

Run the **display ike sa** command on RouterA and RouterB to view the IKE SA configuration. The display on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
117 60.1.2.1:500 RD|ST V1:2 IP 60.1.2.1
116 60.1.2.1:500 RD|ST V1:1 IP 60.1.2.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

Run the **display ipsec efficient-vpn** command on RouterA to view information about the Efficient VPN policy. In the command output, **Auto-update url** and **Auto-update version** indicate that the URL and version number are delivered to RouterA.

```
[RouterA] display ipsec efficient-vpn
=====
IPsec efficient-vpn name: evpn
Using interface : GigabitEthernet1/0/0
=====
IPsec Efficient-vpn Name : evpn
IPsec Efficient-vpn Mode : 1 (1:Client 2:Network 3:Network-plus 4:Network-auto-cfg)
ACL Number :
Auth Method : 8 (8:PSK 9:RSA)
VPN name :
Local ID Type : 1 (1:IP 2:Name 3:User-fqdn 9:DN 11:Key-id)
```

```

IKE Version          : 1 (1:IKEv1 2:IKEv2)
Remote Address      : 60.1.2.1 (selected)
Pre Shared Key Cipher : ^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
PFS Type           : 0 (0:Disable 1:Group1 2:Group2 5:Group5 14:Group14
15:Group15 16:Group16)
Remote Name         :
PKI Object          :
Anti-replay window size : 32
Qos pre-classify    : 0 (0:Disable 1:Enable)
Qos group           : -
Service-scheme name :
DPD Msg Type        : seq-notify-hash
Sim-based-username Type :
Interface loopback   : LoopBack100
Interface loopback IP : 100.1.1.254/24
Dns server IP       :
Wins server IP      :
Dns default domain name :
Auto-update url     : ftp://70.1.1.1/huawei.ini
Auto-update version : 1
IP pool             :
  
```

Run the **display ipsec efficient-vpn remote** command on RouterB to view device running information including the device MAC address, version information, and whether the last upgrade is successful.

After an IPsec SA is established successfully, RouterA starts to download the version file, patch file, and configuration file. Then RouterA restarts. Run the **display startup** command to check whether the software version, configuration file, and patch file are target ones.

```

[RouterA] display startup
MainBoard:
Startup system software:          sd1:/devicesoft_nocounter.cc
Next startup system software:    sd1:/devicesoft_nocounter.cc
Backup system software for next startup: null
Startup saved-configuration file: sd1:/cfg_1.cfg
Next startup saved-configuration file: sd1:/cfg_1.cfg
Startup license file:            null
Next startup license file:       null
Startup patch package:           sd1:/patch.pat
Next startup patch package:      sd1:/patch.pat
Startup voice-files:             null
Next startup voice-files:        null
  
```

Run the **display patch-information** command on RouterA to view the patch file version.

---End

Configuration Files

- Configuration file of RouterA

```

#
sysname RouterA
#
ipsec efficient-vpn evpn mode client
remote-address 60.1.2.1 v1
pre-shared-key cipher ^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
dh group14
#
interface GigabitEthernet1/0/0
ip address 60.1.1.1 255.255.255.0
ipsec efficient-vpn evpn
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 60.1.2.0 255.255.255.0 60.1.1.2
  
```

```
ip route-static 10.1.2.0 255.255.255.0 60.1.1.2
ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal prop1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm
 aes-256
 dh
 group14
 authentication-algorithm
 sha2-256
 authentication-method pre-
 share
 integrity-algorithm hmac-
 sha2-256
 prf hmac-sha2-256
#
ike peer rut
 undo version 2
 exchange-mode aggressive
 pre-shared-key cipher %%K{JG:rWVHPMnf;5\|,GW(Luq!qi8BT4nOj%5W5=)%^%#
 ike-proposal 5
 service-scheme schemetest
#
ipsec policy-template temp1 10
 ike-peer rut
 proposal prop1
#
ipsec policy policy1 10 isakmp template temp1
#
ip pool pooltest
 gateway-list 100.1.1.2
 network 100.1.1.0 mask 255.255.255.0
#
aaa
 service-scheme schemetest
 ip-pool pooltest
 auto-update url ftp://username:%%##[HXY>~%M:~$uaIE@=Q.'gp=*B5]QO%zr>MIy+QuK
 %%##@70.1.1.1/huawei.ini version 1
#
interface GigabitEthernet1/0/0
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 60.1.1.0 255.255.255.0 60.1.2.2
ip route-static 10.1.1.0 255.255.255.0 60.1.2.2
#
return
```

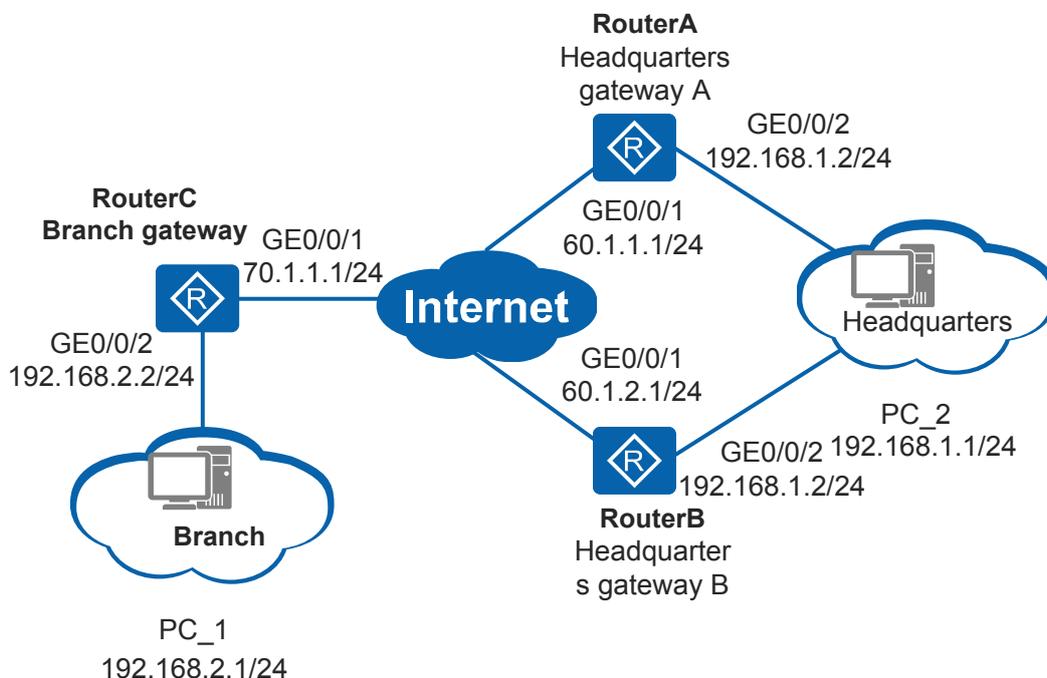
6.12.24 Example for Configuring Rapid Switchover and Revertive Switching

Networking Requirements

As shown in [Figure 6-65](#), the branch communicates with the headquarters over the public network. To improve reliability, the headquarters uses two gateways RouterA and RouterB to connect to the branch gateway RouterC.

The enterprise wants to protect traffic exchanged between the headquarters and branch and has the following requirements: Normally, the branch should communicate with the headquarters through RouterA. Traffic should be switched to RouterB when RouterA becomes faulty but back to RouterA when RouterA recovers.

Figure 6-65 Networking diagram for configuring rapid switchover and revertive switching



Configuration Roadmap

Since the branch and headquarters communicate over the public network, you can set up an IPsec tunnel between them to provide security protection. The configuration roadmap is as follows:

1. Configure the IP address on each interface and static routes to the peer to implement communication between interfaces.
2. Configure an NQA test instance to monitor the link between the branch gateway and headquarters gateway A.
3. Configure ACLs to define the data flows to be protected by the IPsec tunnel.
4. Configure IPsec proposals to define the traffic protection methods.

5. Create IKE peers and configure the device to determine the validity of the peer address according to the NQA test instance status, so that traffic can be rapidly switched from gateway A to gateway B when gateway A fails. Enable revertive switching of the IKE peer to ensure that traffic can be switched back to gateway A when gateway A recovers.
6. Configure IPsec security policies to define the data protection methods.
7. Apply the IPsec policies to interfaces so that the interfaces can protect traffic.

Procedure

Step 1 Configure an IP address for each interface and static routes to the peer on RouterA, RouterB, and RouterC to ensure that there are reachable routes among them.

Configure an IP address for each interface and static routes to the peer on RouterA. This example assumes that the next hop address in the route to the branch gateway is 60.1.1.2.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
[RouterA] ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 192.168.2.0 255.255.255.0 60.1.1.2
```

Configure an IP address for each interface and static routes to the peer on RouterB. This example assumes that the next hop address in the route to the branch gateway is 60.1.2.2.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet0/0/1] quit
[RouterB] interface gigabitethernet 0/0/2
[RouterB-GigabitEthernet0/0/2] ip address 192.168.1.2 255.255.255.0
[RouterB-GigabitEthernet0/0/2] quit
[RouterB] ip route-static 70.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 192.168.2.0 255.255.255.0 60.1.2.2
```

Configure an IP address for each interface and a static route to the peer on RouterC. This example assumes that the next hop addresses in the route to the headquarters gateways A and B are both 70.1.1.2.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ip address 70.1.1.1 255.255.255.0
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ip address 192.168.2.2 255.255.255.0
[RouterC-GigabitEthernet0/0/2] quit
[RouterC] ip route-static 0.0.0.0 0.0.0.0 70.1.1.2
```

Step 2 Configure an NQA test instance on RouterC.

Configure an NQA test instance of ICMP type (administrator name **admin** and instance name **test**) on RouterC to detect faults on the link 70.1.1.1/24 -> 60.1.1.1/24.

```
[RouterC] nqa test-instance admin test
[RouterC-nqa-admin-test] test-type icmp
[RouterC-nqa-admin-test] destination-address ipv4 60.1.1.1
[RouterC-nqa-admin-test] frequency 10
[RouterC-nqa-admin-test] probe-count 2
```

```
[RouterC-nqa-admin-test] start now
[RouterC-nqa-admin-test] quit
```

Step 3 Configure an ACL on RouterA, RouterB, and RouterC respectively to define the data flows to be protected.

Configure an ACL on RouterA to define the data flows from subnet 192.168.1.0/24 to subnet 192.168.2.0/24. The configuration of RouterB is similar to that of RouterA, and is not provided here.

```
[RouterA] acl number 3002
[RouterA-acl-adv-3002] rule permit ip source 192.168.1.0 0.0.0.255 destination
192.168.2.0 0.0.0.255
[RouterA-acl-adv-3002] quit
```

Configure an ACL on RouterC to define the data flows from subnet 192.168.2.0/24 to subnet 192.168.1.0/24.

```
[RouterC] acl number 3002
[RouterC-acl-adv-3002] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.1.0 0.0.0.255
[RouterC-acl-adv-3002] quit
```

Step 4 Create an IPsec proposal on RouterA, RouterB, and RouterC respectively.

Create an IPsec proposal on RouterA. The configurations of RouterB and RouterC are similar to that of RouterA, and are not provided here.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Step 5 Configure an IKE proposal and an IKE peer on RouterA, RouterB, and RouterC respectively.

Configure an IKE proposal and an IKE peer on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterA-ike-peer-rut1] quit
```

Configure an IKE proposal and an IKE peer on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterB-ike-peer-rut1] quit
```

Configure an IKE proposal and IKE peer **rut1** on RouterC, and set the address 60.1.1.1 to take effect when the status of the NQA test instance is Up and the address 60.1.2.1 to take effect when the status of the NQA test instance is Down.

```
[RouterC] ike proposal 5
[RouterC-ike-proposal-5] encryption-algorithm aes-128
[RouterC-ike-proposal-5] authentication-algorithm sha2-256
[RouterC-ike-proposal-5] dh group14
```

```
[RouterC-ike-proposal-5] quit
[RouterC] ike peer rut1
[RouterC-ike-peer-rut1] ike-proposal 5
[RouterC-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterC-ike-peer-rut1] remote-address 60.1.1.1 track nqa admin test up
[RouterC-ike-peer-rut1] remote-address 60.1.2.1 track nqa admin test down
[RouterC-ike-peer-rut1] switch-back enable
[RouterC-ike-peer-rut1] quit
```

Step 6 Configure an IPsec policy on RouterA, RouterB, and RouterC respectively.

Configure an IPsec policy using an IPsec policy template on RouterA.

```
[RouterA] ipsec policy-template temp1 10
[RouterA-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterA-ipsec-policy-templet-temp1-10] proposal tran1
[RouterA-ipsec-policy-templet-temp1-10] quit
[RouterA] ipsec policy policy1 10 isakmp template temp1
```

Configure an IPsec policy using an IPsec policy template on RouterB.

```
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterB-ipsec-policy-templet-temp1-10] proposal tran1
[RouterB-ipsec-policy-templet-temp1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Create an IPsec policy in ISAKMP mode on RouterC.

```
[RouterC] ipsec policy policy1 10 isakmp
[RouterC-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterC-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterC-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterC-ipsec-policy-isakmp-policy1-10] quit
```

Step 7 Apply the IPsec policies to the corresponding interfaces on RouterA, RouterB, and RouterC to make the interfaces able to protect traffic.

Apply the IPsec policy to the interface of RouterA.

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy to the interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy to the interface of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
```

Step 8 Verify the configuration.

After completing the configuration:

1. PC_1 can ping PC_2 successfully and data transmitted between them is encrypted.

Run the **display ike sa** command on RouterA and RouterB to view the IKE configuration. The command output on RouterA is used as an example.

```
[RouterA] display ike sa
IKE SA information :
  Conn-ID  Peer           VPN  Flag(s)  Phase  RemoteType  RemoteID
-----
  24366    70.1.1.1:500      RD   v2:2     IP     IP           70.1.1.1
```

```

24274 70.1.1.1:500 RD v2:1 IP 70.1.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
  
```

Run the **display ike sa** command on RouterC. The command output is as follows:

```

[RouterC] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
937 60.1.1.1:500 RD|ST v2:2 IP 60.1.1.1
936 60.1.1.1:500 RD|ST v2:1 IP 60.1.1.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
  
```

The command output shows that an IKE SA is successfully established between the branch gateway and headquarters gateway A.

2. Disconnect the link from the Internet to the headquarters gateway A. The IP address of the IKE peer changes to RouterB.

After you disconnect the link from the Internet to the headquarters gateway A, run the **display nqa results test-instance admin test** command on RouterC. The command output is as follows:

```

[RouterC] display nqa results test-instance admin
test

NQA entry(admin, test) :testflag is active ,testtype is
icmp
1 . Test 26 result The test is
finished
Send operation times: 2 Receive response times:
0 Completion:failed RTD OverThresholds number:
0 Attempts number:1 Drop operation number:
0 Disconnect operation number:0 Operation timeout number:
2 System busy operation number:0 Connection fail number:
0 Operation sequence errors number:0 RTT Status errors number:
0 Destination ip address:
60.1.1.1
Min/Max/Average Completion Time:
0/0/0
Sum/Square-Sum Completion Time:
0/0
Last Good Probe Time: 0000-00-00
00:00:00.0
Lost packet ratio: 100 %
.....
  
```

The command output shows that the status of gateway A is **failed** in NQA test results, and the status of the NQA test instance is Down.

Run the **display ike sa** command on RouterC. The command output is as follows:

```
[RouterC] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
21576 60.1.2.1:500 RD v2:2 IP 60.1.2.1
21575 60.1.2.1:500 RD v2:1 IP 60.1.2.1

Number of IKE SA : 2
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

The command output shows that the IKE peer address is **60.1.2.1**, indicating that traffic is switched to gateway B.

3. Recover the link from the Internet to the headquarters gateway A. The IP address of the IKE peer changes to RouterA.

After you recover the link from the Internet to the headquarters gateway A, run the **display nqa results test-instance admin test** command on RouterC. The command output is as follows:

```
[RouterC] display nqa results test-instance admin test

NQA entry(admin1, test) :testflag is active ,testtype is icmp
1 . Test 17 result The test is finished
Send operation times: 2 Receive response times:
2 Completion: success RTD OverThresholds number:
0 Attempts number:1 Drop operation number:
0 Disconnect operation number:0 Operation timeout number:
0 System busy operation number:0 Connection fail number:
0 Operation sequence errors number:0 RTT Status errors number:
0
Destination ip address:
60.1.1.1
Min/Max/Average Completion Time:
3/4/3
Sum/Square-Sum Completion Time:
7/25
Last Good Probe Time: 2014-09-26
16:38:07.3
Lost packet ratio:
0 %
.....
```

The command output shows that the status of gateway A is **success** in NQA test results, and the status of the NQA test instance is Up.

Run the **display ike sa** command on RouterC. The command output is as follows:

```
[RouterC] display ike sa
IKE SA information :
Conn-ID Peer VPN Flag(s) Phase RemoteType RemoteID
-----
21578 60.1.1.1:500 RD|ST v2:2 IP 60.1.1.1
21577 60.1.1.1:500 RD|ST v2:1 IP 60.1.1.1

Number of IKE SA : 2
-----
```

```
Flag Description:
RD--READY   ST--STAYALIVE   RL--REPLACED   FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT   LKG--LAST KNOWN GOOD SEQ NO.   BCK--BACKED UP
M--ACTIVE   S--STANDBY   A--ALONE   NEG--NEGOTIATING
```

The command output shows that the IKE peer address is **60.1.1.1**, indicating that traffic is switched back to gateway A. Rapid switchover and revertive switching are successfully configured.

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
acl number 3002
rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.2.0
0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
ike proposal 5
encryption-algorithm aes-128
dh group14
authentication-algorithm sha2-256
authentication-method pre-share
integrity-algorithm hmac-sha2-256
prf hmac-sha2-256
#
ike peer rut1
pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wnU] (q"Y2O&b'L=,NI`-qWE%%##
ike-proposal 5
#
ipsec policy-template temp1 10
ike-peer rut1
proposal tran1
#
ipsec policy policy1 10 isakmp template temp1
#
interface GigabitEthernet0/0/1
ip address 60.1.1.1 255.255.255.0
ipsec policy policy1
#
interface GigabitEthernet0/0/2
ip address 192.168.1.2 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
ip route-static 192.168.2.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
acl number 3002
rule 5 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.2.0
0.0.0.255
#
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
#
```

```
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wwnU] (q"Y2O&b'L=,NI`-qWE%###
 ike-proposal 5
#
ipsec policy-template templ 10
 ike-peer rut1
 proposal tran1
#
ipsec policy policy1 10 isakmp template templ
#
interface GigabitEthernet0/0/1
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.1.2 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.2.2
ip route-static 192.168.2.0 255.255.255.0 60.1.2.2
#
return
```

● Configuration file of RouterC

```
#
sysname RouterC
#
acl number 3002
 rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.1.0
 0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wwnU] (q"Y2O&b'L=,NI`-qWE%###
 ike-proposal 5
 remote-address 60.1.1.1 track nqa admin test up
 remote-address 60.1.2.1 track nqa admin test down
 switch-back enable
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
#
interface GigabitEthernet0/0/1
 ip address 70.1.1.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.2.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 70.1.1.2
```

```
#
nqa test-instance admin
test
  test-type
  icmp
  destination-address ipv4
  60.1.1.1
  frequency
  10
  probe-count
  2
  start now
#
return
```

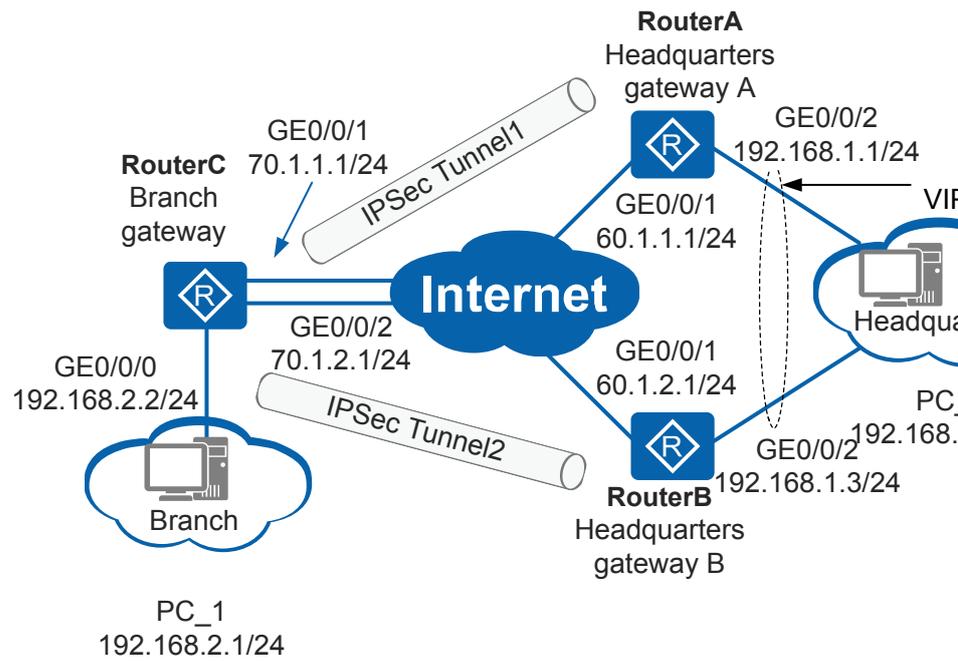
6.12.25 Example for Configuring Redundancy Control of IPsec Tunnels

Networking Requirements

As shown in [Figure 6-66](#), the branch communicates with the headquarters over the public network. To improve reliability, the headquarters uses two gateways RouterA and RouterB to connect to the branch gateway RouterC. RouterC sets up IPsec Tunnel1 with RouterA through GE0/0/1 and IPsec Tunnel2 with RouterB through GE0/0/2.

The enterprise wants to protect traffic exchanged between the headquarters and branch and requires that traffic be switched to the other IPsec tunnel when one IPsec tunnel fails and back to the faulty IPsec tunnel when the faulty IPsec tunnel recovers.

Figure 6-66 Networking diagram for configuring redundancy control of IPsec tunnels



Configuration Roadmap

Since the branch and headquarters communicate over the public network, you can set up an IPsec tunnel between them to provide security protection. The configuration roadmap is as follows:

1. Configure the IP address on each interface and static routes to the peer to implement communication between interfaces.
2. Configure an NQA group and an NQA test instance to monitor the link between the branch gateway and headquarters gateway A.
3. Configure ACLs to define the data flows to be protected by the IPsec tunnel.
4. Configure IPsec proposals to define the traffic protection methods.
5. Configure IKE peers.
6. Configure IPsec security policies to define the data protection methods. Configure the device to control IPsec tunnel setup and teardown according to the NQA group status and enable the device to switch traffic to the other IPsec tunnel when one IPsec tunnel becomes faulty.
7. Apply the IPsec policies to interfaces so that the interfaces can protect traffic.

NOTE

VRRP backup is configured on the two gateways in the headquarters. For detailed configuration, see VRRP Configuration.

Procedure

- Step 1** Configure an IP address for each interface and static routes to the peer on RouterA, RouterB, and RouterC to ensure that there are reachable routes among them.

Configure an IP address for each interface and static routes to the peer on RouterA. This example assumes that the next hop address in the route to the branch gateway is 60.1.1.2.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 60.1.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 0/0/2
[RouterA-GigabitEthernet0/0/2] ip address 192.168.1.1 255.255.255.0
[RouterA-GigabitEthernet0/0/2] quit
[RouterA] ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 70.1.2.0 255.255.255.0 60.1.1.2
[RouterA] ip route-static 192.168.2.0 255.255.255.0 60.1.1.2
```

Configure an IP address for each interface and static routes to the peer on RouterB. This example assumes that the next hop address in the route to the branch gateway is 60.1.2.2.

```
<Huawei> system-view
[Huawei] sysname RouterB
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ip address 60.1.2.1 255.255.255.0
[RouterB-GigabitEthernet0/0/1] quit
[RouterB] interface gigabitethernet 0/0/2
[RouterB-GigabitEthernet0/0/2] ip address 192.168.1.3 255.255.255.0
[RouterB-GigabitEthernet0/0/2] quit
[RouterB] ip route-static 70.1.1.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 70.1.2.0 255.255.255.0 60.1.2.2
[RouterB] ip route-static 192.168.2.0 255.255.255.0 60.1.2.2
```

Configure an IP address for each interface and static routes to the peer on RouterC. This example assumes that the next hop addresses in the route to the headquarters gateways A and B are 70.1.1.2 and 70.1.2.2, respectively.

```
<Huawei> system-view
[Huawei] sysname RouterC
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ip address 70.1.1.1 255.255.255.0
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ip address 70.1.2.1 255.255.255.0
[RouterC-GigabitEthernet0/0/2] quit
[RouterC] interface gigabitethernet 0/0/0
[RouterC-GigabitEthernet0/0/0] ip address 192.168.2.2 255.255.255.0
[RouterC-GigabitEthernet0/0/0] quit
[RouterC] ip route-static 60.1.1.0 255.255.255.0 70.1.1.2
[RouterC] ip route-static 60.1.2.0 255.255.255.0 70.1.2.2
[RouterC] ip route-static 192.168.1.0 255.255.255.0 70.1.1.2
[RouterC] ip route-static 192.168.1.0 255.255.255.0 70.1.2.2
```

Step 2 Configure an NQA test instance on RouterC.

Configure an NQA test instance of ICMP type (administrator name **admin** and instance name **test**) on RouterC to detect faults on the link 70.1.1.1/24 -> 60.1.1.1/24.

```
[RouterC] nqa test-instance admin test
[RouterC-nqa-admin-test] test-type icmp
[RouterC-nqa-admin-test] destination-address ipv4 60.1.1.1
[RouterC-nqa-admin-test] frequency 10
[RouterC-nqa-admin-test] probe-count 2
[RouterC-nqa-admin-test] start now
[RouterC-nqa-admin-test] quit
```

Step 3 Configure an ACL on RouterC to define the data flows to be protected.

NOTE

An IPsec policy is created on RouterA and RouterB using the IPsec policy template; therefore, this step is optional. If you configure an ACL on RouterA and RouterB, you must specify the destination address in the ACL rule.

Configure an ACL on RouterC to define the data flows from subnet 192.168.2.0/24 to subnet 192.168.1.0/24.

```
[RouterC] acl number 3002
[RouterC-acl-adv-3002] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.1.0 0.0.0.255
[RouterC-acl-adv-3002] quit
```

Step 4 Create an IPsec proposal on RouterA, RouterB, and RouterC respectively.

Create an IPsec proposal on RouterA. The configurations of RouterB and RouterC are similar to that of RouterA, and are not provided here.

```
[RouterA] ipsec proposal tran1
[RouterA-ipsec-proposal-tran1] esp authentication-algorithm sha2-256
[RouterA-ipsec-proposal-tran1] esp encryption-algorithm aes-128
[RouterA-ipsec-proposal-tran1] quit
```

Step 5 Configure an IKE proposal and an IKE peer on RouterA, RouterB, and RouterC respectively.

NOTE

RouterA and RouterB function as responders to respond to an IKE negotiation request; therefore, IPsec policies are created on them through IPsec policy templates. You do not need to set **remote-address**.

Configure an IKE proposal and an IKE peer on RouterA.

```
[RouterA] ike proposal 5
[RouterA-ike-proposal-5] encryption-algorithm aes-128
[RouterA-ike-proposal-5] authentication-algorithm sha2-256
[RouterA-ike-proposal-5] dh group14
[RouterA-ike-proposal-5] quit
[RouterA] ike peer rut1
[RouterA-ike-peer-rut1] undo version 2
[RouterA-ike-peer-rut1] ike-proposal 5
[RouterA-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterA-ike-peer-rut1] quit
```

Configure an IKE proposal and an IKE peer on RouterB.

```
[RouterB] ike proposal 5
[RouterB-ike-proposal-5] encryption-algorithm aes-128
[RouterB-ike-proposal-5] authentication-algorithm sha2-256
[RouterB-ike-proposal-5] dh group14
[RouterB-ike-proposal-5] quit
[RouterB] ike peer rut1
[RouterB-ike-peer-rut1] undo version 2
[RouterB-ike-peer-rut1] ike-proposal 5
[RouterB-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterB-ike-peer-rut1] quit
```

Configure an IKE proposal and IKE peer rut1 and rut2 on RouterC.

```
[RouterC] ike proposal 5
[RouterC-ike-proposal-5] encryption-algorithm aes-128
[RouterC-ike-proposal-5] authentication-algorithm sha2-256
[RouterC-ike-proposal-5] dh group14
[RouterC-ike-proposal-5] quit
[RouterC] ike peer rut1
[RouterC-ike-peer-rut1] undo version 2
[RouterC-ike-peer-rut1] ike-proposal 5
[RouterC-ike-peer-rut1] pre-shared-key cipher Huawei@123
[RouterC-ike-peer-rut1] remote-address 60.1.1.1
[RouterC-ike-peer-rut1] quit
[RouterC] ike peer rut2
[RouterC-ike-peer-rut2] undo version 2
[RouterC-ike-peer-rut2] ike-proposal 5
[RouterC-ike-peer-rut2] pre-shared-key cipher Huawei@123
[RouterC-ike-peer-rut2] remote-address 60.1.2.1
[RouterC-ike-peer-rut2] quit
```

Step 6 Create an IPsec policy on RouterA, RouterB, and RouterC respectively.

Create an IPsec policy through an IPsec policy template on RouterA.

```
[RouterA] ipsec policy-template temp1 10
[RouterA-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterA-ipsec-policy-templet-temp1-10] proposal tran1
[RouterA-ipsec-policy-templet-temp1-10] quit
[RouterA] ipsec policy policy1 10 isakmp template temp1
```

Create an IPsec policy through an IPsec policy template on RouterB.

```
[RouterB] ipsec policy-template temp1 10
[RouterB-ipsec-policy-templet-temp1-10] ike-peer rut1
[RouterB-ipsec-policy-templet-temp1-10] proposal tran1
[RouterB-ipsec-policy-templet-temp1-10] quit
[RouterB] ipsec policy policy1 10 isakmp template temp1
```

Create IPsec policies policy1 and policy2 in ISAKMP mode on RouterC.

```
[RouterC] ipsec policy policy1 10 isakmp
[RouterC-ipsec-policy-isakmp-policy1-10] ike-peer rut1
[RouterC-ipsec-policy-isakmp-policy1-10] proposal tran1
[RouterC-ipsec-policy-isakmp-policy1-10] security acl 3002
[RouterC-ipsec-policy-isakmp-policy1-10] connect track nqa admin test up
[RouterC-ipsec-policy-isakmp-policy1-10] disconnect track nqa admin test down
```

```
[RouterC-ipsec-policy-isakmp-policy1-10] quit
[RouterC] ipsec policy policy2 20 isakmp
[RouterC-ipsec-policy-isakmp-policy2-20] ike-peer rut2
[RouterC-ipsec-policy-isakmp-policy2-20] proposal tran1
[RouterC-ipsec-policy-isakmp-policy2-20] security acl 3002
[RouterC-ipsec-policy-isakmp-policy2-20] connect track nqa admin test down
[RouterC-ipsec-policy-isakmp-policy2-20] disconnect track nqa admin test up
[RouterC-ipsec-policy-isakmp-policy2-20] quit
```

Step 7 Apply the IPsec policies to the corresponding interfaces on RouterA, RouterB, and RouterC to make the interfaces able to protect traffic.

Apply the IPsec policy to the interface of RouterA.

```
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ipsec policy policy1
[RouterA-GigabitEthernet0/0/1] quit
```

Apply the IPsec policy to the interface of RouterB.

```
[RouterB] interface gigabitethernet 0/0/1
[RouterB-GigabitEthernet0/0/1] ipsec policy policy1
[RouterB-GigabitEthernet0/0/1] quit
```

Apply the IPsec policies to the interfaces of RouterC.

```
[RouterC] interface gigabitethernet 0/0/1
[RouterC-GigabitEthernet0/0/1] ipsec policy policy1
[RouterC-GigabitEthernet0/0/1] quit
[RouterC] interface gigabitethernet 0/0/2
[RouterC-GigabitEthernet0/0/2] ipsec policy policy2
[RouterC-GigabitEthernet0/0/2] quit
```

Step 8 Verify the configuration.

After completing the configuration:

1. PC_1 can ping PC_2 successfully and data transmitted between them is encrypted.

Run the **display ipsec sa** command on RouterC to check the IPsec configuration.

```
[RouterC] display ipsec sa
```

```
=====
Interface:
GigabitEthernet0/0/1
  Path MTU:
1500
=====
```

```
-----
IPSec policy name:
"policy1"
  Sequence number  :
10
  Acl group        :
3002
  Acl rule         :
5
  Mode             :
ISAKMP
```

```
-----
Connection ID    :
21812
```

```

Encapsulation mode:
Tunnel
  Tunnel local      :
70.1.1.1
  Tunnel remote    :
60.1.1.1
  Flow source      : 192.168.2.0/255.255.255.0
0/0
  Flow destination : 192.168.1.0/255.255.255.0
0/0
  Qos pre-classify :
Disable
  Qos group        :
-

[Outbound ESP
SAs]
  SPI: 870098030
(0x33dca46e)
  Proposal: ESP-ENCRYPT-AES-128
SHA2-256-128
  SA remaining key duration (bytes/sec):
1887436800/3395
  Max sent sequence-number:
0
  UDP encapsulation used for NAT traversal:
N

[Inbound ESP
SAs]
  SPI: 2558349639
(0x987d5147)
  Proposal: ESP-ENCRYPT-AES-128
SHA2-256-128
  SA remaining key duration (bytes/sec):
1887436800/3395
  Max received sequence-number:
0
  Anti-replay window size:
32
  UDP encapsulation used for NAT traversal: N
  
```

The command output shows that traffic from PC_1 to PC_2 is transmitted over IPsec Tunnel1 (source IP address: 70.1.1.1, destination IP address: 60.1.1.1).

2. Disable GE0/0/1 of RouterC. Traffic is switched to IPsec Tunnel2 (source IP address: 70.1.2.1/24, destination IP address: 60.1.2.1/24).

Run the **shutdown** command on GE0/0/1 of RouterC, and then run the **display nqa results test-instance admin test** command. The command output is as follows:

```

[RouterC] display nqa results test-instance admin test

NQA entry(admin, test) :testflag is active ,testtype is
icmp
 1 . Test 46392 result   The test is
finished
  Send operation times: 2           Receive response times:
0
  Completion:failed      RTD OverThresholds number:
0
  Attempts number:1      Drop operation number:
2
  Disconnect operation number:0    Operation timeout number:
0
  System busy operation number:0    Connection fail number:
0
  Operation sequence errors number:0 RTT Status errors number:
  
```

```
0
  Destination ip address:
60.1.1.1
  Min/Max/Average Completion Time:
0/0/0
  Sum/Square-Sum Completion Time:
0/0
  Last Good Probe Time: 0000-00-00
00:00:00.0
  Lost packet ratio: 100 %
  .....
```

The command output shows that the NQA test result is **failed**, indicating that the status of the NQA test instance is Down.

Run the **display ipsec sa** command on RouterC to check the IPsec configuration.

```
[RouterC] display ipsec sa

=====

Interface:
GigabitEthernet0/0/2
  Path MTU:
1500
=====

-----
  IPsec policy name:
"policy2"
  Sequence number  :
20
  Acl group        :
3002
  Acl rule         :
5
  Mode             :
ISAKMP

-----

  Connection ID    :
21839
  Encapsulation mode:
Tunnel
  Tunnel local     :
70.1.2.1
  Tunnel remote    :
60.1.2.1
  Flow source      : 192.168.2.0/255.255.255.0
0/0
  Flow destination : 192.168.1.0/255.255.255.0
0/0
  Qos pre-classify :
Disable
  Qos group        :
-

  [Outbound ESP
SAs]
  SPI: 437762941
(0x1a17bb7d)
  Proposal: ESP-ENCRYPT-AES-128
SHA2-256-128
  SA remaining key duration (bytes/sec):
1887436800/3575
  Max sent sequence-number:
```

```

0
    UDP encapsulation used for NAT traversal:
N

    [Inbound ESP
SAs]
    SPI: 1765690761
    (0x693e4d89)
    Proposal: ESP-ENCRYPT-AES-128
    SHA2-256-128
    SA remaining key duration (bytes/sec):
1887436800/3575
    Max received sequence-number:
0
    Anti-replay window size:
32
    UDP encapsulation used for NAT traversal: N
  
```

The command output shows that traffic is switched to IPsec Tunnel2 (source IP address: 70.1.2.1, destination IP address: 60.1.2.1).

3. Enable GE0/0/1 of RouterC again. Traffic is switched back to IPsec Tunnel1 (source IP address: 70.1.1.1, destination IP address: 60.1.1.1).

Run the **undo shutdown** command on GE0/0/1 of RouterC, and then run the **display nqa results test-instance admin test** command. The command output is as follows:

```

[RouterC] display nqa results test-instance admin test

NQA entry(admin, test) :testflag is active ,testtype is
icmp
 1 . Test 46694 result   The test is
finished
  Send operation times: 2           Receive response times:
2      Completion: success           RTD OverThresholds number:
0      Attempts number:1           Drop operation number:
0      Disconnect operation number:0   Operation timeout number:
0      System busy operation number:0   Connection fail number:
0      Operation sequence errors number:0   RTT Status errors number:
0
  Destination ip address:
60.1.1.1
  Min/Max/Average Completion Time:
4/4/4
  Sum/Square-Sum Completion Time:
8/32
  Last Good Probe Time: 2014-09-29
20:43:23.2
  Lost packet ratio: 0 %
  .....
  
```

The command output shows that the NQA detection result is **success**, indicating that the status of the NQA test instance is Up.

Run the **display ipsec sa** command on RouterC to check the IPsec configuration.

```

[RouterC] display ipsec sa

=====

Interface:
GigabitEthernet0/0/1
  Path MTU:
1500
  
```

```
=====
-----
IPSec policy name:
"policy1"
Sequence number  :
10
Acl group       :
3002
Acl rule        :
5
Mode            :
ISAKMP
-----
Connection ID    :
21992
Encapsulation mode:
Tunnel
Tunnel local     :
70.1.1.1
Tunnel remote    :
60.1.1.1
Flow source      : 192.168.2.0/255.255.255.0
0/0
Flow destination : 192.168.1.0/255.255.255.0
0/0
Qos pre-classify :
Disable
Qos group        :
-

[Outbound ESP
SAs]
SPI: 2749069243
(0xa3db77bb)
Proposal: ESP-ENCRYPT-AES-128
SHA2-256-128
SA remaining key duration (bytes/sec):
1887436800/3583
Max sent sequence-number:
0
UDP encapsulation used for NAT traversal:
N

[Inbound ESP
SAs]
SPI: 21830677
(0x14d1c15)
Proposal: ESP-ENCRYPT-AES-128
SHA2-256-128
SA remaining key duration (bytes/sec):
1887436800/3583
Max received sequence-number:
0
Anti-replay window size:
32
UDP encapsulation used for NAT traversal: N
```

The command output shows that traffic is switched back to IPsec Tunnel1 (source IP address: 70.1.1.1, destination IP address: 60.1.1.1). The configuration succeeds.

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 undo version 2
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wnnUJ (q"Y2O&b'L=,NI`-qWE%%##
 ike-proposal 5
#
ipsec policy-template temp1 10
 ike-peer rut1
 proposal tran1
#
ipsec policy policy1 10 isakmp template temp1
#
interface GigabitEthernet0/0/1
 ip address 60.1.1.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.1.1 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.1.2
ip route-static 70.1.2.0 255.255.255.0 60.1.1.2
ip route-static 192.168.2.0 255.255.255.0 60.1.1.2
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 undo version 2
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wnnUJ (q"Y2O&b'L=,NI`-qWE%%##
 ike-proposal 5
#
ipsec policy-template temp1 10
 ike-peer rut1
 proposal tran1
#
ipsec policy policy1 10 isakmp template temp1
#
```

```
interface GigabitEthernet0/0/1
 ip address 60.1.2.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 192.168.1.3 255.255.255.0
#
ip route-static 70.1.1.0 255.255.255.0 60.1.2.2
ip route-static 70.1.2.0 255.255.255.0 60.1.2.2
ip route-static 192.168.2.0 255.255.255.0 60.1.2.2
#
return
```

● Configuration file of RouterC

```
#
sysname RouterC
#
acl number 3002
 rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.1.0
0.0.0.255
#
ipsec proposal tran1
 esp authentication-algorithm sha2-256
 esp encryption-algorithm aes-128
#
ike proposal 5
 encryption-algorithm aes-128
 dh group14
 authentication-algorithm sha2-256
 authentication-method pre-share
 integrity-algorithm hmac-sha2-256
 prf hmac-sha2-256
#
ike peer rut1
 undo version 2
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wnU] (q"Y2O&b'L=,NI`-qWE%%##
 ike-proposal 5
 remote-address 60.1.1.1
#
ike peer rut2
 undo version 2
 pre-shared-key cipher %%%#u;3RGfy.^D2'oEC%wnU] (q"Y2O&b'L=,NI`-qWE%%##
 ike-proposal 5
 remote-address 60.1.2.1
#
ipsec policy policy1 10 isakmp
 security acl 3002
 ike-peer rut1
 proposal tran1
 connect track nqa admin test up
 disconnect track nqa admin test down
#
ipsec policy policy2 20 isakmp
 security acl 3002
 ike-peer rut2
 proposal tran1
 connect track nqa admin test down
 disconnect track nqa admin test up
#
interface GigabitEthernet0/0/0
 ip address 192.168.2.2 255.255.255.0
#
interface GigabitEthernet0/0/1
 ip address 70.1.1.1 255.255.255.0
 ipsec policy policy1
#
interface GigabitEthernet0/0/2
 ip address 70.1.2.1 255.255.255.0
 ipsec policy policy2
#
```

```
ip route-static 60.1.1.0 255.255.255.0 70.1.1.2
ip route-static 60.1.2.0 255.255.255.0 70.1.2.2
ip route-static 192.168.1.0 255.255.255.0 70.1.1.2
ip route-static 192.168.1.0 255.255.255.0 70.1.2.2
#
nqa test-instance admin test
 test-type
 icmp
  destination-address ipv4
 60.1.1.1
  frequency
 10
  probe-count
 2
  start now
#
nqa-group group1
 nqa admin test
#
return
```

6.13 Troubleshooting IPsec

6.13.1 IKE SA Negotiation Failed

Symptom

The IPsec service cannot be normally transmitted. The output of the **display ike sa** command shows that IKE SA negotiation failed.

The following shows an example of the command output. If the **Flag** parameter is displayed as **RD** or **RD|ST**, an SA is established successfully. **ST** indicates that the local end is the IKE initiator.

Conn-ID	Peer	VPN	Flag(s)	Phase
13118	10.1.3.2	0	RD	v1:2
12390	10.1.3.2	0	RD	v1:1

Number of IKE SA : 2

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING

If IKE SA negotiation fails, the **Flag** parameter is empty, the **Peer** parameter is 0.0.0.0, or the command output contains no record.

Procedure

- Step 1** Run the **display ike proposal** command to check whether the IKE peer uses the same IKE proposal.

If not, change IKE proposals on the peer to be the same. If the authentication algorithms in the IKE proposals are different, perform the following operations.

On the IKE initiator:

```
ike proposal 10
 authentication-algorithm sha2-256
```

On the IKE responder:

```
ike proposal 10
 authentication-algorithm sha2-384
```

Step 2 Run the **display ike peer** command to check whether the configuration in the peer view is correct.

- Check whether the remote IP address is configured.

When the ACL mode is used for IPsec tunnel establishment, the remote IP address must be specified for the device in IKE main negotiation mode. In addition, the remote IP addresses specified for the IKE peer must match each other.

If the IP addresses of the IKE initiator and responder are 10.1.1.2 and 10.2.1.2, the configuration is as follows.

On the IKE initiator:

```
ike peer mypeer1
 remote-address 10.2.1.2
```

On the IKE responder:

```
ike peer mypeer2
 remote-address 10.1.1.2
```

If the IKE responder uses the policy template mode, you do not need to configure the remote IP address for the responder.

- Check whether the pre-shared keys of the IKE peer are the same.

```
ike peer mypeer
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\!*%^%# //
The key is Huawei@123.
```

If not, change the pre-shared keys to be the same.

- Check whether the IKE proposals referenced by the IKE peer are the same.

For example, the IKE initiator references IKE proposal 10.

```
ike peer mypeer
 ike-proposal 10
```

The related configuration of IKE proposal 10 is as follows.

```
ike proposal 10
 encryption-algorithm aes-128
 authentication-algorithm sha2-256
```

If the configurations in the IKE proposals are different, change them to be the same.

----End

6.13.2 IPsec SA Negotiation Failed

Symptom

The IPsec service cannot be normally transmitted. The output of the **display ike sa** command shows that IPsec SA negotiation failed.

The following shows an example of the command output. If the **Flag** parameter is displayed as **RD** or **RD|ST**, an SA is established successfully. **ST** indicates that the local end is the IKE initiator.

Conn-ID	Peer	VPN	Flag(s)	Phase
-----	-----	-----	-----	-----

```
13118    10.1.3.2    0    RD    v1:2
12390    10.1.3.2    0    RD    v1:1

Number of IKE SA : 2
-----

Flag Description:
RD--READY    ST--STAYALIVE    RL--REPLACED    FD--FADING    TO--TIMEOUT
HRT--HEARTBEAT    LKG--LAST KNOWN GOOD SEQ NO.    BCK--BACKED UP
M--ACTIVE    S--STANDBY    A--ALONE    NEG--NEGOTIATING
```

If IKE SA negotiation is successful, but IPsec SA negotiation fails, the command output contains no information about phase 2 or the **Flag** parameter is empty.

Procedure

Step 1 Run the **display ipsec proposal** command to check whether the IKE peer uses the same IPsec proposal.

If not, change IPsec proposals on the peer to be the same. If the ESP authentication algorithms in the IPsec proposals are different, perform the following operations.

On the IKE initiator:

```
ipsec proposal prop1
 esp authentication-algorithm sha2-512
```

On the IKE responder:

```
ipsec proposal prop2
 esp authentication-algorithm sha2-384
```

Step 2 Run the **display ipsec policy** command to check whether the configuration in the IPsec policy view is correct.

- Check whether the ACLs referenced in the IPsec policies are the same.

If the ACLs referenced by IPsec policies at both ends of the IPsec tunnel mirror each other, an IPsec SA can be successfully established when either party initiates the negotiation. If the ACLs do not mirror each other, an IPsec SA can be established only when the IP address range defined in the ACL on the initiator is included in the IP address range defined in the ACL on the responder. Therefore, it is recommended that the ACLs at both ends of the IPsec tunnel mirror each other. That is, the source and destination addresses in the ACL at one end are the same as the destination and source addresses in the ACL at the other end.

For example, if the source and destination addresses of the initiator are 10.1.1.2 and 10.2.1.2, the source and destination addresses of the responder are 10.2.1.2 and 10.1.1.2.

On the IKE initiator:

```
acl number 3101
 rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.2.1.0 0.0.0.255
 ipsec policy map1 10 isakmp
 security acl 3101
```

On the IKE responder:

```
acl number 3101
 rule 5 permit ip source 10.2.1.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
 ipsec policy map2 10 isakmp
 security acl 3101
```

- Check whether the IKE peer configurations referenced in the IPsec policies are the same.

For example, the IKE initiator reference IKE proposal peer **spub**.

```
ipsec policy map1 10 isakmp
ike-peer spub
```

The related configuration of the IKE peer is as follows.

```
ike peer spub
undo version 2
pre-shared-key cipher %%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%# //
The key is Huawei@123.
ike-proposal 5
remote-address 202.138.162.1
```

If the IKE peer configurations at two ends of the tunnel are different, change them to be the same.

- Check whether the IPsec proposal configurations referenced in the IPsec policies are the same.

For example, the IKE initiator reference IPsec proposal **tran1**.

```
ipsec policy policy1 100 isakmp
proposal tran1
```

The related configuration of the IPsec proposal is as follows.

```
ipsec proposal tran1
esp authentication-algorithm sha2-256
esp encryption-algorithm aes-128
```

If the IPsec proposal configurations at two ends of the tunnel are different, change them to be the same.

----End

6.13.3 Services Are Interrupted After an IPsec Tunnel Is Established

Fault Symptom

If IPsec SA information is displayed in the **display ike sa** command output, an IPsec tunnel has been established successfully. However, the following problems exist:

- Users in the branches and the headquarters cannot communicate with each other.
- Only unidirectional communication is implemented between users of the branches and headquarters. For example, users of the headquarters can access servers in the branch, but users of the branches cannot access servers in the headquarters.
- Users of the branch and headquarters can access only some network segments.
- On a point-to-multipoint network, users of the branches may communicate with the headquarters normally, but users of different branches cannot communicate with each other.

Procedure

Step 1 Check whether security ACLs on both ends are correctly configured.

Run the **display ipsec sa** command to check whether the source and destination network segments of the protected data flows matching security ACLs include actual service flows. If not, run the **display acl acl-number** command to check whether ACLs are correctly configured on both ends.

For details about configuring IPsec-protected data flows, see [6.7.1 Defining Data Flows to Be Protected](#).

For example, run the **display ipsec sa** command to check the security ACL 3101.

```
<Huawei> system-view
[Huawei] display ipsec sa
ipsec sa information:
=====
Interface: GigabitEthernet1/0/0
=====
-----
IPSec policy name: "policy1"
Sequence number   : 1
Acl group         : 3101
Acl rule          : 5
Mode              : ISAKMP
-----
Connection ID     : 67108879
Encapsulation mode: Tunnel
Holding time      : 0d 0h 4m 29s
Tunnel local      : 1.1.1.1:500
Tunnel remote     : 2.1.1.1:500
Flow source       : 10.1.1.0/255.255.255.0 17/1701
Flow destination  : 10.2.1.0/255.255.255.0 17/39725
.....
```

If the protected data flows matching the security ACL do not include actual service flows, run the **display acl 3101** command to check the ACL configuration.

```
[Huawei] display acl 3101
Advanced ACL 3101, 1 rule
Acl's step is 5
 rule 5 permit ip source 10.1.1.0 0.0.0.255 destination 10.2.1.0 0.0.0.255
```

If the security ACLs on both ends do not match, modify the security ACL configurations to make them match.

Step 2 Check whether the NAT policy configuration affects the IPsec-protected data flows.

Run the **display ipsec interface brief** command to check the interface to which an IPsec policy is applied, and then run the **display current-configuration interface interface-type interface-number** command to check whether a NAT policy is configured on the specified IPsec interface.

If NAT is configured on the interface to which an IPsec policy is applied, IPsec does not take effect because the device performs NAT first. In this case, you need to ensure:

- The destination IP address denied in the ACL rule referenced by NAT is the destination IP address in the ACL rule referenced by IPsec. This prevents the device from performing NAT on the IPsec-protected data flows.
- The ACL rule referenced by IPsec matches the post-NAT IP address.

For example, the following command output indicates that a NAT policy is configured on GE1/0/0.

```
[Huawei] display current-configuration interface gigabitethernet 1/0/0
interface GigabitEthernet1/0/0
 ip address 1.1.1.1 255.255.255.0
 nat outbound 3000 //A NAT policy is configured for the interface.
 ipsec policy policy1
```

The ACL rule configuration referenced by IPsec is as follows:

```
acl number 3101
 rule 5 permit ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
```

The ACL rule configuration referenced by NAT is as follows:

```
acl number 3000
 rule 5 permit ip
```

You can modify the configuration using either of the following methods:

- If NAT needs to be performed on data flows to the peer need before IPsec encryption, change the source IP address in the ACL 3101 to the post-NAT IP address.

```
acl number 3101
 rule permit ip source 1.1.1.0 0.0.0.255 destination 2.1.1.0 0.0.0.255
```

- If NAT does not need to be performed on data flows to the peer need before IPsec encryption, add a deny policy to the ACL.

```
acl number 3000
 rule 1 deny ip source 10.1.2.0 0.0.0.255 destination 10.1.1.0 0.0.0.255
 rule 5 permit ip
```

Step 3 Check whether NAT traversal is enabled on both ends if a NAT device exists between both ends.

Run the **display ike peer** command to check whether NAT traversal is enabled on both ends. If not, run the **nat traversal** command in the IKE peer view.

```
[Huawei] display ike peer
Number of IKE peers: 1
-----
Peer name           : pa
IKE version        : v1v2
VPN instance       : -
Remote IP          : 2.1.1.1
Authentic IP address : -
Proposal          : tran1
Pre-shared-key     : %^%#G7 (t:%yFw/PVF>Jsva;"zx]oL!sw-8z\C;I}%%RY
%^%#
Local ID type      :
IP
Local ID           : -
Remote ID type     : -
Remote ID         : -
certificate peer-name : -
PKI realm         : -
.....
NAT-traversal     : Disable
.....
[Huawei] ike peer pa
[Huawei-ike-peer-pa] nat traversal
```

Step 4 Check whether the security protocol is AH when a NAT device exists between both ends and NAT traversal is enabled.

Run the **display ipsec proposal brief** command to check the security protocol. The security protocol can only be ESP during NAT traversal.

If the security protocol is AH, run the **transform** command to change the security protocol to ESP.

```
[Huawei] display ipsec proposal brief
Current ipsec proposal number: 1
-----
Proposal Name      Encapsulation mode  Transform
-----
tran1             Tunnel              ah-new
[Huawei] ipsec proposal tran1
[Huawei-ipsec-proposal-tran1] transform esp
```

Step 5 Check whether the encryption/decryption modes on both ends are consistent if the authentication algorithm used in an IPsec proposal is SHA2.

Run the **display ipsec proposal** command to check whether the authentication algorithm is SHA2-256, SHA2-384, or SHA2-512.

```
[Huawei] display ipsec proposal
Number of proposals: 1

IPsec proposal name: 1
Encapsulation mode: Tunnel
Transform           : esp-new
ESP protocol        : Authentication SHA2-HMAC-256 //Authentication algorithm
                    : Encryption AES-256
```

When IPsec uses the SHA-2 algorithm, if the devices on both ends of an IPsec tunnel are from different vendors or run different software versions, they may use different encryption/decryption modes. In this situation, IPsec traffic between the devices will be interrupted.

To solve the problem, run the **ipsec authentication sha2 compatible enable** command in the system view to enable the SHA-2 algorithm to be compatible with earlier versions.

```
[Huawei] ipsec authentication sha2 compatible enable
```

----End

6.14 FAQ About IPsec

This section describes the FAQ about IPsec.

6.14.1 Private Network Communication Fails After IPsec Is Configured. What Are the Causes?

Private networks fail to communicate with each other after IPsec is configured. The possible causes are as follows:

- The public addresses of two IPsec-enabled devices cannot ping each other.
- The data flow defined for IPsec encapsulation is the same as that defined for NAT. You can run the **display acl all** command to view the matching ACL rule. In this case, use either of the following methods to prevent the data flow overlapping:
 - Ensure that the destination IP address in the ACL rule referenced by IPsec is denied in the ACL rule referenced by NAT. By doing so, the device does not perform NAT on the data flow protected by IPsec.
 - The ACL rule referenced by IPsec matches NAT-translated IP address.
- The device incorrectly learns private routes. The outbound interface to the destination private network is not the public network interface with IPsec enabled.
- When the authentication algorithm used in an IPsec proposal is SHA2, the encryption and decryption methods on both ends are inconsistent.

6.14.2 How Do I Rectify the Failure to View SA Information by Running the display ipsec sa Command After IPsec Is Configured?

To rectify the failure, perform the following operations:

1. Perform the ping operation to check the public network connectivity.

2. If the IPsec tunnel is established through IKE negotiation, run the **display ike sa** command to check whether the IKE SA is successfully established.
3. Wait about 10 seconds and run the **display ipsec sa** command again.
4. Run the **display interface brief** command to verify that the interface bound to the IPsec policy is in the **Up** state.
5. Verify that IPsec is configured correctly.

6.14.3 Does the Interface with a Dynamic IP Address Support IPsec?

Yes.

When the local interface has a dynamic IP address and the peer interface has a fixed IP address, configure an IPsec policy template on the peer interface to implement IPsec.

The following uses the 3G interface as an example to implement IKE auto negotiation.

Dynamic IP address

```
#
ike peer peer_3g_1
  pre-shared-key cipher %%@VsiNAx"H;$1jaO'QE%[=I\O6%0%0 //Set the pre-shared key
to huawei.
  remote-address 10.5.39.160 //Specify a fixed IP address for the peer end.
#
ipsec proposal ipsec //Use the default security parameters.
#
ipsec policy ipsec 1 isakmp //Configure an IPsec policy and import the policy on
a 3G interface.
  security acl 3000
  ike-peer peer_3g_1
  proposal ipsec
#
interface Cellular0/0/0
  ipsec policy ipsec //Configure the IPSEC policy on the 3G interface.
#
acl 3000 //Configure ACL rules. The IPsec policy protects packets that match ACL
rules.
...
#
```

Fixed IP address

```
#
ipsec proposal ipsec
#
ike peer peer_3g_2 //The peer end uses a dynamic IP address.
  pre-shared-key cipher %%@VsiNAx"H;$1jaO'QE%[=I\O6%0%0 //Set the pre-shared key
to huawei.
#
ipsec policy-template temp 1 //Configure an IPsec policy template.
  ike-peer peer_3g_2
  proposal ipsec
#
ipsec policy ipsec 1 isakmp template temp //Configure an IPsec policy and bind
the policy to the template.
#
interface GigabitEthernet 1/0/0 //This interface uses a fixed IP address.
  ipsec policy ipsec
  ip address 10.5.39.160 255.255.255.255
#
```

 **NOTE**

In V200R002C00 and earlier versions, run the **pre-shared-key huawei** command to set the pre-shared key to huawei.

In V200R008C00 and later versions, the **v1** and **v2** parameters are deleted from the **ike peer peer-name [v1 | v2]** command. To configure the IKE protocol, run the **version { 1 | 2 }** command.

6.14.4 IPsec Does Not Take Effect When Both IPsec and NAT Are Configured on a Device Interface. How This Problem Is Solved?

If NAT is configured on an interface to which an IPsec policy is applied, IPsec may not take effect. You can use the following methods:

- Configure the destination IP address that matches the deny clause in an ACL referenced by NAT as the destination IP address in an ACL referenced by IPsec. In this case, data flows protected by IPsec are not translated by NAT.
- Configure the ACL rule referenced by NAT to match the IP address translated by NAT.

 **NOTE**

After a deny rule is configured in NAT, you are advised to run the **reset session all** or **reset nat session all** command to delete incorrect NAT entries.

6.14.5 Why Cannot an IPsec Tunnel Be Established Until It Is Restarted?

When the same traffic needs to be sent to the headquarters but an IPsec tunnel exists between the branch and headquarters to protect access traffic from existing users, a new IPsec tunnel cannot be established before the old IPsec tunnel is torn down after it is restarted on the headquarters device.

To solve the problem, run the **ipsec remote traffic-identical accept** command to enable new users with the same IPsec rule to quickly access the headquarters. This function quickly ages an existing IPsec SA between the branch and headquarters to re-establish a new IPsec tunnel.

6.15 References for IPsec

The following table lists the references for IPsec.

Table 6-12 References for IPsec

Document	Description
RFC 1829	The ESP DES-CBC Transform
RFC 1851	The ESP Triple DES Transform
RFC 2367	PF_KEY Key Management API, Version 2
RFC 2401	Security Architecture for the Internet Protocol
RFC 2402	IP Authentication Header
RFC 2403	The Use of HMAC-MD5-96 within ESP and AH

Document	Description
RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH
RFC 2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC 2406	IP Encapsulating Security Payload (ESP)
RFC 2407	The Internet IP Security Domain of Interpretation for ISAKMP
RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)
RFC 2409	The Internet Key Exchange (IKE)
RFC 2410	The NULL Encryption Algorithm and Its Use With IPsec
RFC 3456	Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode
RFC 3706	A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
RFC 3947	Negotiation of NAT-Traversal in the IKE
RFC 3948	UDP Encapsulation of IPsec ESP Packets
RFC 4301	Security Architecture for the Internet Protocol
RFC 4302	IP Authentication Header
RFC 4303	IP Encapsulating Security Payload (ESP)
RFC 4306	Internet Key Exchange (IKEv2) Protocol
RFC 4307	Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
RFC 4308	Cryptographic Suites for IPsec
RFC 4434	The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)
RFC 4478	Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
RFC 4718	IKEv2 Clarifications and Implementation Guidelines
RFC 4835	Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4945	The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
RFC 5996	Internet Key Exchange Protocol Version 2 (IKEv2)

Document	Description
draft-dukes-ike-mode-cfg-02	The ISAKMP Configuration Method
draft-ietf-ipsec-heartbeats-00	Using Isakmp Heartbeats for Dead Peer Detection

7 A2A VPN Configuration

About This Chapter

- [7.1 Overview of A2A VPN](#)
- [7.2 Understanding A2A VPN](#)
- [7.3 Application Scenarios for A2A VPN](#)
- [7.4 Licensing Requirements and Limitations for A2A VPN](#)
- [7.5 Default Settings for A2A VPN](#)
- [7.6 Configuring A2A VPN](#)
- [7.7 Maintaining A2A VPN](#)
- [7.8 Configuration Examples for A2A VPN](#)
- [7.9 Troubleshooting A2A VPN](#)
- [7.10 A2A VPN FAQ](#)
- [7.11 References for A2A VPN](#)

7.1 Overview of A2A VPN

Definition

Any to Any VPN (A2A VPN) is a VPN solution that uses the Group Domain of Interpretation (GDOI) protocol to manage keys and GDOI policies in a centralized manner. A2A VPN is mainly used to protect enterprises' internal service traffic that is transmitted over a wide area network (WAN).

For details about GDOI, see RFC 6407.

Purpose

As networks develop, enterprises have not only data services but also increasing intelligent services such as voice and video services. These new services impose demands for instant

interconnection between enterprise branches. Generally, enterprises deploy dedicated lines such as MPLS VPN to implement interconnection between branches.

However, dedicated lines provide secure communication for enterprises to only a certain extent. Some government regulations, such as Health Insurance Portability and Accountability Act (HIPAA) and Payment Card Industry Data Security Standard (PCI DSS), require that data must be encrypted before it can be transmitted over dedicated lines.

Currently, IPSec is the commonly used encryption solution for dedicated lines. IPSec is a Layer 3 encryption protocol defined by the Internet Engineering Task Force (IETF) and is widely used for data encryption in WAN interconnections between branches. As a traditional Layer 3 VPN technology, IPSec sets up tunnels between specified communicating parties to protect data confidentiality, providing high-quality, interoperable, and cryptology-based security.

IPSec VPN is a point-to-point tunneling technology that focuses on data security and encryption. It has the following disadvantages:

- Networks face the N^2 problem (N branches require $N(N-1)/2$ tunnels). The configuration and management are complicated and network expansion is difficult.
- IPSec VPN results in changes to the original route deployment and cannot provide better QoS processing.
- IPSec VPN does not support multicast services and can hardly support intelligent services.

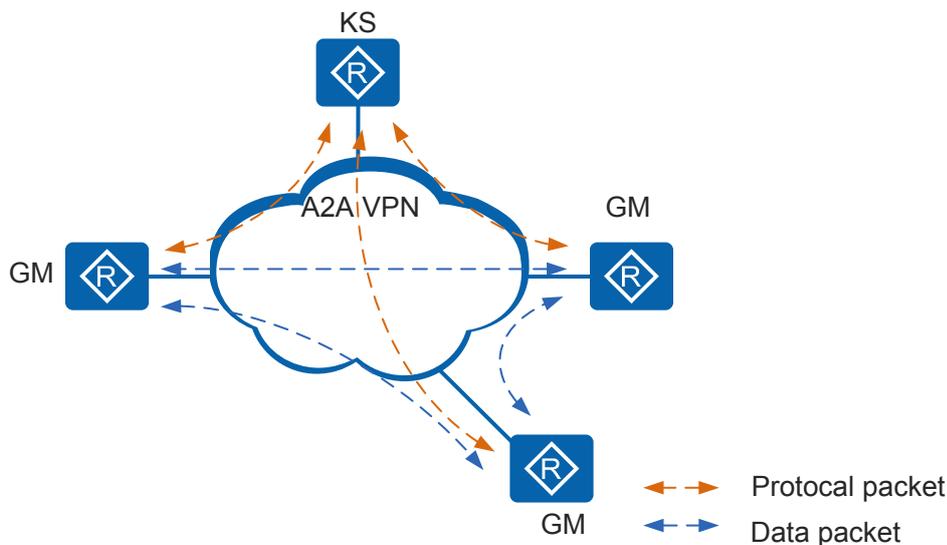
The A2A VPN solution is developed to overcome the preceding disadvantages. A2A VPN adds a new IP header, same as the raw IP header, to establish non-tunnel connections between branches. It manages keys and GDOI policies in a centralized manner, simplifying network deployment and facilitating network expansion. In addition, it supports multicast features and provides QoS guarantee for voice and video services.

7.2 Understanding A2A VPN

7.2.1 Basic Networking

As shown in [Figure 7-1](#), the basic A2A VPN networking is composed of two types of devices: key server (KS) and group member (GM). A2A VPN provides a group-based IPSec security model. A group is a collection of GDOI policies, and all the GMs in the same group share the same GDOI policies and keys.

Figure 7-1 Basic A2A VPN networking



GM

GMs are a group of network devices that share the same GDOI policies and keys and have the same security requirements. Generally, GMs are branch egress routers. A GM registers with the KS, and obtains GDOI policies from the KS to communicate with other GMs in the same group. A GM provides a group identifier (ID) when it registers with the KS, and the KS delivers matching GDOI policies and keys to the GM based on the group ID.

KS

The KS is a network device that creates and maintains GDOI policies and keys. Generally, the KS is a router located beside the egress router of a data center. The KS responds to registration requests from GMs and sends Rekey messages to GMs. After a GM registers with the KS, the KS delivers the GDOI policies and keys to the GM. The keys will be updated periodically. Before the key lifetime is reached, the KS sends Rekey messages to instruct all the GMs to update keys.

The KS delivers two types of keys:

- Traffic encryption key (TEK): shared by all the GMs in a group and used for encryption and decryption of traffic between GMs.
- Key encryption key (KEK): shared by all the GMs in a group and used for encryption and decryption of Rekey messages between the KS and GM.

NOTE

AR routers cannot function as the KS.

7.2.2 Implementation

7.2.2.1 GM Registering with the KS

After GDOI policies of A2A VPN are applied to an interface of a GM, the GM registers with the KS. The registration process consists of two stages: IKE negotiation and GDOI negotiation.

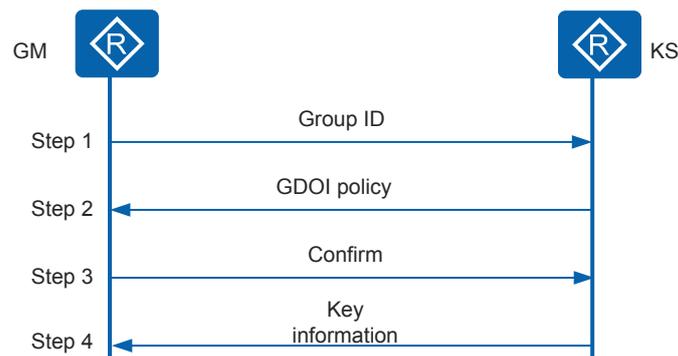
1. IKE negotiation (first stage): The GM and KS negotiate with each other to authenticate the peer's identity and to establish an IKE SA after identity authentication succeeds.

For details on the IKE negotiation process, see "**IKE**" in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - IPsec*.

2. GDOI negotiation (second stage): After the IKE SA is established in the first stage, the GM uses the GDOI protocol to negotiate with the KS and download keys (KEK and TEK) from the KS.

Figure 7-2 outlines the detailed GDOI negotiation process.

Figure 7-2 GDOI negotiation process



- a. The GM obtains the group ID configured by an administrator and sends the group ID to the KS.
- b. The KS authenticates the received group ID. After the authentication succeeds, the KS delivers GDOI policies (including information about the data flows to be protected, authentication algorithm, encryption algorithm, and encapsulation mode) to the GM based on the group ID.
- c. The GM authenticates the received GDOI policies. If the policies are acceptable (for example, the authentication and encryption algorithms are supported by the GM), the GM sends a Confirm message to the KS.
- d. After receiving the Confirm message, the KS sends keys (KEK and TEK) to the GM.

After the negotiation process completes, the GM obtains GDOI policies and keys from the KS, saves the information locally, and generates the TEK SA and KEK SA. The TEK SA protects data flows between GMs, and the KEK SA protects rekey messages between the GM and KS.

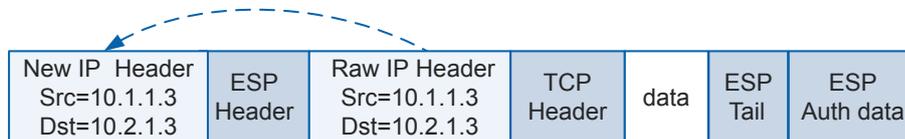
7.2.2.2 GM Data Protection

After a GM registers with the KS, it uses the obtained TEK SA to protect packets matching the GDOI policies. The protected packets can be unicast or multicast packets.

A2A VPN uses a data encapsulation mode similar to that used in IPsec. However, A2A VPN can only use the Encapsulating Security Payload (ESP) protocol to encrypt packets, and supports only the tunnel encapsulation mode. The encapsulation and encryption modes are configured on the KS and delivered to the GMs.

In tunnel mode, the device adds an ESP header to the raw IP header and then adds a new IP header that is the same as the raw IP header to the ESP header. In this way, the encrypted packets retain the raw IP header information, including the source and destination addresses and protocol type. Using a TCP packet as an example, **Figure 7-3** shows the structure of the IP packet generated after the encapsulation.

Figure 7-3 A2A VPN tunnel encapsulation mode



In an A2A VPN solution, encapsulated packets contain raw IP header information; therefore, the packets can be forwarded using the existing routing architecture, making full use of the existing network structure.

Apart from the encapsulation mode, the data processing method of A2A VPN is of great importance. A2A VPN supports three SA modes:

- **Receive_Only:** After a GM successfully registers with the KS, the GM can receive both ciphertext and plaintext packets but can send only plaintext packets.
- **Receive_Option:** After a GM successfully registers with the KS, the GM can receive both ciphertext and plaintext packets but can send only ciphertext packets.
- **Normal:** After a GM successfully registers with the KS, the GM can send or receive ciphertext packets only.

If SA modes cannot be configured in stages when you deploy the A2A VPN on an existing network (such as the MPLS VPN), services will be interrupted because a GM that joins a group can send and receive only ciphertext packets but the GM that has not joined the group can send and receive only plaintext packets. In this case, you can deploy the A2A VPN in stages in SA mode to solve this issue. The process is as follows:

1. Deploy the KS and set the SA mode to **Receive_Only** on the KS. The KS will deliver this SA mode to the GMs.
2. Deploy GMs on the A2A VPN network one by one. The registered GMs work in **Receive_Only** mode, so they can communicate with the newly deployed GMs before they register with the KS.
3. After all the GMs are deployed, set the SA mode to **Receive_Option** on the GMs. A GM in **Receive_Option** mode can still communicate with a GM in **Receive_Only** mode.
4. After all the GMs are switched to **Receive_Option** mode, set the SA mode to **Normal** on the KS.

7.2.2.3 Rekey

The KS not only creates and encrypts GDOI policies and keys, but also updates keys and allocates keys to GMs. After GMs register with the KS, the KS periodically sends new TEK SAs or KEK SAs to GMs through rekey messages to improve security. A rekey message carries information about the data flows to be protected, authentication algorithm, encryption algorithm, encapsulation mode, and the lifetime of keys and SAs. A rekey message is protected by the current KEK SA. GMs will receive rekey messages from the KS periodically. If a GM does not receive any rekey message within the lifetime of the TEK SA or KEK SA, it needs to register with the KS again to download GDOI policies and keys.

Two rekey modes are available: multicast rekey and unicast rekey.

Multicast Rekey

In multicast rekey mode, the KS multicasts rekey messages to members in a multicast group. After successfully registering with the KS, GMs join a specific multicast group. All GMs in the group will receive the rekey messages. If the GDOI policies on the KS are modified, the KS sends rekey messages to all GMs in the multicast group.

Unicast Rekey

In unicast rekey mode, the KS unicasts rekey messages to each GM. The KS sends a rekey message carrying the same new TEK SA or KEK SA to all the GMs before the old SA expires. After receiving the rekey message, each GM sends an ACK message to the KS. The KS updates the current active GM list based on the received ACK message. The KS sends rekey messages only to active GMs.

In unicast mode, if the KS does not receive ACK messages from a GM for three consecutive rekey messages, it removes the GM from the current active GM list and stops sending rekey messages to the GM. If the GM wants to receive rekey messages, it must register with the KS again.

7.3 Application Scenarios for A2A VPN

7.3.1 Typical A2A VPN Networking

Although the traditional IPsec VPN can meet the encryption requirement, it cannot implement instant interconnection between enterprise branches and cannot provide better QoS or multicast services. Besides, the network deployment is complex and network maintenance is difficult. With ever increasing network security risks, a WAN interconnection solution is in urgent need for a balance among security, intelligence, and easy management.

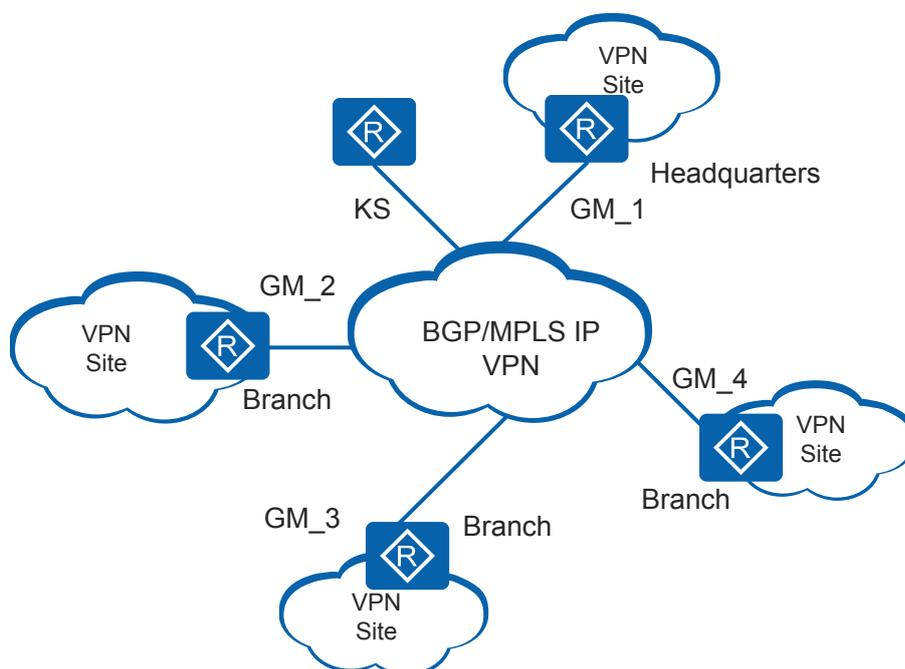
A2A VPN uses a KS to manage keys in a centralized manner, shares keys among GDOI group members, and allows for hierarchical encryption and decryption among GMs. Compared with traditional tunnel encryption, this solution simplifies configuration, facilitates network expansion, and improves reliability, providing security for WAN interconnection and intelligent service deployment.

As shown in [Figure 7-4](#), a large-sized enterprise leases an MPLS VPN network (for example, BGP/MPLS IP VPN) from a carrier to construct dedicated lines for connecting its branches across the country. The headquarters and branches need to frequently exchange packets of

complex services including data, voice, and video services. These service packets must be encrypted to prevent data tampering.

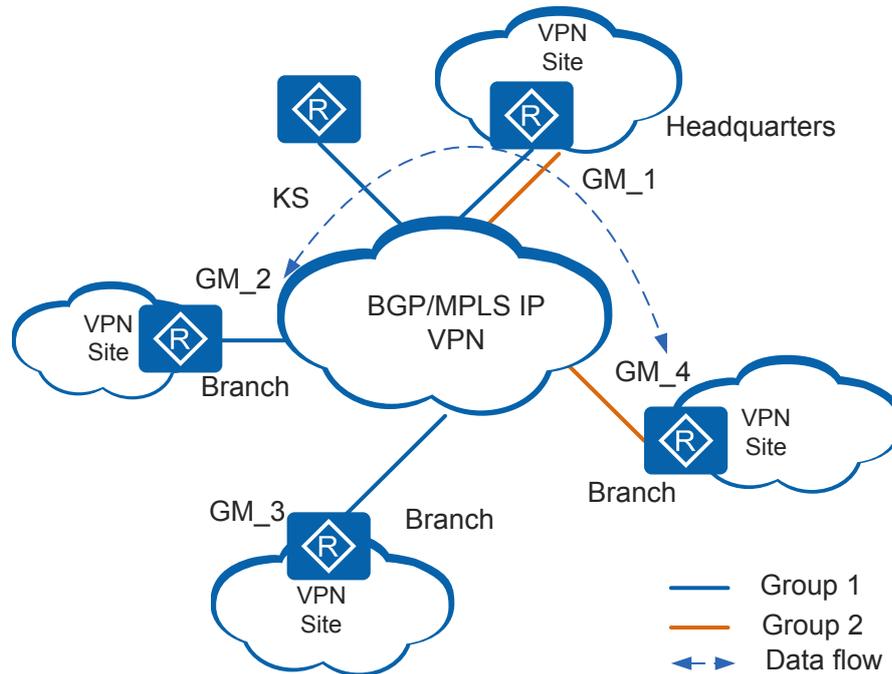
To meet the enterprise's requirements, a KS is deployed in the headquarters, with shared keys and GDOI policies configured. The GDOI policies define the mode for encrypting service packets transmitted between the branches and headquarters. The egress routers of the headquarters and branches function as GMs. The GMs register with the KS to obtain the shared keys and GDOI policies, which they use to encrypt/decrypt and forward packets. After this solution is deployed, data packets transmitted between the branches or between a branch and headquarters are encrypted; therefore, services can be safely transmitted over the carrier network.

Figure 7-4 Typical A2A VPN networking



The enterprise requires that branch GM_4 can directly communicate with headquarter GM_1, but traffic between GM_4 and other branches must pass through the headquarters. As shown in [Figure 7-5](#), two outbound interfaces are configured on GM_1: one interface is added to group 1 together with GM_2 and GM_3, and the other interface is added to group 2 together with GM_4. GM_4 cannot directly communicate with other branches as they are not in the same group. Traffic between GM_4 and another branch, such as GM_2, must pass through the headquarters.

Figure 7-5 Headquarters router interfaces joining different groups



NOTE

When the router functions the PE device, it does not support multicast Rekey.

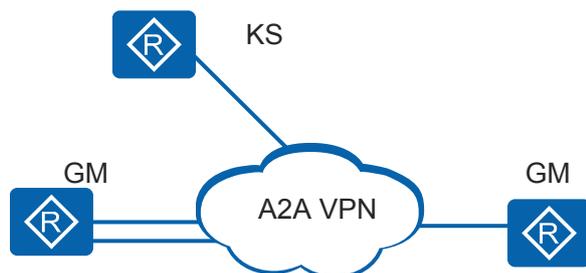
7.3.2 A2A VPN Redundancy

To improve network reliability, redundant links can be deployed for GMs (link redundancy) or a GM can register with four KSs (KS redundancy).

Link Redundancy

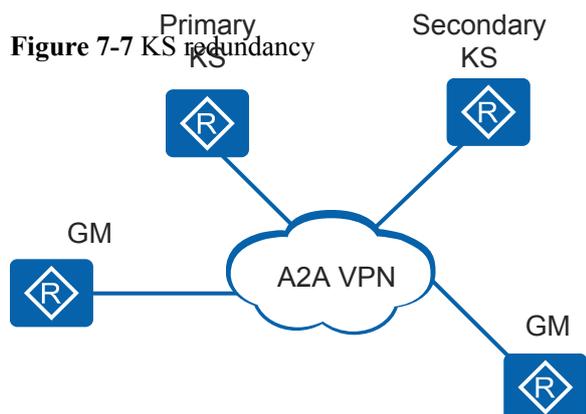
As shown in [Figure 7-6](#), each GM has two egress links that have the same group ID and are bound with different GDOI policy groups. The egress links register with the KS and download the same GDOI policies. When traffic needs to be forwarded, the GM selects the outbound interface based on routing information. If any link fails, traffic is switched to the other link based on route convergence, ensuring uninterrupted traffic between the GMs and service availability.

Figure 7-6 Link redundancy



KS Redundancy

As shown in [Figure 7-7](#), each GM registers with two KSs: one primary KS and one secondary KS. Generally, a GM registers with the primary KS. If the primary KS fails, the GM registers with the secondary KS. This improves reliability of the entire network.



7.4 Licensing Requirements and Limitations for A2A VPN

Involved Network Elements

- Key server
- Group member

Licensing Requirements

A2A VPN is a basic feature of the device and is not under license control.

Feature Limitations

- The device can only function as the GM but not the KS.

- When the device connects to a Cisco device that is used as the KS, you are advised to use the Cisco device version of 2013 or later, for example, **Cisco IOS Software, C3900e Software (C3900e-UNIVERSALK9-M), Version 15.2(4)M5, RELEASE SOFTWARE (fc2)**. If the Cisco device of a version earlier than 2013 is used, the multicast rekey may fail.
- When the TEK SA lifetime is reached, the GM sends a re-registration request to the KS. When the KEK SA lifetime is reached, the GM does not send a re-registration request to the KS.
- The GM does not support time-based anti-replay. If time-based anti-replay is configured on a KS, traffic will be interrupted when the GM communicates with a GM from another vendor that supports time-based anti-replay.
- A2A VPN does not support NAT traversal.
- The GM can register with four KSs. The KSs work in primary/secondary mode according to the configuration sequence. The GM first attempts to register with the first KS. If the registration fails, the GM tries the second KS. This process continues in the preceding manner.

7.5 Default Settings for A2A VPN

Table 7-1 Default settings for A2A VPN

Parameter	Default Setting
IKE proposal	The system provides an IKE proposal with the lowest priority by default. For the detailed parameters, see ike proposal .
Fragmentation mode of A2A VPN packets	Fragmentation after encryption

7.6 Configuring A2A VPN

7.6.1 Configuring a GM

Pre-configuration Tasks

Before configuring a GM, complete the following tasks:

- Configure reachable routes between the GMs and between the GM and KS.
- Determine the data flows to be protected by A2A VPN.
- Determine the encryption algorithm for IKE negotiation, that is, configure parameters for an IKE proposal.
- Specify the PKI domain to which the peer belongs if digital certificate PKI authentication is used.

 **NOTE**

For details on the PKI configuration, see "PKI Configuration" in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - Security*.

Configuration Procedure

Perform the following operations in sequence to configure a GM. You can determine whether to perform optional operations based on site requirements.

7.6.1.1 Configuring IKE

Context

To configure IKE, you need to configure an IKE proposal and an IKE peer for the first-stage IKE negotiation between the GM and KS.

 **NOTE**

GM does not support IKEv2.

Procedure

In A2A VPN, the detailed IKE configuration procedure is as follows:

1. Configure an IKE proposal. For details, see "[6.10.1 Configuring an IKE Proposal](#)" in "IPSec Configuration".
2. (Optional) Configure the IKE SA Lifetime, see "[6.10.3 \(Optional\) Setting the IKE SA Lifetime](#)" in "IPSec Configuration".
3. Configure an IKE peer. For details, see "[6.10.2 Configuring an IKE Peer](#)" in "IPSec Configuration".
4. (Optional) Bind a VPN instance to A2A VPN. For details, see "[6.10.8 \(Optional\) Configuring IPSec VPN Multi-instance](#)" in "IPSec Configuration".

7.6.1.2 (Optional) Defining Data Flows Not to Be Protected

Context

After a GM registers with the KS, the GM obtains security policies including the ACL rules from the KS. The ACL rules configured on the KS determine data flows to be encrypted and data flows to be forwarded in plain text. The GM matches received data flows with the ACL rules obtained from the KS. If the matched ACL rule defines a permit action, the GM encrypts the data flows and forwards them; if the matched ACL rule defines a deny action, the GM forwards the data flows in plain text. If a data flow matches an ACL rule that defines a permit action, but the GM wants to forward the data flow in plain text, you can configure a local ACL rule to define data flows not to be protected. When a device forwards a data flow, it matches the data flow with the local ACL rule first. If the data flow matches the local ACL rule, the device forwards the data flow in plain text. If not, the device matches the data flow with the ACL rules obtained from the KS.

 **NOTE**

If the KS sends more than 100 ACL rules to a GM, only 100 ACL rules take effect. If more than 100 ACL rules are configured on a GM, only 100 ACL rules take effect.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Run **acl** [**number**] *acl-number* [**match-order** { **config** | **auto** }]

An advanced ACL with *acl-number* ranging from 3000 to 3999 is created and the advanced ACL view is displayed.

Step 3 Run **rule** [*rule-id*] **deny ip** [**destination** { *destination-address destination-wildcard* | **any** } | **source** { *source-address source-wildcard* | **any** } | **vpn-instance** *vpn-instance-name* | **dscp** *dscp*] *

An ACL rule is configured in the ACL view.

A2A VPN also support advanced ACLs that define various protocols such as ICMP, TCP, UDP, and IGMP.

---End

7.6.1.3 Configuring a GDOI Policy

Context

The name and sequence number identify a GDOI policy.

A GDOI policy contains key information submitted by a GM when it registers with the KS. The information includes the group ID, referenced IKE peer, and the registration interface.

- Group ID: identifies a GDOI group in A2A VPN. The KS determines the GDOI group to which a GM is to be added based on the group ID submitted by the GM.
- Referenced IKE peer: used for the first stage IKE negotiation.
- Registration interface: A GM registers with the KS through the registration interface.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy** *policy-name seq-number* **gdoi**

A GDOI policy is created and the GDOI policy view is displayed.

By default, no GDOI policy is configured in the system.

Step 3 Run **group identity number** { *group-number* | *ip-address* }

An identifier is configured for a GDOI group.

By default, a GDOI group has no identifier.

The group identifier is also called group ID. A GDOI group can only use its group number or IP address as the group ID and can only have one group ID.

Step 4 Run **ike-peer** *peer-name*

An IKE peer is applied to the GDOI policy.

peer-name specifies the name of a created IKE peer.

By default, no IKE peer is referenced in the system.

Step 5 (Optional) Run **security acl** *acl-number*

An ACL is referenced in a GDOI policy.

acl-number specifies an advanced ACL that defines the data flows not to be protected.

By default, no ACL is referenced.

A GDOI policy can reference only one ACL.

A local ACL (supporting the deny action only) has a higher priority than an ACL downloaded from the KS.

Step 6 (Optional) Run **tunnel local** { *ip-address* | **applied-interface** }

The local address is configured for A2A VPN.

By default, no local address is configured for A2A VPN.

For a GDOI policy, you do not need to configure a local address for A2A VPN because the device will select an appropriate local address based on routing information during SA negotiation.

- If the IP address of the interface to which a GDOI policy is applied varies or is unknown, run the **tunnel local** *ip-address* command to specify the IP address of another interface (such as the loopback interface) on the device as the local IP address for A2A VPN. Otherwise, run the **tunnel local** **applied-interface** command to specify the IP address of the interface as the local IP address for A2A VPN.
- If the interface to which a GDOI policy is applied has multiple IP addresses (one primary IP address and several secondary IP addresses), run the **tunnel local** *ip-address* command to specify one of these IP addresses as the local IP address for A2A VPN. Otherwise, run the **tunnel local** **applied-interface** command to specify the primary IP address of the interface as the local IP address for A2A VPN.
- If equal-cost routes exist between the local and remote ends, run the **tunnel local** { *ip-address* | **applied-interface** } command to specify a local IP address for A2A VPN.

----End

7.6.1.4 Configuring an IP Address for Multicast Rekey Messages

Context

After an IP address for multicast Rekey messages is configured, GMs with this IP address and a UDP port number 848/4500 update TEK SAs or KEK SAs based on the multicast Rekey messages.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec gdoi multicast-rekey ip** *ip-address*

An IP address is configured for multicast Rekey messages.

By default, no IP address is configured for multicast Rekey messages.

The IP address for multicast Rekey messages configured on the GM must be the same as that configured on the KS.

----End

7.6.1.5 (Optional) Configuring the Receive_Option Mode

Context

If SA modes cannot be configured in stages when you deploy A2A VPN on an existing network (such as the MPLS VPN network), services will be interrupted because a GM that joins a group can send and receive only cipher text packets but the GM that has not joined the group can send and receive only plain text packets. To prevent this problem, you can set the SA mode to Receive_Option on the GMs and set the SA mode to Receive_Only or Normal in different stages, to implement the phased deployment to A2A VPN.

The SA mode deployment of GMs and the KS are as follows:

1. Deploy the KS and set the SA mode to Receive_Only on the KS. The KS will deliver this SA mode to the GMs.
2. Deploy GMs on the A2A VPN network one by one. The registered GMs work in Receive_Only mode, so they can communicate with the newly deployed GMs before they register with the KS.
3. After all the GMs are deployed, set the SA mode to Receive_Option on the GMs. A GM in Receive_Option mode can still communicate with a GM in Receive_Only mode.
4. After all the GMs are switched to Receive_Option mode, set the SA mode to Normal on the KS.

When configuring the SA mode on the KS, refer to the KS configuration of other vendors.

Procedure

Step 1 Run `gdoi sa direction receive-option`

The SA mode is set to Receive_Option. In this mode, the device can receive both cipher text and plain text packets but can send only cipher text packets.

----End

7.6.1.6 (Optional) Configuring the QoS Function for A2A VPN

Context

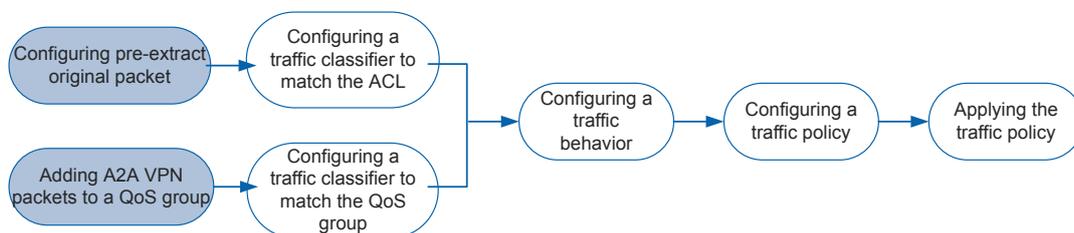
In network planning, the QoS function needs to be configured to improve network service capabilities and provide differentiated services to different types of traffic. QoS groups the packets sharing common features into one class and provides the same QoS level for traffic of the same type. In this manner, the class-based QoS technology provides differentiated services.

You can configure the QoS function to implement refined QoS management on A2A VPN packets. Choose either of the following method based on site requirements:

- If you want to classify packets to be encapsulated based on quintuple information including the source address, destination address, protocol type, source port number, and destination port number, configure the pre-extract original packet function. If you want to classify packets based on the source address, destination address, or protocol type only, you do not need to configure the pre-extract original packet function because the new IP header added to A2A VPN packets is the same as the raw IP header.
- Packet encapsulation and decapsulation may result in transmission delay and consume network bandwidth; therefore, A2A VPN requires differentiated services to shorten the transmission delay, reduce the packet loss ratio, and maximize bandwidth. Classify A2A VPN packets into a QoS group to provide differentiated services for A2A VPN packets in the QoS group.

Figure 7-8 shows the procedure for configuring QoS.

Figure 7-8 Procedure for configuring QoS



For details on QoS, see "**MQC Configuration**" in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series V200R009 Configuration Guide - QoS*.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec policy policy-name seq-number gdoi**

The GDOI policy view is displayed.

Step 3 Configure the QoS function for A2A VPN.

- Run **qos pre-classify**
The device is configured to pre-extract original packet information.
By default, the device does not pre-extract original packet information.
- Run **qos group qos-group-value**
A QoS group is configured for A2A VPN packets.
By default, A2A VPN packets do not belong to a QoS group.

----End

Follow-up Procedure

- After configuring the pre-extract original packet function, you need to run the **if-match acl { acl-number | acl-name }** command in the traffic classifier view to create ACL-based matching rules.

- After adding A2A VPN packets to a QoS group, you need to run the **if-match qos-group qos-group-value** command in the traffic classifier view to create QoS group-based matching rules.

7.6.1.7 (Optional) Configuring Fragmentation Before Encryption

Context

After packets are encapsulated in A2A VPN, the packet length may exceed the maximum transmission unit (MTU) allowed by the device outbound interface. In this case, you need to fragment the packets to prevent packet loss. Two methods are available to fragment packets:

- Fragmentation before encryption: Before encapsulating A2A VPN packets, the encryption device calculates the predicted length of the encapsulated packets. If the predicted length of the encapsulated packets exceeds the MTU of the outbound interface, the device fragments the A2A VPN packets and then encapsulates the fragmented packets. In this case, the terminal host decrypts and assembles A2A VPN fragments. This reduces the CPU usage of the peer decryption device.
- Fragmentation after encryption: If the size of the encapsulated A2A VPN packets exceeds the MTU of the outbound interface, the encryption device fragments the packets based on the MTU of the outbound interface. In this case, the peer decryption device assembles and decrypts A2A VPN fragments and then sends decrypted packets to the terminal host.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ipsec df-bit { clear | set | copy }**

The Don't Fragment (DF) flag bit is configured for A2A VPN packets, indicating whether packets can be fragmented.

By default, the DF flag bit in A2A VPN is the flag bit of original packets.

Only after **clear** is specified to allow packet fragmentation, the following command takes effect.

Step 3 Run **ipsec fragmentation before-encryption**

The fragmentation mode of packets is set to fragmentation before-encryption.

By default, packets are fragmented after being encrypted.

---End

7.6.1.8 Applying a GDOI Policy Group to an Interface

Context

A GDOI policy group is a collection of security policies with the same name but different sequence numbers. In a GDOI policy group, a GDOI policy with a smaller sequence number has a higher priority.

To protect data flows on an interface through A2A VPN, you need to apply a GDOI policy group to the interface. If the GDOI policy group is deleted from the interface, the interface no longer provides the A2A VPN protection function.

When sending a data flow, an interface matches the data flow with the GDOI policies in the GDOI policy group in ascending order of sequence numbers. If the data flow matches a local ACL referenced in a GDOI policy, the interface forwards the data flow in plain text; if the data flow does not match any local ACL, the interface matches the data flow with the ACLs downloaded from the KS; if the data flow does not match any ACL downloaded from the KS, the interface forwards the data flow in plain text by default.

 **NOTE**

Only one GDOI policy group can be applied on an interface. A GDOI policy group can be applied to only one interface.

In a GDOI policy group, if multiple policies are bound to different IKE peers, the remote addresses specified in the IKE peers cannot be the same. Otherwise, IKE negotiation of some GDOI policies fails.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The interface view is displayed.

Step 3 Run **ipsec policy** *policy-name*

A GDOI policy group is applied to the interface.

By default, no GDOI policy group is applied to an interface.

----End

7.6.1.9 Verifying the GM Configuration

Prerequisites

All A2A VPN configurations are complete.

Procedure

- Run the **display ike peer** [**brief** | **name** *peer-name*] command to check information about the IKE peer.
- Run the **display ike proposal** [**number** *proposal-number*] command to check parameters in the IKE proposal.
- Run the **display ike sa** [**remote** *ipv4-address*] command to check brief information about IKE SAs.
- Run the **display ike sa** [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* command to check brief information about IKE SAs based on the remote ID.
- Run the **display ike sa verbose** { **remote** *ipv4-address* | **connection-id** *connection-id* | [**remote-id-type** *remote-id-type*] **remote-id** *remote-id* } command to check detailed information about IKE SAs.

- Run the **display ipsec gdoi-sa** [*policy-name* [*seq-number*]] command to check information about the GDOI SA.
- Run the **display ipsec gdoi-policy** [*policy-name* [*seq-number*]] command to check information about GDOI policies.

----End

7.7 Maintaining A2A VPN

7.7.1 Monitoring the A2A VPN Status

Context

All A2A VPN configurations are complete.

Procedure

- Run the **display ipsec gdoi-sa** [*policy-name* [*seq-number*]] command to check information about the GDOI SA.
- Run the **display ipsec statistics** command to check the statistics about IPSec packets.
- Run the **display ike statistics** { *v1* | *v2* } command to check the statistics about IKE packets.

----End

7.7.2 Clearing A2A VPN Statistics

Context



Statistics cannot be restored after being cleared. Exercise caution when you run the reset command.

Procedure

- To clear statistics about IPSec packets, run the **reset ipsec statistics** command in the user view.
- To clear statistics about IKE packets, run the **reset ike statistics** command in the user view.
- To clear the TEK SA and KEK SA in a created GDOI policy, run the **reset ipsec gdoi-sa** [*policy* *policy-name* [*seq-number*]] command in the user view.
- To clear the SA established by the current IKE, run the **reset ike sa** { *remote ip-address* | *conn-id* *conn-id* } command in the user view.

The **reset ike sa** command does not clear the GDOI SA.

----End

7.8 Configuration Examples for A2A VPN

7.8.1 Example for Configuring a Typical A2A VPN Networking

Networking Requirements

A large-scale enterprise has many branches distributed in a wide area and a large number of multicast services. BGP/MPLS IP VPN is deployed on the enterprise to implement secure communication between the headquarters and branches. As shown in [Figure 7-9](#), GM_1 is the enterprise branch gateway and GM_2 is the enterprise headquarters gateway. (The enterprise has only one branch in this example.)

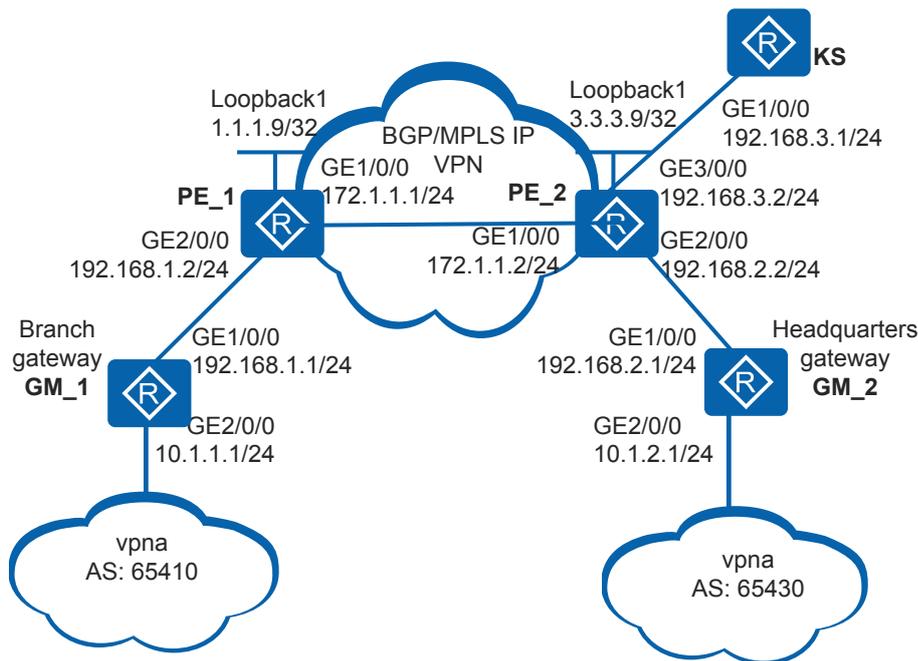
The enterprise requires that traffic between the branch and headquarters be encrypted and the existing MPLS VPN network structure be made full use of. A2A VPN can be deployed between the branch and headquarters to meet these requirements.

NOTE

The device can not function as a KS.

When the router functions the PE device, it does not support multicast Rekey.

Figure 7-9 Typical A2A VPN networking



Configuration Roadmap

The configuration roadmap is as follows:

1. Set the AS number of the KS to 65420 and perform the same configuration on PE_2 to add the KS to vpna, ensuring reachable routes between the KS and GMs.

2. Configure an ACL on the KS to define the data flows to be protected by A2A VPN.
3. Configure an IKE peer on the GMs and KS and define the attributes used for IKE negotiation.
4. Configure a security proposal on the KS to define the protection method used for A2A VPN.
5. Configure GDOI policies on the GMs and KS and reference an IKE peer.
6. Apply GDOI policy groups to the GM interfaces to enable the A2A VPN protection function on the interfaces.

 **NOTE**

When configuring the KS, refer to the configuration guide for KSs from other vendors.

Procedure

- Step 1** Configure OSPF on the MPLS backbone network so that the PE devices can communicate with each other.

Configure PE_1.

```
<Huawei> system-view
[Huawei] sysname PE_1
[PE_1] interface loopback 1
[PE_1-LoopBack1] ip address 1.1.1.9 32
[PE_1-LoopBack1] quit
[PE_1] interface gigabitethernet 1/0/0
[PE_1-GigabitEthernet1/0/0] ip address 172.1.1.1 24
[PE_1-GigabitEthernet1/0/0] quit
[PE_1] ospf 1
[PE_1-ospf-1] area 0
[PE_1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE_1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE_1-ospf-1-area-0.0.0.0] quit
[PE_1-ospf-1] quit
```

Configure PE_2.

```
<Huawei> system-view
[Huawei] sysname PE_2
[PE_2] interface loopback 1
[PE_2-LoopBack1] ip address 3.3.3.9 32
[PE_2-LoopBack1] quit
[PE_2] interface gigabitethernet 1/0/0
[PE_2-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[PE_2-GigabitEthernet1/0/0] quit
[PE_2] ospf
[PE_2-ospf-1] area 0
[PE_2-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE_2-ospf-1-area-0.0.0.0] network 3.3.3.9 0.0.0.0
[PE_2-ospf-1-area-0.0.0.0] quit
[PE_2-ospf-1] quit
```

- Step 2** Enable basic MPLS capabilities and MPLS LDP on the PE devices to set up LDP LSPs on the MPLS backbone network.

Configure PE_1.

```
[PE_1] mpls lsr-id 1.1.1.9
[PE_1] mpls
[PE_1-mpls] quit
[PE_1] mpls ldp
[PE_1-mpls-ldp] quit
[PE_1] interface gigabitethernet 1/0/0
[PE_1-GigabitEthernet1/0/0] mpls
```

```
[PE_1-GigabitEthernet1/0/0] mpls ldp
[PE_1-GigabitEthernet1/0/0] quit
```

Configure PE_2.

```
[PE_2] mpls lsr-id 3.3.3.9
[PE_2] mpls
[PE_2-mpls] quit
[PE_2] mpls ldp
[PE_2-mpls-ldp] quit
[PE_2] interface gigabitethernet 1/0/0
[PE_2-GigabitEthernet1/0/0] mpls
[PE_2-GigabitEthernet1/0/0] mpls ldp
[PE_2-GigabitEthernet1/0/0] quit
```

Step 3 Configure a VPN instance on each PE device and connect the GMs to the PEs.

Configure PE_1.

```
[PE_1] ip vpn-instance vpna
[PE_1-vpn-instance-vpna] ipv4-family
[PE_1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE_1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE_1-vpn-instance-vpna-af-ipv4] quit
[PE_1-vpn-instance-vpna] quit
[PE_1] interface gigabitethernet 2/0/0
[PE_1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE_1-GigabitEthernet2/0/0] ip address 192.168.1.2 24
[PE_1-GigabitEthernet2/0/0] quit
```

Configure PE_2.

```
[PE_2] ip vpn-instance vpna
[PE_2-vpn-instance-vpna] ipv4-family
[PE_2-vpn-instance-vpna-af-ipv4] route-distinguisher 200:1
[PE_2-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE_2-vpn-instance-vpna-af-ipv4] quit
[PE_2-vpn-instance-vpna] quit
[PE_2] interface gigabitethernet 2/0/0
[PE_2-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE_2-GigabitEthernet2/0/0] ip address 192.168.2.2 24
[PE_2-GigabitEthernet2/0/0] quit
```

Configure GM_1.

```
<Huawei> system-view
[Huawei] sysname GM_1
[GM_1] interface gigabitethernet 1/0/0
[GM_1-GigabitEthernet1/0/0] ip address 192.168.1.1 24
[GM_1-GigabitEthernet1/0/0] quit
```

Configure GM_2.

```
<Huawei> system-view
[Huawei] sysname GM_2
[GM_2] interface gigabitethernet 1/0/0
[GM_2-GigabitEthernet1/0/0] ip address 192.168.2.1 24
[GM_2-GigabitEthernet1/0/0] quit
```

Step 4 Set up an MP-IBGP peer relationship between PE1 and PE2.

Configure PE_1.

```
[PE_1] bgp 100
[PE_1-bgp] peer 3.3.3.9 as-number 100
[PE_1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE_1-bgp] ipv4-family vpnv4
[PE_1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE_1-bgp-af-vpnv4] quit
[PE_1-bgp] quit
```

Configure PE_2.

```
[PE_2] bgp 100
[PE_2-bgp] peer 1.1.1.9 as-number 100
[PE_2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE_2-bgp] ipv4-family vpnv4
[PE_2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE_2-bgp-af-vpnv4] quit
[PE_2-bgp] quit
```

Step 5 Set up EBGP peer relationships between the PEs and GMs and import VPN routes.

Configure GM_1.

```
[GM_1] bgp 65410
[GM_1-bgp] peer 192.168.1.2 as-number 100
[GM_1-bgp] import-route direct
[GM_1-bgp] quit
```

Configure GM_2.

```
[GM_2] bgp 65430
[GM_2-bgp] peer 192.168.2.2 as-number 100
[GM_2-bgp] import-route direct
[GM_2-bgp] quit
```

Configure PE_1.

```
[PE_1] bgp 100
[PE_1-bgp] ipv4-family vpn-instance vpna
[PE_1-bgp-vpna] peer 192.168.1.1 as-number 65410
[PE_1-bgp-vpna] import-route direct
[PE_1-bgp-vpna] quit
[PE_1-bgp] quit
```

Configure PE_2.

```
[PE_2] bgp 100
[PE_2-bgp] ipv4-family vpn-instance vpna
[PE_2-bgp-vpna] peer 192.168.2.1 as-number 65430
[PE_2-bgp-vpna] import-route direct
[PE_2-bgp-vpna] quit
[PE_2-bgp] quit
```

After the configuration is complete, the GMs can successfully ping each other.

For example, GM_1 can ping GM_2 (192.168.2.1).

```
[GM_1] ping 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=253 time=4 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=253 time=3 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=253 time=4 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=253 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=253 time=2 ms

--- 192.168.2.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/2/4 ms
```

Step 6 Configure PE_2 and the KS to add the KS to vpna and implement communication between the KS and GMs.

Configure PE_2.

```
[PE_2] bgp 100
[PE_2-bgp] ipv4-family vpn-instance vpna
```

```
[PE_2-bgp-vpna] peer 192.168.3.1 as-number 65420
[PE_2-bgp-vpna] quit
[PE_2-bgp] quit
[PE_2] interface gigabitethernet 3/0/0
[PE_2-GigabitEthernet3/0/0] ip binding vpn-instance vpna
[PE_2-GigabitEthernet3/0/0] ip address 192.168.3.2 24
[PE_2-GigabitEthernet3/0/0] quit
```

The VPN configuration of the KS is similar to that of GM_2. For details, see the *L3VPN Configuration Guide* of other vendors.

- Step 7** Configure an IKE peer for the two GMs. The IKE negotiation parameters must be the same as those on the KS.

Configure an IKE proposal on GM_1.

```
[GM_1] ike proposal 5
[GM_1-ike-proposal-5] quit
```

Create an IKE peer on GM_1.

```
[GM_1] ike peer spub
[GM_1-ike-peer-spua] undo version 2
[GM_1-ike-peer-spua] ike-proposal 5
[GM_1-ike-peer-spua] pre-shared-key cipher Huawei@1234
[GM_1-ike-peer-spua] remote-address 192.168.3.1
[GM_1-ike-peer-spua] quit
```

Configure an IKE proposal on GM_2.

```
[GM_2] ike proposal 5
[GM_2-ike-proposal-5] quit
```

Create an IKE peer on GM_2.

```
[GM_2] ike peer spua
[GM_2-ike-peer-spua] undo version 2
[GM_2-ike-peer-spua] ike-proposal 5
[GM_2-ike-peer-spua] pre-shared-key cipher Huawei@1234
[GM_2-ike-peer-spua] remote-address 192.168.3.1
[GM_2-ike-peer-spua] quit
```

- Step 8** Create GDOI policies for the GMs. The group ID of the GMs must be the same as that of the KS.

Configure GM_1.

```
[GM_1] ipsec policy map1 10 gdoi
[GM_1-ipsec-policy-gdoi-map1-10] group identity number 10
[GM_1-ipsec-policy-gdoi-map1-10] ike-peer spub
[GM_1-ipsec-policy-gdoi-map1-10] quit
```

Configure GM_2.

```
[GM_2] ipsec policy map2 10 gdoi
[GM_2-ipsec-policy-gdoi-map2-10] group identity number 10
[GM_2-ipsec-policy-gdoi-map2-10] ike-peer spua
[GM_2-ipsec-policy-gdoi-map2-10] quit
```

- Step 9** Apply a GDOI policy group to each interface on each GM.

Apply a GDOI policy group to the interface on GM_1.

```
[GM_1] interface gigabitethernet 1/0/0
[GM_1-GigabitEthernet1/0/0] ipsec policy map1
[GM_1-GigabitEthernet1/0/0] quit
```

Apply a GDOI policy group to the interface on GM_2.

```
[GM_2] interface gigabitethernet 1/0/0
[GM_2-GigabitEthernet1/0/0] ipsec policy map2
[GM_2-GigabitEthernet1/0/0] quit
```

Step 10 Verify the configuration.

After the configurations are complete, run the **display ike sa** command on each device to check the configuration. The command output of GM_1 is used as an example.

```
[GM_1] display ike sa
Conn-ID Peer VPN Flag(s) Phase
-----
1109 192.168.3.1 0 RD|ST v1:1

Number of IKE SA : 1
-----

Flag Description:
RD--READY ST--STAYALIVE RL--REPLACED FD--FADING TO--TIMEOUT
HRT--HEARTBEAT LKG--LAST KNOWN GOOD SEQ NO. BCK--BACKED UP
M--ACTIVE S--STANDBY A--ALONE NEG--NEGOTIATING
```

After the configurations are complete, run the **display ipsec gdoi-sa** command on each device to check the configuration. The command output of GM_1 is used as an example.

```
[GM_1] display ipsec gdoi-sa
=====
Interface: GigabitEthernet1/0/0
Path MTU: 1500
=====
-----
Gdoi policy name : "map1"
Sequence number : 10
-----
[TEK SA]
Protected vrf : vpna
Protocol : 0
Flow source : 10.1.1.0/255.255.255.0/0
Flow destination : 10.1.2.0/255.255.255.0/0

Inpacket count : 0
Inpacket decap count : 0
Outpacket count : 0
Outpacket encap count : 0
Inpacket drop count : 0
Outpacket drop count : 0

SA mode : normal
SPI: 99152831 (0x5e8f3bf)
Proposal : ESP-ENCRYPT-3DES-192 ESP-AUTH-SHA1
SA remaining lifetime(secs) : 66409
Anti-Replay(Time Based) : disable

[KEK POLICY]
Rekey Transport Type : multicast
SPI: 0x2ad569a935d15b75174446fbb0feaf5b
Lifetime (secs) : 75342
Encrypt Algorithm : DES
Encrypt Key Size : 64
Signature Hash Algorithm : HMAC_AUTH_SHA
Signature Key Length (bits) : 512
Signature Algorithm : SIG_ALG_RSA
```

 **NOTE**

The ESP-ENCRYPT-3DES-192, ESP-AUTH-SHA1, DES, HMAC_AUTH_SHA, and SIG_ALG_RSA algorithms have security risks; therefore, exercise caution when you use them.

----End

Configuration Files

- Configuration file of PE_1

```
#
sysname PE_1
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 172.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpna
  ip address 192.168.1.2 255.255.255.0
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpna
    import-route direct
    peer 192.168.1.1 as-number 65410
#
ospf 1
  area 0.0.0.0
    network 1.1.1.9 0.0.0.0
    network 172.1.1.0 0.0.0.255
#
return
```

- Configuration file of PE_2

```
#
sysname PE_2
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 200:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 172.1.1.2 255.255.255.0
```

```
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpna
ip address 192.168.2.2 255.255.255.0
#
interface GigabitEthernet3/0/0
ip binding vpn-instance vpna
ip address 192.168.3.2 255.255.255.0
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpna
import-route direct
peer 192.168.2.1 as-number 65430
peer 192.168.3.1 as-number 65420
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● Configuration file of GM_1

```
#
sysname GM_1
#
ike proposal 5
encryption-algorithm
aes-256
dh
group2
authentication-algorithm
sha2-256
authentication-method pre-
share
integrity-algorithm hmac-
sha2-256
prf hmac-sha2-256
#
ike peer spub
undo version 2
pre-shared-key cipher %^%#03uIP\`YNF+`AcJhbZ&C7y*iV100U@DraF58J4=;%^%#
ike-proposal 5
remote-address 192.168.3.1
#
ipsec policy map1 10 gdoi
group identity number 10
ike-peer spub
#
interface GigabitEthernet1/0/0
ip address 192.168.1.1 255.255.255.0
ipsec policy map1
#
interface GigabitEthernet2/0/0
```

```
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 192.168.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 192.168.1.2 enable
#
returnreturn
```

● Configuration file of GM_2

```
#
sysname GM_2
#
ike proposal 5
 encryption-algorithm
 aes-256
 dh
group2
 authentication-algorithm
 sha2-256
 authentication-method pre-
 share
 integrity-algorithm hmac-
 sha2-256
 prf hmac-sha2-256
#
ike peer spua
 undo version 2
 pre-shared-key cipher %^%#03uIP\ /YNF+`AcJhbZ&C7y*iv100U@DraF58J4=;%^%#
 ike-proposal 5
 remote-address 192.168.3.1
#
ipsec policy map2 10 gdoi
 group identity number 10
 ike-peer spua
#
interface GigabitEthernet1/0/0
 ip address 192.168.2.1 255.255.255.0
 ipsec policy map2
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
bgp 65430
peer 192.168.2.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 192.168.2.2 enable
#
return
```

7.8.2 Example for Configuring GM Link Redundancy

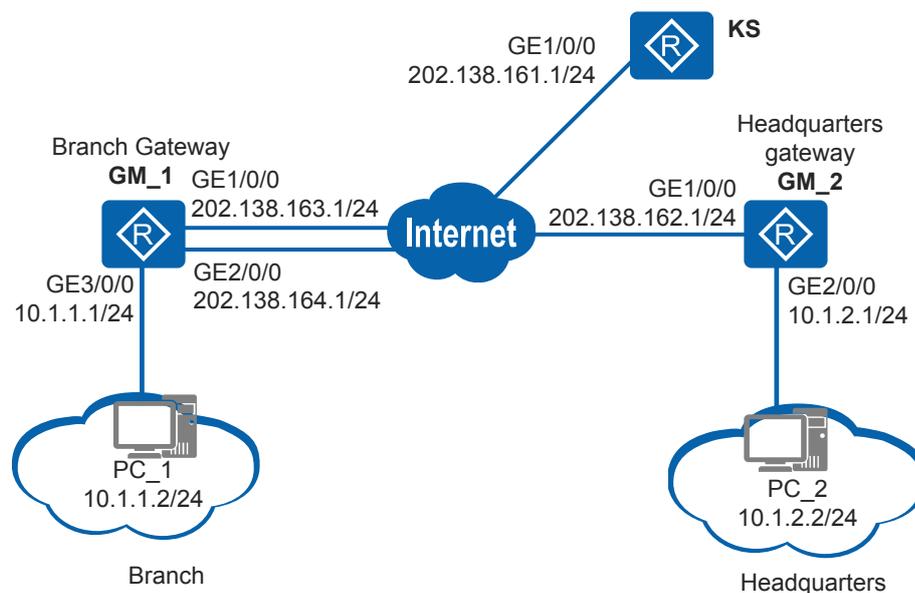
Networking Requirements

A large-scale enterprise has many branches distributed in a wide area and a large number of multicast services. As shown in [Figure 7-10](#), GM_1 is the enterprise branch gateway and GM_2 is the enterprise headquarters gateway. (The enterprise has only one branch in this example.) GM_1 uses two egress links in backup or load balancing mode to communicate with the headquarters over the public network.

It is required that traffic between the branch and headquarters be protected and services be transmitted securely when an active/standby switching occurs or one egress link becomes faulty.

A2A VPN can be deployed between the branch and headquarters to ensure secure communication within the enterprise. Meanwhile, GM link redundancy can be used to allow the two outbound interfaces on GM_1 to register with the KS and download the same group SA. GM_1 selects an appropriate outbound interface based on routing information to forward services. When any egress link fails, traffic is switched to the other link based on route convergence and the SA between group members is not changed.

Figure 7-10 Networking for configuring GM link redundancy



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the IP address and static route on each physical interface of the GM and KS to implement communication between interfaces.
2. Configure an ACL on the KS to define the data flows to be protected by A2A VPN.
3. Configure an IKE peer on the GMs and KS and define the attributes used for IKE negotiation.
4. Configure a security proposal on the KS to define the protection method used for A2A VPN.
5. Configure GDOI policies on the GMs and KS. You need to configure two GDOI policies on GM_1 and reference an IKE peer.
6. Configure an IP address for multicast Rekey messages on the GMs. The IP address must be the same as that configured on the KS.

7. Apply GDOI policy groups to the GM interfaces. You need to apply two GDOI policy groups to the two outbound interfaces on GM_1 and add the two interfaces to the same group to enable the A2A VPN protection function on the interfaces.

 **NOTE**

When configuring the KS, refer to the configuration guide for KSs from other vendors.

Procedure

- Step 1** Configure the IP address and static route on each physical interface to implement communication between interfaces.

Assign an IP address to each interface on GM_1.

```
<Huawei> system-view
[Huawei] sysname GM_1
[GM_1] interface gigabitethernet 1/0/0
[GM_1-GigabitEthernet1/0/0] ip address 202.138.163.1 255.255.255.0
[GM_1-GigabitEthernet1/0/0] quit
[GM_1] interface gigabitethernet 2/0/0
[GM_1-GigabitEthernet2/0/0] ip address 202.138.164.1 255.255.255.0
[GM_1-GigabitEthernet2/0/0] quit
[GM_1] interface gigabitethernet 3/0/0
[GM_1-GigabitEthernet3/0/0] ip address 10.1.1.1 255.255.255.0
[GM_1-GigabitEthernet3/0/0] quit
```

Configure static routes to the peer on GM_1.

```
[GM_1] ip route-static 202.138.161.0 255.255.255.0 202.138.163.2 preference 10
[GM_1] ip route-static 202.138.161.0 255.255.255.0 202.138.164.2 preference 20
[GM_1] ip route-static 202.138.162.0 255.255.255.0 202.138.163.2 preference 10
[GM_1] ip route-static 202.138.162.0 255.255.255.0 202.138.164.2 preference 20
[GM_1] ip route-static 10.1.2.0 255.255.255.0 202.138.163.2 preference 10
[GM_1] ip route-static 10.1.2.0 255.255.255.0 202.138.164.2 preference 20
```

Assign an IP address to each interface on GM_2.

```
<Huawei> system-view
[Huawei] sysname GM_2
[GM_2] interface gigabitethernet 1/0/0
[GM_2-GigabitEthernet1/0/0] ip address 202.138.162.1 255.255.255.0
[GM_2-GigabitEthernet1/0/0] quit
[GM_2] interface gigabitethernet 2/0/0
[GM_2-GigabitEthernet2/0/0] ip address 10.1.2.1 255.255.255.0
[GM_2-GigabitEthernet2/0/0] quit
```

Configure static routes to the peer on GM_2.

```
[GM_2] ip route-static 202.138.161.0 255.255.255.0 202.138.162.2
[GM_2] ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
[GM_2] ip route-static 202.138.164.0 255.255.255.0 202.138.162.2
[GM_2] ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
```

- Step 2** Configure an IKE peer for the two GMs. The IKE negotiation parameters must be the same as those on the KS.

Configure an IKE proposal on GM_1.

```
[GM_1] ike proposal 5
[GM_1-ike-proposal-5] quit
```

Create an IKE peer on GM_1.

```
[GM_1] ike peer spub
[GM_1-ike-peer-spub] undo version 2
[GM_1-ike-peer-spub] ike-proposal 5
[GM_1-ike-peer-spub] pre-shared-key cipher Huawei@1234
```

```
[GM_1-ike-peer-spua] remote-address 202.138.161.1  
[GM_1-ike-peer-spua] quit
```

Configure an IKE proposal on GM_2.

```
[GM_2] ike proposal 5  
[GM_2-ike-proposal-5] quit
```

Create an IKE peer on GM_2.

```
[GM_2] ike peer spua  
[GM_2-ike-peer-spua] undo version 2  
[GM_2-ike-peer-spua] ike-proposal 5  
[GM_2-ike-peer-spua] pre-shared-key cipher Huawei@1234  
[GM_2-ike-peer-spua] remote-address 202.138.161.1  
[GM_2-ike-peer-spua] quit
```

Step 3 Create GDOI policies for the GMs. The group ID of the GMs must be the same as that of the KS.

Configure GM_1.

```
[GM_1] ipsec policy map1 10 gdoi  
[GM_1-ipsec-policy-gdoi-map1-10] group identity number 10  
[GM_1-ipsec-policy-gdoi-map1-10] ike-peer spub  
[GM_1-ipsec-policy-gdoi-map1-10] tunnel local applied-interface  
[GM_1-ipsec-policy-gdoi-map1-10] quit  
[GM_1] ipsec policy map2 10 gdoi  
[GM_1-ipsec-policy-gdoi-map2-10] group identity number 10  
[GM_1-ipsec-policy-gdoi-map2-10] ike-peer spub  
[GM_1-ipsec-policy-gdoi-map2-10] tunnel local applied-interface  
[GM_1-ipsec-policy-gdoi-map2-10] quit
```

Configure GM_2.

```
[GM_2] ipsec policy map3 10 gdoi  
[GM_2-ipsec-policy-gdoi-map3-10] group identity number 10  
[GM_2-ipsec-policy-gdoi-map3-10] ike-peer spua  
[GM_2-ipsec-policy-gdoi-map3-10] quit
```

Step 4 Configure an IP address for multicast Rekey messages on the GMs. The IP address must be the same as that configured on the KS.

Configure GM_1.

```
[GM_1] multicast routing-enable  
[GM_1] ipsec gdoi multicast-rekey ip 239.0.1.2  
[GM_1] interface gigabitethernet 1/0/0  
[GM_1-GigabitEthernet1/0/0] pim dm  
[GM_1-GigabitEthernet1/0/0] igmp static-group 239.0.1.2  
[GM_1-GigabitEthernet1/0/0] quit  
[GM_1] interface gigabitethernet 2/0/0  
[GM_1-GigabitEthernet2/0/0] pim dm  
[GM_1-GigabitEthernet2/0/0] igmp static-group 239.0.1.2  
[GM_1-GigabitEthernet2/0/0] quit
```

Configure GM_2.

```
[GM_2] multicast routing-enable  
[GM_2] ipsec gdoi multicast-rekey ip 239.0.1.2  
[GM_2] interface gigabitethernet 1/0/0  
[GM_2-GigabitEthernet1/0/0] pim dm  
[GM_2-GigabitEthernet1/0/0] igmp static-group 239.0.1.2  
[GM_2-GigabitEthernet1/0/0] quit
```

Step 5 Apply a GDOI policy group to each interface on each GM.

Apply a GDOI policy group to each interface on GM_1.

```
[GM_1] interface gigabitethernet 1/0/0
[GM_1-GigabitEthernet1/0/0] ipsec policy map1
[GM_1-GigabitEthernet1/0/0] quit
[GM_1] interface gigabitethernet 2/0/0
[GM_1-GigabitEthernet2/0/0] ipsec policy map2
[GM_1-GigabitEthernet2/0/0] quit
```

Apply a GDOI policy group to the interface on GM_2.

```
[GM_2] interface gigabitethernet 1/0/0
[GM_2-GigabitEthernet1/0/0] ipsec policy map3
[GM_2-GigabitEthernet1/0/0] quit
```

Step 6 Verify the configuration.

After the configurations are complete, run the **display ike sa** command on each device to check the configuration. The command output of GM_1 is used as an example.

```
[GM_1] display ike sa
-----
Conn-ID   Peer           VPN   Flag(s)           Phase
-----
  1109    202.138.161.1  0     RD|ST              v1:1

Number of IKE SA : 1
-----

Flag Description:
RD--READY   ST--STAYALIVE  RL--REPLACED  FD--FADING   TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE   S--STANDBY    A--ALONE     NEG--NEGOTIATING
```

After the configurations are complete, run the **display ipsec gdoi-sa** command on each device to check the configuration. The command output of GM_1 is used as an example.

```
[GM_1] display ipsec gdoi-sa
=====
Interface: GigabitEthernet1/0/0
Path MTU: 1500
=====
-----
Gdoi policy name       : "map1"
Sequence number       : 10
-----

[TEK SA]
Protected vrf : 0
Protocol      : 0/permit
Flow source   : 10.1.1.0/255.255.255.0/0
Flow destination : 10.1.2.0/255.255.255.0/0

Inpacket count      : 0
Inpacket decap count : 0
Outpacket count     : 0
Outpacket encap count : 0
Inpacket drop count : 0
Outpacket drop count : 0

SA mode : normal
SPI: 99152831 (0x5e8f3bf)
Proposal : ESP-ENCRYPT-3DES-192 ESP-AUTH-SHA1
SA remaining lifetime(secs) : 65813
Anti-Replay(Time Based) : disable

[KEK POLICY]
Rekey Transport Type      : multicast
SPI: 0x2ad569a935d15b75174446fbb0feaf5b
Received rekey seqno     : 8
Lifetime (secs)         : 74737
Encrypt Algorithm        : DES
```

```

Encrypt Key Size      : 64
Signature Hash Algorithm : HMAC_AUTH_SHA
Signature Key Length (bits) : 512
Signature Algorithm   : SIG_ALG_RSA
  
```

Shut down GigabitEthernet1/0/0 on GM_1 and run the display ipsec gdoi-sa command to check the SA status. The command output shows that the SA is established with GigabitEthernet2/0/0 and traffic goes out from GigabitEthernet2/0/0; thereby, service continuity is ensured.

```

[GM_1] display ipsec gdoi-sa

=====
Interface: GigabitEthernet2/0/0
Path MTU: 1500
=====

-----
Gdoi policy name      : "map2"
Sequence number      : 10
-----

[TEK SA]
Protected vrf : 0
Protocol      : 0/permit
Flow source   : 10.1.1.0/255.255.255.0/0
Flow destination : 10.1.2.0/255.255.255.0/0

Inpacket count      : 0
Inpacket decap count : 0
Outpacket count     : 0
Outpacket encap count : 0
Inpacket drop count : 0
Outpacket drop count : 0

SA mode : normal
SPI: 99152831 (0x5e8f3bf)
Proposal : ESP-ENCRYPT-3DES-192 ESP-AUTH-SHA1
SA remaining lifetime(secs) : 65813
Anti-Replay(Time Based) : disable

[KEK POLICY]
Rekey Transport Type : multicast
SPI: 0x2ad569a935d15b75174446fbb0feaf5b
Lifetime (secs)      : 74737
Encrypt Algorithm    : DES
Encrypt Key Size     : 64
Signature Hash Algorithm : HMAC_AUTH_SHA
Signature Key Length (bits) : 512
Signature Algorithm   : SIG_ALG_RSA
  
```

 **NOTE**

The ESP-ENCRYPT-3DES-192, ESP-AUTH-SHA1, DES, HMAC_AUTH_SHA, and SIG_ALG_RSA algorithms have security risks; therefore, exercise caution when you use them.

----End

Configuration Files

- GM_1 configuration file

```

#
sysname GM_1
#
multicast routing-enable
#
ipsec gdoi multicast-rekey ip 239.0.1.2
#
ike proposal 5
encryption-algorithm
  
```

```
aes-256
 dh
group2
 authentication-algorithm
sha2-256
 authentication-method pre-
share
 integrity-algorithm hmac-
sha2-256
 prf hmac-sha2-256
#
ike peer spub
undo version 2
pre-shared-key cipher %^%#03uIP\YNF+`AcJhbZ&C7y*iV10OU@DraF58J4=;%^%#
ike-proposal 5
remote-address 202.138.161.1
#
ipsec policy map1 10 gdoi
group identity number 10
ike-peer spub
tunnel local applied-interface
#
ipsec policy map2 10 gdoi
group identity number 10
ike-peer spub
tunnel local applied-interface
#
interface GigabitEthernet1/0/0
 ip address 202.138.163.1 255.255.255.0
 pim dm
 igmp static-group 239.0.1.2
 ipsec policy map1
#
interface GigabitEthernet2/0/0
 ip address 202.138.164.1 255.255.255.0
 pim dm
 igmp static-group 239.0.1.2
 ipsec policy map2
#
interface GigabitEthernet3/0/0
 ip address 10.1.1.1 255.255.255.0
#
ip route-static 10.1.2.0 255.255.255.0 202.138.163.2 preference 10
ip route-static 10.1.2.0 255.255.255.0 202.138.164.2 preference 20
ip route-static 202.138.161.0 255.255.255.0 202.138.163.2 preference 10
ip route-static 202.138.161.0 255.255.255.0 202.138.164.2 preference 20
ip route-static 202.138.162.0 255.255.255.0 202.138.163.2 preference 10
ip route-static 202.138.162.0 255.255.255.0 202.138.164.2 preference 20
#
return
```

● GM_2 configuration file

```
#
sysname GM_2
#
multicast routing-enable
#
ipsec gdoi multicast-rekey ip 239.0.1.2
#
ike proposal 5
 encryption-algorithm
aes-256
 dh
group2
 authentication-algorithm
sha2-256
 authentication-method pre-
share
 integrity-algorithm hmac-
```

```
sha2-256
 prf hmac-sha2-256
#
ike peer spua
 undo version 2
 pre-shared-key cipher %%#03uIP\YNF+`AcJhbZ&C7y*iVl0OU@DraF58J4=;%%#
 ike-proposal 5
 remote-address 202.138.161.1
#
ipsec policy map3 10 gdoi
 group identity number 10
 ike-peer spua
#
interface GigabitEthernet1/0/0
 ip address 202.138.162.1 255.255.255.0
 pim dm
 igmp static-group 239.0.1.2
 ipsec policy map3
#
interface GigabitEthernet2/0/0
 ip address 10.1.2.1 255.255.255.0
#
ip route-static 10.1.1.0 255.255.255.0 202.138.162.2
ip route-static 202.138.161.0 255.255.255.0 202.138.162.2
ip route-static 202.138.163.0 255.255.255.0 202.138.162.2
ip route-static 202.138.164.0 255.255.255.0 202.138.162.2
#
return
```

7.9 Troubleshooting A2A VPN

7.9.1 GM Fails to Register with the KS

Fault Description

After a GM and a KS are configured, the GM fails to register with the KS.

Procedure

Step 1 Run the **display ike sa** command to check whether the first-stage IKE SA is successfully established.

- If no IKE SA is established, the IKE proposal or IKE peer configured on the GM differs from that on the KS. You can run the **display ike peer (all views)** or **display ike proposal (All views)** command to check the IKE proposal or IKE peer.

Change the IKE proposal or IKE peer configuration to the same as that on the KS. For details, see [7.6.1.1 Configuring IKE](#).

- If an IKE SA is established successfully, go to Step 2.

Step 2 Run the **display ipsec gdoi-policy** command to check whether the GM and KS have the same group ID.

- If the group ID of the GM differs from that of the KS, change them to the same. For details, see [7.6.1.3 Configuring a GDOI Policy](#).
- If the GM and KS have the same group ID, check whether other related configurations are correct.

----End

7.10 A2A VPN FAQ

7.10.1 Why Some Service Packets Are Lost After A2A VPN Is Deployed?

After A2A VPN is deployed, A2A VPN packets are discarded if their size exceeds the interface MTU but the DF flag is not set to 0. To solve the problem, run the **ipsec df-bit clear** command to enable fragmentation of A2A VPN packets.

7.11 References for A2A VPN

The following table lists the references for A2A VPN.

Table 7-2 References for A2A VPN

Document	Description
RFC2401	Security Architecture for the Internet Protocol
RFC2402	IP Authentication Header
RFC2403	The Use of HMAC-MD5-96 within ESP and AH
RFC2404	The Use of HMAC-SHA-1-96 within ESP and AH
RFC2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406	IP Encapsulating Security Payload (ESP)
RFC2407	The Internet IP Security Domain of Interpretation for ISAKMP
RFC2408	Internet Security Association and Key Management Protocol (ISAKMP)
RFC2409	The Internet Key Exchange (IKE)
RFC2410	The NULL Encryption Algorithm and Its Use With IPsec
RFC3706	A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
RFC6407	The Group Domain of Interpretation
RFC3740	The Multicast Group Security Architecture
RFC3947	Negotiation of NAT-Traversal in the IKE
RFC3948	UDP Encapsulation of IPsec ESP Packets
RFC4301	Security Architecture for the Internet Protocol

Document	Description
RFC4302	IP Authentication Header
RFC4303	IP Encapsulating Security Payload (ESP)
RFC4308	Cryptographic Suites for IPsec
RFC4835	Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC4945	The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
RFC5374	Multicast Extensions to the Security Architecture for the Internet Protocol
draft-dukes-ike-mode-cfg-02.txt	The ISAKMP Configuration Method

8 BGP/MPLS IP VPN Configuration

About This Chapter

An enterprise can build its own BGP/MPLS IP VPN network to implement secure interconnection between its headquarters and branches. The BGP/MPLS IP VPN network ensures high-quality communication within the enterprise network.

[8.1 Overview of BGP/MPLS IP VPN](#)

This section describes the definition, background, and functions of BGP/MPLS IP VPN.

[8.2 Understanding BGP/MPLS IP VPN](#)

This section describes the implementation of BGP/MPLS IP VPN.

[8.3 Application Scenarios for BGP/MPLS IP VPN](#)

This section describes the application scenarios for BGP/MPLS IP VPN.

[8.4 Summary of BGP/MPLS IP VPN Configuration Tasks](#)

After basic BGP/MPLS IP VPN configurations are complete, a simple VPN network can be established using MPLS technology. To deploy special BGP/MPLS IP VPN networking, perform other configuration tasks according to the reference sections provided in the following table.

[8.5 Licensing Requirements and Limitations for BGP/MPLS IP VPN](#)

[8.6 Default Settings for BGP/MPLS IP VPN](#)

This section describes the default settings for BGP/MPLS IP VPN.

[8.7 Configuring BGP/MPLS IP VPN](#)

This section describes the procedures for configuring BGP/MPLS IP VPN functions.

[8.8 Maintaining BGP/MPLS IP VPN](#)

You can check route summary information in a VPN instance, monitor network connectivity, and reset BGP connections when maintaining a BGP/MPLS IP VPN network.

[8.9 Configuration Examples for BGP/MPLS IP VPN](#)

This section provides several configuration examples of BGP/MPLS IP VPN networking. In each configuration example, the networking requirements, configuration roadmap, configuration procedures, and configuration files are provided.

[8.10 FAQ About BGP/MPLS IP VPN](#)

This section describes the FAQ about BGP/MPLS IP VPN.

8.11 References for BGP/MPLS IP VPN

This section lists references for BGP/MPLS IP VPN.

8.1 Overview of BGP/MPLS IP VPN

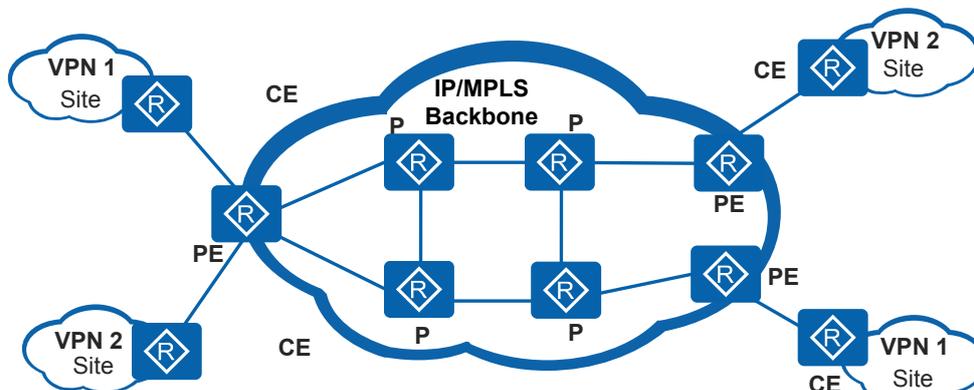
This section describes the definition, background, and functions of BGP/MPLS IP VPN.

Definition

A BGP/MPLS IP VPN is a Layer 3 virtual private network (L3VPN). A BGP/MPLS IP VPN uses the Border Gateway Protocol (BGP) to advertise VPN routes and uses Multiprotocol Label Switching (MPLS) to forward VPN packets on backbone networks. Here, IP that Internet Protocol (IP) packets are carried by the VPN.

Figure 8-1 shows the BGP/MPLS IP VPN model.

Figure 8-1 BGP/MPLS IP VPN model



The BGP/MPLS IP VPN model consists of the following entities:

- Customer Edge (CE): a device that is deployed at the edge of a customer network and has interfaces directly connected to the service provider (SP) network. A CE device can be a router, a switch, or a host. Generally, CE devices do not detect VPNs and do not need to support MPLS.
- Provider Edge (PE): a device that is deployed at the edge of an SP network and directly connected to a CE device. On an MPLS network, PE devices process all VPN services and must have high performance.
- Provider (P): a backbone device that is deployed on an SP network and is not directly connected to CE devices. P devices only need to provide basic MPLS forwarding capabilities and do not maintain VPN information.

PE and P devices are managed by SPs. CE devices are managed by customers unless customers authorize SPs to manage their CE devices.

A PE device can connect to multiple CE devices. A CE device can connect to multiple PE devices of the same SP or different SPs.

Purpose

A traditional VPN sets up full-mesh tunnels or permanent virtual circuits (PVCs) between all sites to forward VPN data. This method makes networks difficult to maintain and expand. When a new site is added to an established VPN, a network administrator must modify the configuration of all edge nodes connected to this site.

A BGP/MPLS IP VPN uses a peer model that enables SPs and customers to exchange routing information. The SPs are responsible for forwarding data of customers, without participation of the customers. A BGP/MPLS IP VPN is more scalable and more easier to manage than a traditional VPN. When a new site is added, a network administrator only needs to modify the configuration of the edge nodes serving the new site.

BGP/MPLS IP VPN allows overlapping address spaces and overlapping VPNs so that VPNs can be flexibly deployed and expanded. In addition, BGP/MPLS IP VPN supports MPLS Traffic Engineering (TE). Because of these merits, BGP/MPLS IP VPN becomes an important approach for IP network carriers to provide value-added services and is now widely used.

8.2 Understanding BGP/MPLS IP VPN

This section describes the implementation of BGP/MPLS IP VPN.

8.2.1 Concepts

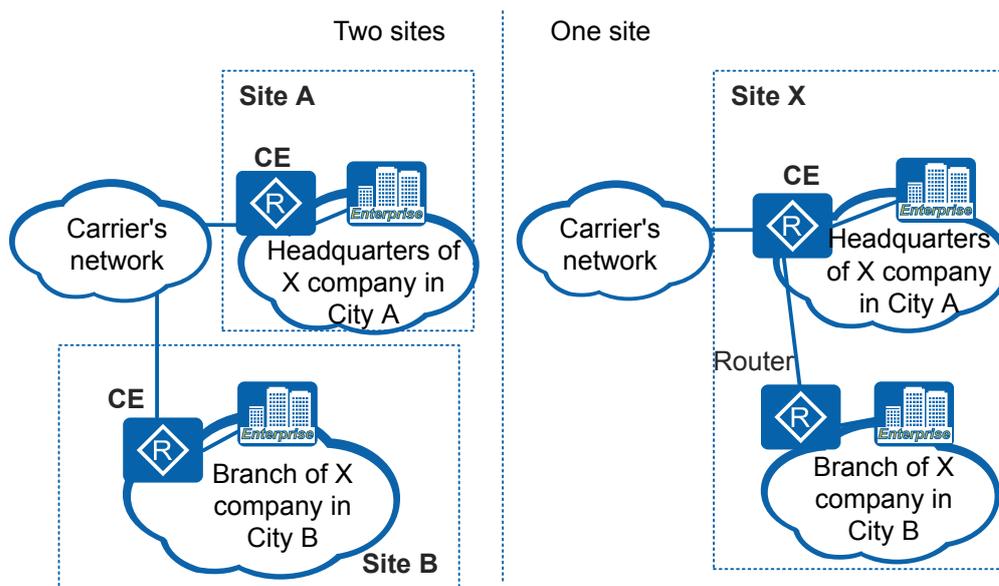
Site

The site is frequently mentioned in VPN technology. The following describes a site from different aspects:

- A site is a group of IP systems with IP connectivity, which can be achieved independent of SP networks.

Figure 8-2 shows an example of sites. On the networks on the left side in **Figure 8-2**, the headquarters of company X in city A is a site, and the branch of company X in city B is another site. IP devices can communicate within each site without using the carrier network.

Figure 8-2 Sites



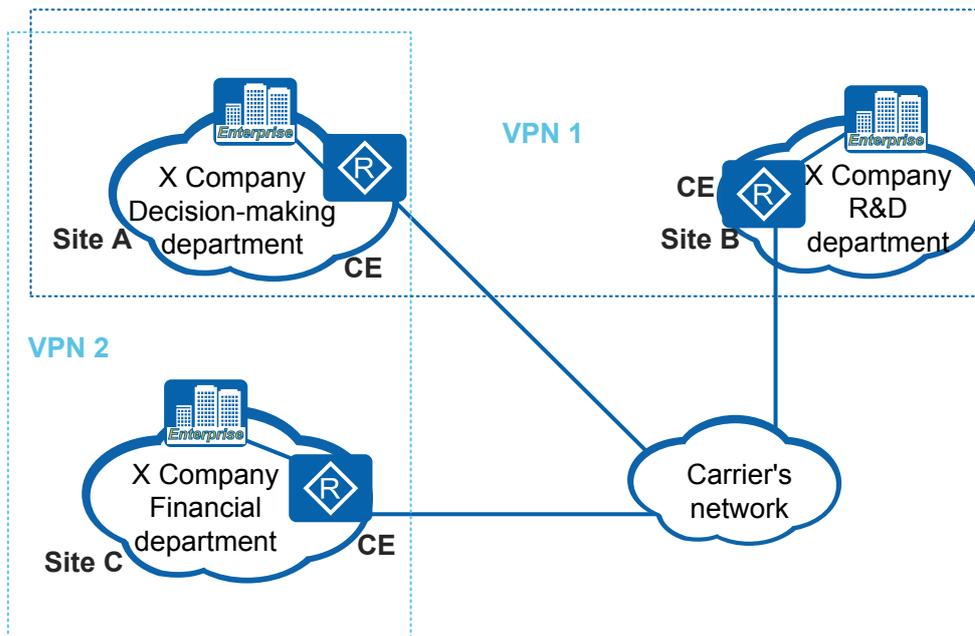
- Sites are configured based on topologies between devices but not their geographic locations, although devices in a site are geographically adjacent to each other in most cases. Two geographically separated IP systems can also compose a site if they are connected through leased lines and can communicate without the use of the carrier network.

On the right of **Figure 8-2**, the branch network in city B connects to the headquarters network in city A through leased lines but not a carrier network. The branch network and the headquarters network compose a site.

- The devices in a site may belong to multiple VPNs. That is, a site may belong to more than multiple VPNs.

As shown in **Figure 8-3**, the decision-making department of company X in city A (Site A) is allowed to communicate with the R&D department in city B (Site B) and the financial department in city C (Site C). Site B and Site C are not allowed to communicate with each other. In this case, two VPNs, VPN1 and VPN2, can be established. Site A and Site B belong to VPN1; Site A and Site C belong to VPN2. Site A belongs to two VPNs.

Figure 8-3 One site belonging to multiple VPNs



- A site connects to a carrier network through CE devices. A site may have more than one CE device, but a CE device belongs to only one site.
CE devices are selected according to sites:
If a site is a host, the host is the CE device of the site.
If a site is a subnet, switches are used as CE devices.
If a site has multiple subnets, routers are used as CE devices.
Sites connected to the same carrier network can be grouped into different sets using policies. Only sites that belong to the same set can communicate with each other through the carrier network. Such a set is a VPN.

Address Space Overlapping

As a private network, each VPN manages an address space. Address spaces of different VPNs may overlap. For example, if both VPN1 and VPN2 use addresses on the network segment 10.110.10.0/24, their address spaces overlap.

VPNs can use overlapping address spaces in the following situations:

- Two VPNs do not cover the same site.
- Two VPNs cover the same site, but devices in the site do not need to communicate with devices using overlapping address spaces in the VPNs.

VPN Instance

In BGP/MPLS IP VPN implementation, routes of different VPNs are isolated by VPN instances.

A PE device establishes and maintains a VPN instance for each directly connected site. A VPN instance contains VPN member interfaces and routes of the corresponding site.

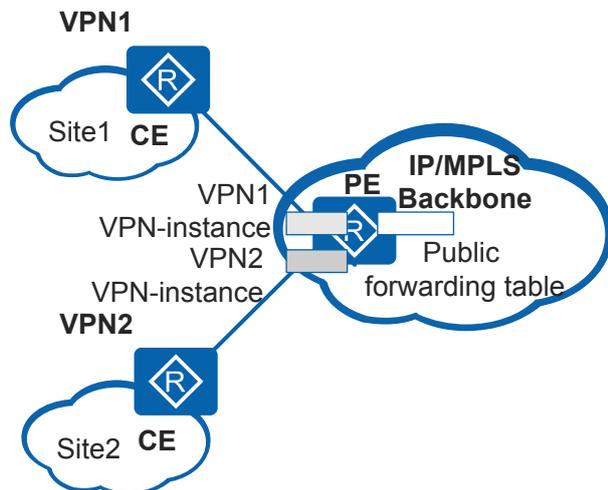
Specifically, information in a VPN instance includes the IP routing table, label forwarding table, interface bound to the VPN instance, and VPN instance management information. VPN instance management information includes the route distinguisher (RD), route filtering policy, and member interface list of the VPN instance.

The relationships between VPNs, sites, and VPN instances are as follows:

- A VPN consists of multiple sites. A site may belong to multiple VPNs.
- A site is associated with a VPN instance on a PE device. A VPN instance integrates VPN members and routing policies of associated sites. Multiple sites compose a VPN based on rules of the VPN instance.
- VPN instances are not mapped to VPNs on a one-to-one basis, whereas VPN instances are mapped to sites on a one-to-one basis.

A VPN instance is also called a VPN routing and forwarding table (VRF). A PE device has multiple routing and forwarding tables, including a public routing and forwarding table and one or more VRFs. [Figure 8-4](#) shows VPN instances.

Figure 8-4 VPN instances



A public routing and forwarding table and a VRF differ in the following aspects:

- A public routing table contains IPv4 routes of all the PE and P devices. The routes are static routes or dynamic routes generated by routing protocols on the backbone network.
- A VPN routing table contains routes of all sites that belong to a VPN instance. The routes are obtained through the exchange of VPN routing information between PE devices or between CE and PE devices.
- Information in a public forwarding table is extracted from the public routing table according to route management policies, whereas information in a VPN forwarding table is extracted from the corresponding VPN routing table.

VPN instances on a PE device are independent of each other and maintain a VRF independent of the public routing and forwarding table.

Each VPN instance can be considered as a virtual device, which maintains an independent address space and connects to VPNs through interfaces.

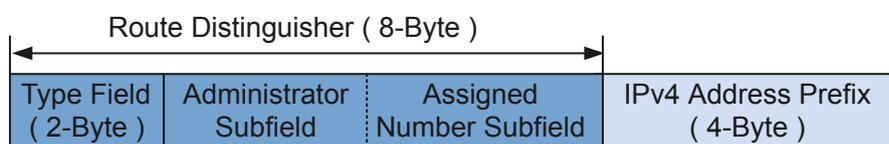
RD and VPN-IPv4 Address

Traditional BGP cannot process VPN routes with overlapping address spaces. For example, VPN1 and VPN2 use addresses on the network segment 10.110.10.0/24, and they each advertise a route to this network segment. The local PE device can identify routes based on VPN instances. However, when the routes are advertised to the remote PE device, BGP selects only one of the two routes because load balancing is not performed between routes of different VPNs. The other route is lost.

To address the preceding problem, PE devices use Multiprotocol Extensions for BGP-4 (MP-BGP) to advertise VPN routes and use the VPN-IPv4 address.

A VPN-IPv4 address has 12 bytes. The first eight bytes represent the RD, and the last four bytes represent the IPv4 address prefix, as shown in [Figure 8-5](#).

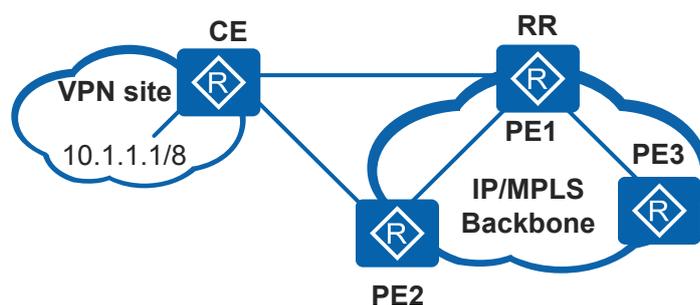
Figure 8-5 VPN-IPv4 address



RDs distinguish IPv4 prefixes with the same address space. IPv4 addresses with RDs are VPN-IPv4 addresses (VPNv4 addresses). After receiving IPv4 routes from a CE device, a PE device converts the routes into globally unique VPN-IPv4 routes and advertises the routes on the public network.

SPs can allocate RDs independently because of the RD format. When CE devices are dual-homed to PE devices, the RD must be globally unique to ensure correct routing. As shown in [Figure 8-6](#), a CE device is dual-homed to PE1 and PE2. PE1 also functions as a route reflector (RR).

Figure 8-6 Networking diagram of CE dual-homing



PE1 is an edge device of the backbone network and advertises a VPN-IPv4 route with the IPv4 prefix 10.1.1.1/8 to PE3. PE1 also functions as an RR and reflects a VPN-IPv4 route with the IPv4 prefix 10.1.1.1/8 from PE2 to PE3.

- If the VPN has the same RD on PE1 and PE2, the two VPN-IPv4 routes to 10.1.1.1/8 have the same destination address. Therefore, PE3 receives only one VPN-IPv4 route (CE -> PE1 -> PE3) to 10.1.1.1/8 from PE1. When the direct link between PE1 and CE

becomes faulty, PE3 deletes the VPN-IPv4 route to 10.1.1.1/8. As a result, VPN data destined for 10.1.1.1/8 cannot be forwarded to the destination. Actually, PE3 has another route to 10.1.1.1/8, PE3 -> PE1 -> PE2 -> CE.

- If the VPN has the same RD on PE1 and PE2, the two VPN-IPv4 routes to 10.1.1.1/8 have different destination addresses. Therefore, PE3 receives two VPN-IPv4 route to 10.1.1.1/8 from PE1. When any link between PE1 and CE becomes faulty, PE3 deletes the corresponding route and reserves the other one. Data destined for 10.1.1.1/8 can still be correctly forwarded.

VPN Target

A VPN target, also called the route target (RT), is a BGP extension community attribute. BGP/MPLS IP VPN uses VPN targets to control VPN routes advertisement.

A VPN instance is associated with one or more VPN target attributes. VPN target attributes are classified into the following types:

- **Export target:** After a PE device learns IPv4 routes from directly connected sites, it converts the routes to VPN-IPv4 routes and sets the export target attribute for those routes. The export target attribute is advertised with the routes as a BGP extended community attribute.
- **Import target:** After a PE device receives VPN-IPv4 routes from other PE devices, it checks the export target attribute of the routes. If the export target is the same as the import target of a VPN instance on the local PE device, the local PE device adds the route to the VPN routing table.

BGP/MPLS IP VPN uses VPN targets to control advertisement and receiving of VPN routes between sites. VPN export targets are independent of import targets. An export target and an import target can be configured with multiple values to implement flexible VPN access control and VPN networking.

For example, if the import target of a VPN instance contains 100:1, 200:1, and 300:1, any route with the export target of 100:1, 200:1, or 300:1 is added to the routing table of the VPN instance.

8.2.2 Implementation

This section describes BGP/MPLS IP VPN implementation:

- [VPN Label Distribution](#)
- [VPN Route Cross](#)
- [Public Network Tunnel Iteration](#)
- [VPN Route Selection Rules](#)
- [Route Advertisement in BGP/MPLS IP VPN](#)
- [Packet Forwarding in BGP/MPLS IP VPN](#)

VPN Label Distribution

Before advertising private routes to other PE devices on the backbone network through MP-BGP, a PE device must assign MPLS labels (VPN label) to the private routes. Packets transmitted over the backbone network carry MPLS labels.

A PE device allocates MPLS labels in either of the following ways:

- One label per route
Each route in a VRF is assigned one label. When a large number of routes exist on the network, the Incoming Label Map (ILM) maintains a large number of entries, which requires high router capacity.
- One label per instance
Each VPN instance is assigned one label. All the routes of a VPN instance share the same label, saving labels.

 **NOTE**

MP-BGP can allocate labels to private routes only after MPLS is enabled on the PE device.

VPN Route Cross

The routes exchanged between two PE devices through MP-BGP are VPNv4 routes. A PE device checks received VPNv4 routes and drops the following routes:

- VPNv4 routes with unreachable next hops
- VPNv4 routes received from an RR with the cluster_id of the PE device in the cluster_list
- VPNv4 routes that are denied by the BGP routing policy

The PE device matches the remaining routes with the Import Targets of VPN instances. The matching process is called VPN route cross.

Some routes sent from local CE devices belong to different VPNs. The PE device also matches these routes with Import Targets of local VPN instances if these routes have reachable next hops or can be iterated. The matching process is called local VPN route cross. For example, CE1 resides in a site of VPN1, and CE2 resides in a site of VPN2. Both CE1 and CE2 connect to PE1. When PE1 receives routes of VPN1 from CE1, PE1 also matches the routes with the Import Target of the instance of VPN2.

 **NOTE**

To correctly forward a packet, a BGP-enabled device must find out a directly reachable address, through which the packet can be forwarded to the next hop in the routing table. The route to the directly reachable address is called dependent route, because BGP guides packet forwarding based on the route. The process of finding a dependent route based on the next-hop address is called route iteration.

Public Network Tunnel Iteration

To transmit traffic of private networks across a public network, tunnels need to be established on the public network. After VPN route cross is complete, PE devices perform route iteration based on destination IPv4 prefixes to find the appropriate tunnels (except for local cross routes). Then tunnel iteration is performed. The routes are injected into the VPN routing table only after tunnel iteration succeeds. The process of iterating routes to corresponding tunnels is called tunnel iteration.

After tunnel iteration succeeds, tunnel IDs are reserved for subsequent packet forwarding. A tunnel ID identifies a tunnel. In VPN packet forwarding, the PE devices search for tunnels based on tunnel IDs.

VPN Route Selection Rules

Not all the cross routes processed by tunnel iteration are installed to VPN routing tables. Similarly, not all the routes received from the local CE devices and the local cross routes are injected into VPN routing tables.

When multiple routes to the same destination are available, a PE device selects one route based on the following rules if load balancing is not configured:

- If a route received from a local CE device and a cross route are destined to the same destination, the PE device selects the route received from the local CE device.
- If a local cross route and a cross route received from another PE device are destined for the same destination, the PE device selects the local cross route.

If load balancing is configured, the PE device selects one route based on the following rules:

- Preferentially selects the route from the local CE device. When one route from the local CE device and multiple cross routes exist, the PE device selects the route from the local CE device.
- Performs load balancing between the routes from the local CE device or between the cross routes. The PE device does not perform load balancing between the routes from the local CE device and the cross routes.
- The AS_Path attributes of the routes participating in load balancing must be the same.

Route Advertisement in BGP/MPLS IP VPN

In basic BGP/MPLS IP VPN application, CE and PE devices are responsible for advertising VPN routes, whereas P devices only need to maintain routes of the backbone network without knowing VPN routes. Generally, PE devices maintain all VPN routes.

VPN routes are advertised from the local CE device to the ingress PE device, from the ingress PE device to the egress PE device, and from the egress PE device to the remote CE device. After the whole route advertisement process is complete, the local and remote CE devices have reachable routes to each other, and VPN routes can be advertised on the backbone network.

The route advertisement process is as follows:

- Route advertisement from the local CE device to the ingress PE device
After a neighbor or peer relationship is set up between a CE device and the directly connected PE device, the CE device advertises the local IPv4 routes to the PE device. The CE and PE devices can use static routes, the Routing Information Protocol (RIP), the Open Shortest Path First (OSPF) protocol, the Intermediate System-to-Intermediate System (IS-IS) protocol, or BGP (Border Gateway Protocol). No matter which routing protocol is used, the routes advertised by the CE device to the PE device are standard IPv4 routes.
- Route advertisement from the ingress PE device to the egress PE device
 - After learning VPN routes from a CE device, the egress PE device adds RDs to standard IPv4 routes. The routes are changed into VPN-IPv4 routes.
 - The ingress PE device advertises the MP-BGP Update messages containing VPN-IPv4 routes to the egress PE device. The Update messages contain Export Targets and MPLS labels.
 - When the egress PE device receives the VPN-IPv4 routes and if the next hops are reachable, it performs VPN route cross, tunnel iteration, and route selection to

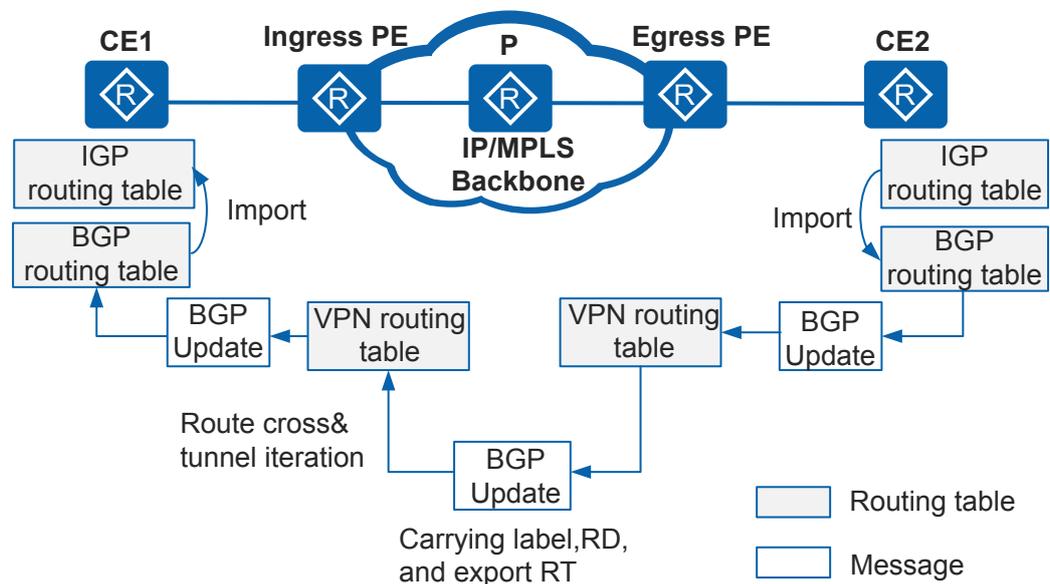
determine whether to inject the routes into the VRF. For the routes added to the VPN routing table, the local PE stores the tunnel IDs and MPLS labels carried in MP-BGP Update messages for subsequent packet forwarding.

- Route advertisement from the egress PE device to the remote CE device

The remote CE device can learn VPN routes from the egress PE device through static routes, RIP, OSPF, IS-IS, or BGP. Route advertisement from the egress PE device to the remote CE device is the same as that from the local CE device to the ingress PE device. The routes advertised by the egress PE device to the remote CE device are standard IPv4 routes.

Figure 8-7 shows route advertisement from CE2 to CE1. In this example, BGP runs between CE and PE devices, and LSPs are used.

Figure 8-7 Route advertisement from CE2 to CE1



1. Interior Gateway Protocol (IGP) routes are imported into the BGP IPv4 unicast address family of CE2.
2. CE2 advertises an EBGP Update message with routing information to the egress PE device. After receiving the message, the egress PE device converts the route to a VPN-IPv4 route, and then installs the route to the VPN routing table.
3. The egress PE device allocates an MPLS label to the route. Then it adds the label and VPN-IPv4 routing information to the NLRI field and the export target to the extended community attribute field of the MP-IBGP Update message. After that, the egress PE device sends the Update message to the ingress PE device.
4. After receiving the message, the ingress PE device performs VPN route cross. After the VPN route cross succeeds, the ingress PE device performs tunnel iteration based on the destination IPv4 address to find the appropriate tunnel. If tunnel iteration succeeds, the ingress PE device stores the tunnel ID and label, and then adds the route to the VPN routing table of the VPN instance.
5. The ingress PE device advertises a BGP Update message with the route to CE1. The advertised route is an IPv4 route.

- After receiving the route, CE1 installs the route to the BGP routing table. CE1 can import the route to the IGP routing table by importing BGP routes to IGP.

To ensure that CE1 and CE2 can communicate, CE1 also needs to advertise routes to CE2, of which the process is similar to the preceding process.

Packet Forwarding in Basic BGP/MPLS IP VPN

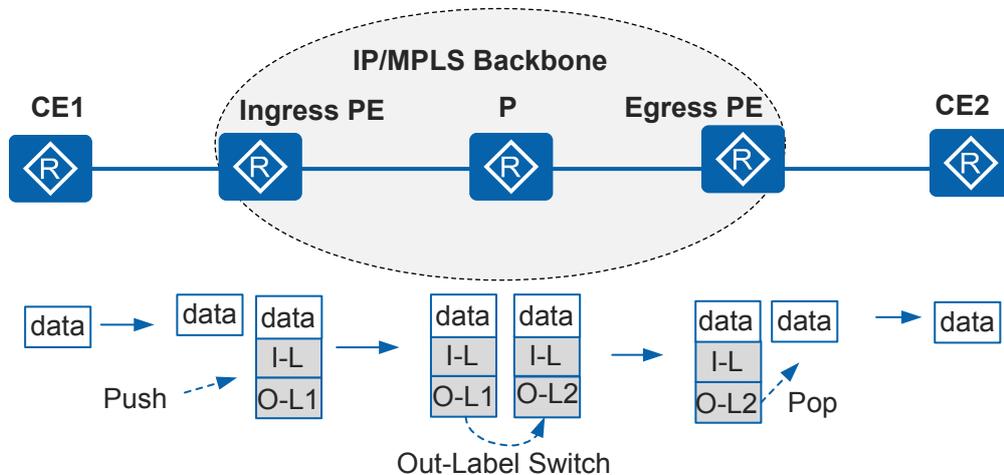
In basic BGP/MPLS IP VPN applications (excluding inter-AS VPN), VPN packets are forwarded with double labels:

- Outer label (public network label): is swapped on the backbone network, identifies an LSP from a PE device to a remote PE device, and enables VPN packets to reach the remote PE device through the LSP.
- Inner label (VPN label): is used when VPN packets are sent from the remote PE device to a CE device, and identifies the site (or specifically, the CE device) to which VPN packets are sent. The remote PE device finds the outbound interface for VPN packets according to the inner label.

If two sites of a VPN connect to the same PE device, the PE device only needs to know how VPN packets can reach the remote CE device.

Figure 8-8 shows packet forwarding from CE1 to CE2. In **Figure 8-8**, I-L indicates an inner label, and O-L indicates an outer label.

Figure 8-8 Forwarding of a VPN packet from CE1 to CE2



- CE1 sends a VPN packet.
- After receiving the packet on the interface bound to a VPN instance, the ingress PE device processes the packet as follows:
 - Searches for the corresponding VPN forwarding table based on the RD of the VPN instance.
 - Matches the destination IPv4 prefix to find the corresponding tunnel ID.
 - Adds I-L to the packet and finds the tunnel based on the tunnel ID.
 - Sends the packet through the tunnel and adds O-L1 to the packet.

Then the packet travels across the backbone network with double MPLS labels. Each P device on the backbone network swaps the outer label of the packet.

3. After receiving the packet with double labels, the egress PE device delivers the packet to MPLS for processing. MPLS pops the outer label. In this example, the final outer label of the packet is O-L2. If the PHP function is configured, the outer label is popped on the hop before the egress PE device, and the egress PE device receives the packet with only the inner label.
4. At this time, the egress PE device can only identify the inner label. Finding the label is at the bottom of the label stack, and the egress PE device pops the inner label.
5. The egress PE device sends the packet to CE2. At this time, the packet is an IP packet. The packet is successfully transmitted from CE1 to CE2. CE2 transmits the packet to the destination according to the IP forwarding process.

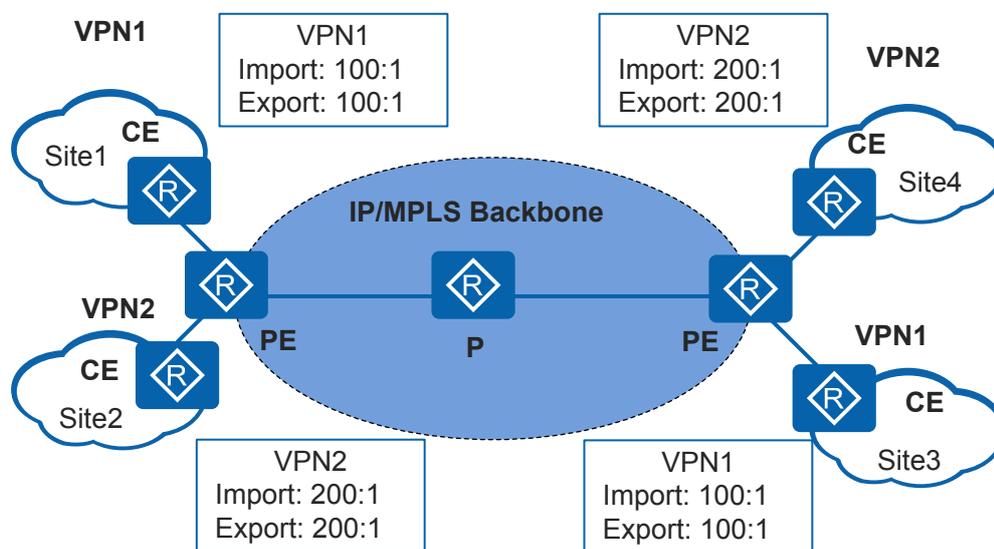
8.2.3 Basic Networking

Intranet VPN

In an intranet VPN, all the users in the VPN can transmit packets to each other, but cannot communicate with users outside the VPN. The sites within an intranet VPN usually belong to the same organization.

In intranet VPN networking, each VPN is allocated a VPN target as the export target and import target. The VPN target of a VPN cannot be used by other VPNs.

Figure 8-9 Intranet VPN networking



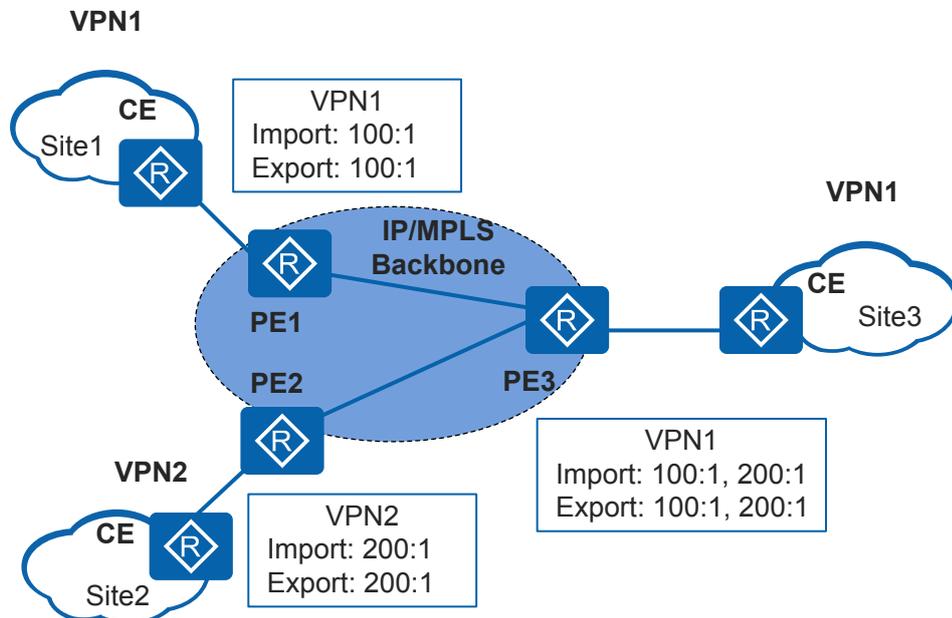
As shown in [Figure 8-9](#), PE devices allocate the VPN target 100:1 to VPN1 and the target 200:1 to VPN2. The two sites in the same VPN can communicate with each other, whereas sites in different VPNs cannot communicate.

Extranet VPN

If users in a VPN need to access some sites of another VPN, extranet networking can be used.

In extranet networking, if a VPN needs to access a shared site, its export target must be included in the import target of the VPN instance covering the shared site, and its import target must be included in the export target of the VPN instance covering the shared site.

Figure 8-10 Extranet VPN networking



As shown in **Figure 8-10**, VPN1 and VPN2 can access Site3 of VPN1.

- PE3 can receive VPN-IPv4 routes advertised by PE1 and PE2.
- PE1 and PE2 can receive VPN-IPv4 routes advertised by PE3.

Site1 and Site3 of VPN1 can communicate with each other. Site2 of VPN2 and Site3 of VPN1 communicate with each other.

PE3 does not advertise the VPN-IPv4 routes learned from PE1 to PE2 and does not advertise the VPN-IPv4 routes learned from PE2 to PE1. Therefore, Site1 of VPN1 and Site2 of VPN2 cannot communicate with each other.

Hub and Spoke

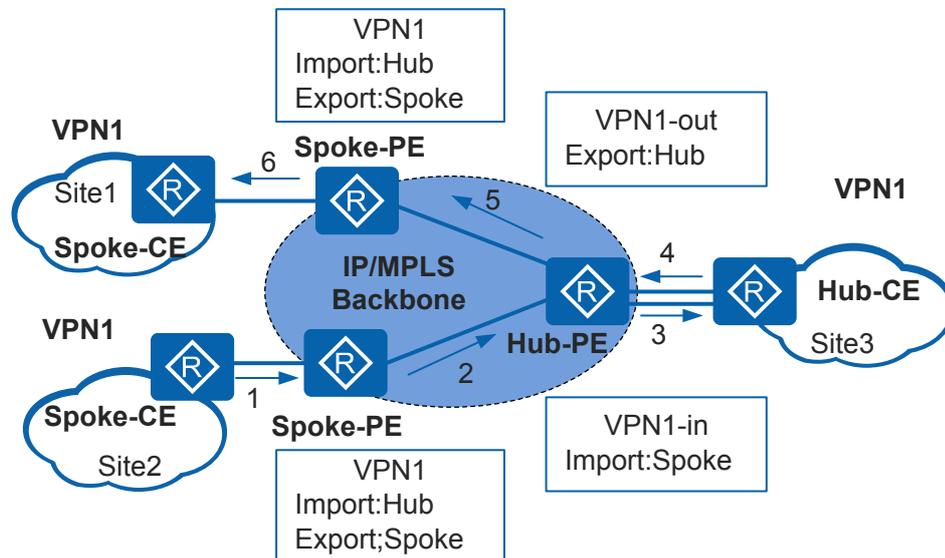
If a central access control device needs to be deployed to control communication between VPN users, the Hub and Spoke networking can be used. The site with the access control device deployed is the Hub site, and other sites are Spoke sites. The following devices are used in Hub and Spoke networking:

- Hub-CE: is deployed in the Hub site and connected to the VPN backbone network.
- Spoke-CE: is deployed in a Spoke site and connected to the VPN backbone network.
- Hub-PE: is deployed on the VPN backbone network and connected to the Hub site.
- Spoke-PE: is deployed on the VPN backbone network and connected to a Spoke site.

A Spoke site advertises routes to the Hub site, and then the Hub site advertises the routes to other Spoke sites. Spoke sites do not advertise routes to each other. The Hub site controls communication between all Spoke sites.

In Hub and Spoke networking, two VPN targets are configured to represent Hub and Spoke respectively. **Figure 8-11** shows the Hub and Spoke networking.

Figure 8-11 Hub and Spoke networking



The VPN targets of a PE device must comply with the following rules:

- The export target and import target of a Spoke-PE device are Spoke and Hub respectively. The import target of any Spoke-PE device must be different from the export target of any other Spoke-PE device.
- A Hub-PE device requires two interfaces or sub-interfaces.
 - One interface or sub-interface receives routes from Spoke-PE devices. The import target of the VPN instance on the interface is Spoke.
 - The other interface or sub-interface advertises routes to Spoke-PE devices. The export target of the VPN instance on the interface is Hub.

As shown in **Figure 8-11**, the Hub site controls communication between Spoke sites. The arrows show the process of advertising a route from Site2 to Site1:

- The Hub-PE device can receive VPN-IPv4 routes advertised by all the Spoke-PE devices.
- All the Spoke-PE devices can receive VPN-IPv4 routes advertised by the Hub-PE.
- The Hub-PE device advertises the routes learned from Spoke-PE devices to the Hub-CE device, and advertises the routes learned from the Hub-CE device to all the Spoke-PE devices. By doing this, the Spoke sites can access each other through the Hub site.
- The import target of any Spoke-PE device is different from the export targets of other Spoke-PE devices. Therefore, any two Spoke-PE devices do not directly advertise VPN-IPv4 routes to each other. The Spoke sites cannot directly communicate with each other.

8.2.4 Inter-AS VPN

The MPLS VPN solution is widely used, serving an increasing number of users in a large number of applications. As more sites are developed in an enterprise, a site at one

geographical location often needs to connect to an ISP network at another geographical location. Consider, for example, the inter-AS issue facing operators who manage different metropolitan area networks (MANs) or backbone networks that span different autonomous systems (AS).

Generally, MPLS VPN architecture runs within an AS. Routes of any VPN can be flooded within the AS, and cannot be flooded to other ASs. To implement exchange of VPN routes between different ASs, the inter-AS MPLS VPN model is used. The inter-AS MPLS VPN model is an extension to MPLS VPN framework. Through this model, route prefixes and labels can be advertised over links between different carrier networks.

RFC 4364 defines the following inter-AS VPN solutions:

- Inter-Provider Backbones Option A: Autonomous system boundary routers (ASBRs) manage VPN routes for inter-AS VPNs through dedicated interfaces. This solution is also called VRF-to-VRF.
- Inter-Provider Backbones Option B: ASBRs advertise labeled VPN-IPv4 routes to each other through MP-EBGP. This solution is also called EBGP redistribution of labeled VPN-IPv4 routes.
- Inter-Provider Backbones Option C: PE devices advertise labeled VPN-IPv4 routes to each other through Multi-hop MP-EBGP. This solution is also called Multi-hop EBGP redistribution of labeled VPN-IPv4 routes.

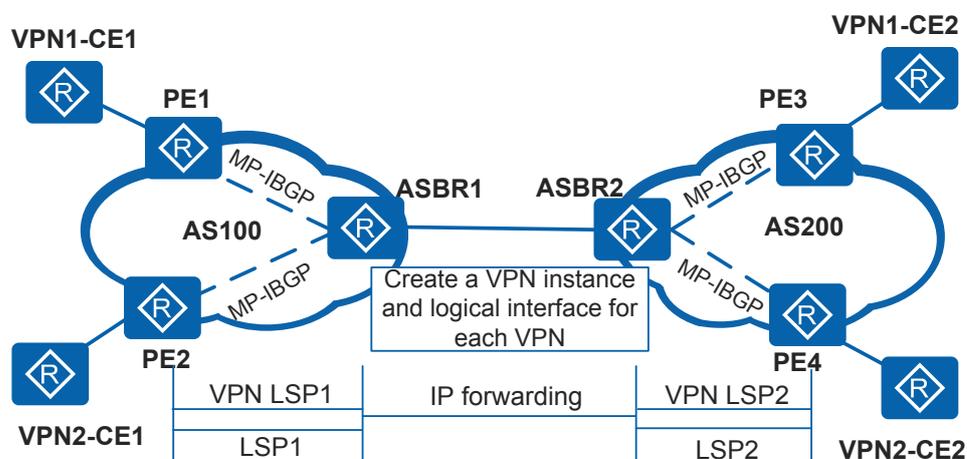
Inter-Provider Backbones Option A

- Introduction

Option A is a basic BGP/MPLS IP VPN application in an inter-AS scenario. In this solution, ASBRs do not require extra configurations for inter-AS VPN or run MPLS. ASBRs of the two ASs are directly connected and function as the PE devices of the ASs. Each ASBR considers the peer ASBR as its CE device and creates a VPN instance for each VPN. The ASBRs use EBGP to advertise IPv4 routes.

As shown in [Figure 8-12](#), ASBR2 in AS200 is a CE of ASBR1 in AS 100, and ASBR1 is the CE of ASBR2. VPN LSP indicates a private tunnel, and LSP indicates a public tunnel.

Figure 8-12 Inter-Provider Backbones Option A

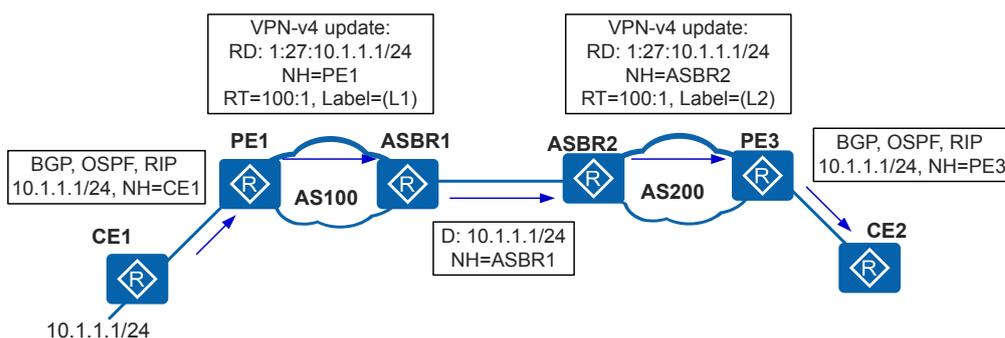


- Route advertisement

In Option A, PE and ASBR devices use MP-IBGP to exchange VPN-IPv4 routes. Two ASBRs can run BGP, IGP multi-instance, or use static routes to exchange VPN information. EBGP is recommended for inter-AS route exchange.

Figure 8-13 shows the process of advertising the route destined for 10.1.1.1/24 from CE1 to CE2. In **Figure 8-13**, D indicates the destination address; NH indicates the next hop; L1 and L2 are private labels. **Figure 8-13** does not show advertisement of public IGP routes and distribution of public network labels.

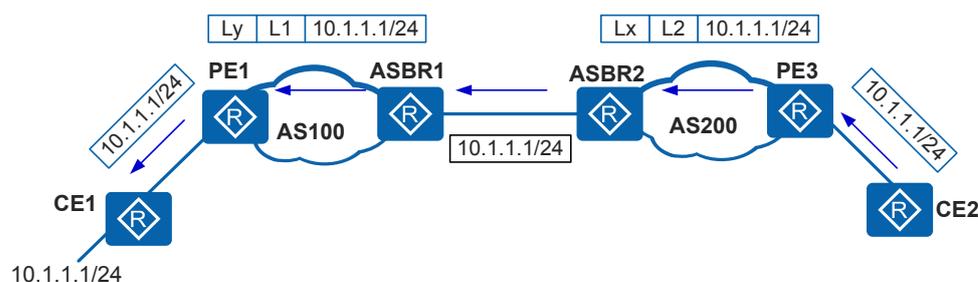
Figure 8-13 Route advertisement of Option A



- Packet forwarding

Figure 8-14 shows how packets are forwarded over the LSPs, which serve as the tunnels on the public network. L1 and L2 are inner labels; Lx and Ly are outer tunnel labels.

Figure 8-14 Packet forwarding of Option A



- Characteristics

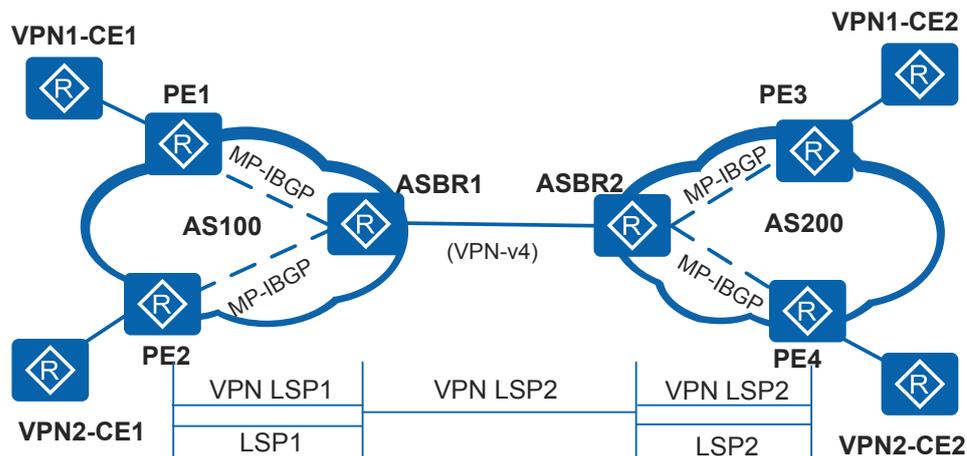
- Simplified configuration: MPLS does not need to run between ASBRs and no extra configuration is required.
- Low scalability: ASBRs need to manage all VPN routes and create VPN instances for each VPN. Because IP forwarding is performed between the ASBRs, the ASBRs must reserve an interface for each inter-AS VPN. Therefore, the PE devices must have high performance. If a VPN spans multiple ASs, the intermediate ASs must support the VPN service. The configuration is complex and intermediate ASs is affected. Option A is applicable when the number of inter-AS VPNs is small.

Inter-Provider Backbones Option B

- Introduction

In Option B, two ASBRs use MP-EBGP to exchange labeled VPN-IPv4 routes received from the PE devices in the ASs. In the figure, VPN LSPs are private network tunnels, and LSPs are public network tunnels.

Figure 8-15 Inter-Provider Backbones Option B



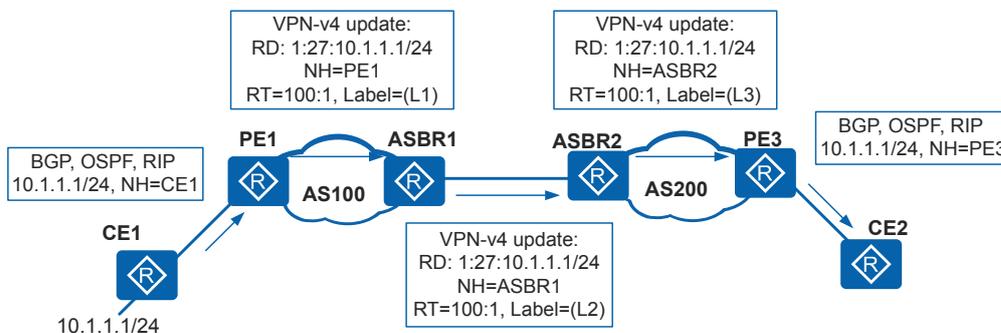
In Option B, the ASBRs receive all inter-AS VPN-IPv4 routes within or outside the local AS and advertise the routes. In basic MPLS VPN implementation, a PE device stores only the VPN routes that match the VPN target of the local VPN instance. The ASBRs are configured to store all the received VPN routes, regardless of whether any local VPN instance matches the routes.

All the traffic is forwarded by the ASBRs. This facilitates traffic control but increases the load on the ASBRs. BGP routing policies, such as VPN target filtering policies, can be configured on the ASBRs so that the ASBRs only save some of VPN-IPv4 routes.

- Route advertisement

Figure 8-16 shows how the route destined for 10.1.1.1/24 is advertised from CE1 to CE2. D indicates the destination address; NH indicates the next hop; L1, L2, and L3 are inner labels. Figure 8-16 does not show advertisement of public IGP routes and distribution of public network labels.

Figure 8-16 Route advertisement of Option B



The route advertisement process is as follows:

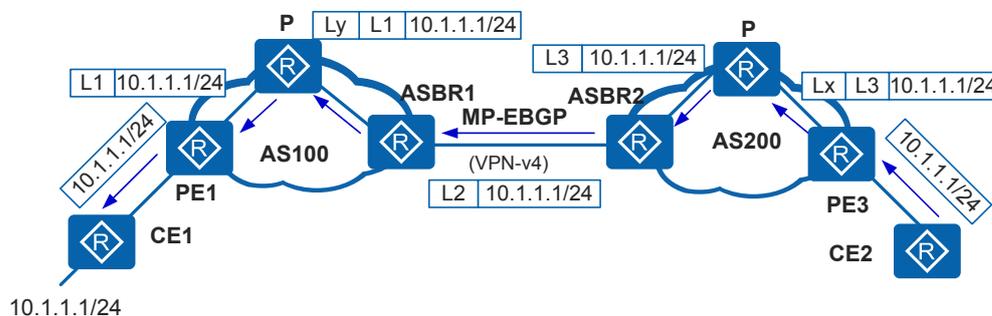
- CE1 uses BGP, OSPF, or RIP to advertise routes to PE1 in AS 100.
- PE1 in AS 100 uses MP-IBGP to advertise labeled VPNv4 routes to ASBR1 in AS 100. If a route reflector (RR) is deployed on the network, PE1 advertises the VPNv4 routes to the RR, and then the RR reflects the routes to ASBR1.
- ASBR1 uses MP-EBGP to advertise the labeled VPNv4 routes to ASBR2. Because MP-EBGP changes the next hop of the routes when advertising the routes, ASBR1 allocates a new label to the VPNv4 routes.
- ASBR2 uses MP-IBGP to advertise the labeled VPNv4 routes to PE3 in AS 200. If an RR is deployed on the network, ASBR2 advertises the VPNv4 routes to the RR, and then the RR reflects the routes to PE3. When ASBR2 advertises routes to an MP-IBGP peer in the local AS, it changes the next hop of the routes to itself.
- PE3 in AS 200 uses BGP, OSPF, or RIP to advertise the routes to CE2.

Both ASBR1 and ASBR2 swap inner labels of the VPNv4 routes. The inter-AS labels are carried in BGP messages, so the ASBRs do not need to run signaling protocols such as Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP).

- Packet forwarding

In Option B, both the ASBRs swap labels during packet forwarding. **Figure 8-17** shows how packets are forwarded over the LSPs, which serve as the tunnels on the public network. L1, L2, and L3 are inner labels; Lx and Ly are outer tunnel labels.

Figure 8-17 Packet forwarding of Option B



- Characteristics

- Unlike Option A, Option B is not limited by the number of links between ASBRs.
- Information about VPN routes is stored on and advertised by ASBRs. When a large number of VPN routes exist, the overburdened ASBRs are likely to encounter bottlenecks. Therefore, in the MP-EBGP solution, the ASBRs that maintain VPN routes do not perform IP forwarding on the public network.

Inter-Provider Backbones Option C

- Introduction

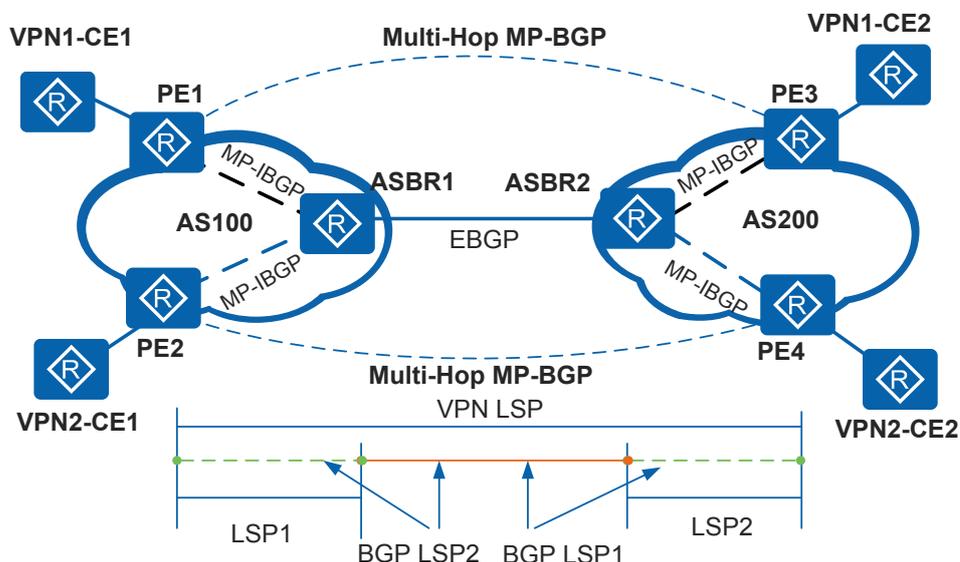
Option A and Option B can meet inter-AS VPN requirements. However, ASBRs need to maintain and distribute VPN-IPv4 routes. When each AS needs to exchange a large number of VPN routes, ASBRs may hinder network extension.

To address this issue, PE devices can directly exchange VPN-IPv4 routes, and ASBRs do not maintain or advertise VPN-IPv4 routes.

- The ASBRs use MP-IBGP to advertise labeled IPv4 routes to PE devices in their respective ASs. The ASBRs also advertise labeled IPv4 routes received from PE devices in the local AS to the ASBR peers in other ASs. The ASBRs in the transit AS also advertise labeled IPv4 routes. A VPN LSP can be established between the ingress PE and egress PE.
- The PE devices in different ASs establish a multi-hop EBGP connection to exchange VPN-IPv4 routes.
- The ASBRs do not store or advertise VPN-IPv4 routes to each other.

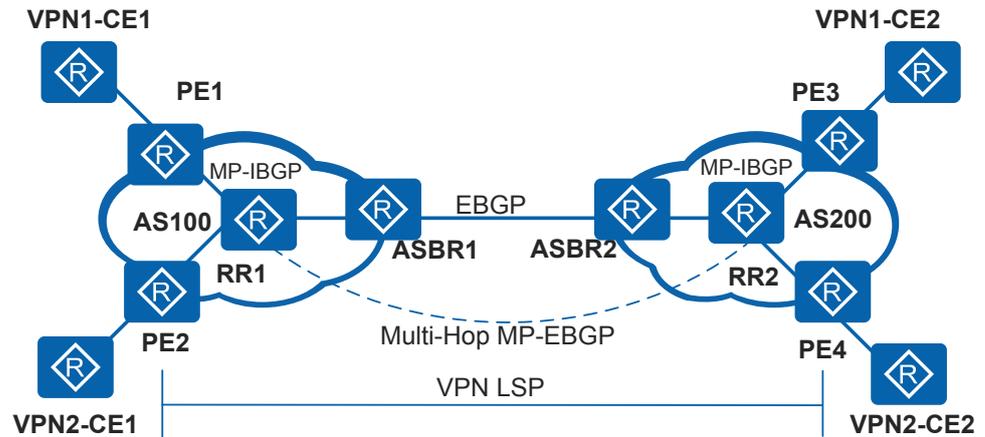
Figure 8-18 shows the networking of inter-AS VPN Option C. In the figure, VPN LSPs are private network tunnels, and LSPs are public network tunnels. A BGP LSP enables two PE devices to exchange loopback interface information, and it consists of two parts, for example, BGP LSP1 from PE1 to PE3 and BGP LSP2 from PE3 to PE1.

Figure 8-18 Inter-Provider Backbones Option C



To improve network scalability, you can specify an RR in each AS. The RR stores all VPN-IPv4 routes and exchanges VPN-IPv4 routes with the PE devices in the local AS. The RRs in two ASs establish an MP-EBGP connection to advertise VPN-IPv4 routes.

Figure 8-19 Inter-Provider Backbones Option C with an RR

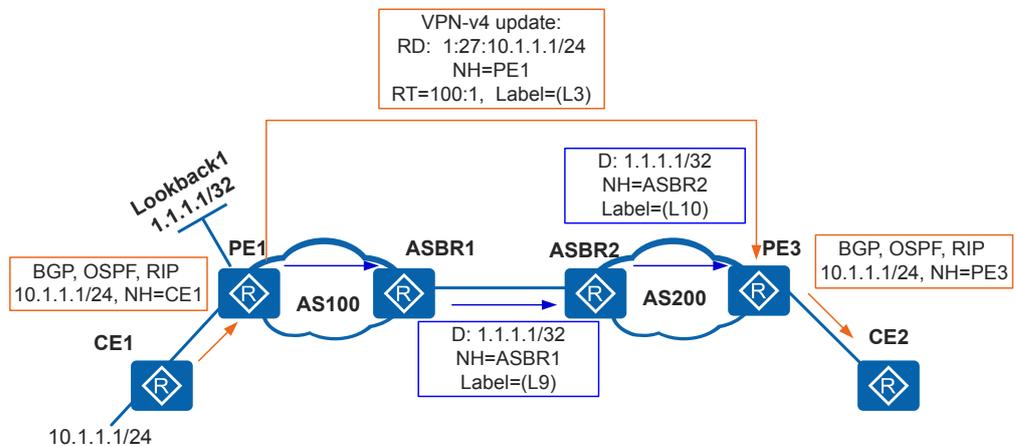


- Route advertisement

The key to Option C is establishment of inter-AS tunnels on a public network.

Figure 8-20 shows how the route destined for 10.1.1.1/24 is advertised from CE1 to CE2. D indicates the destination address; NH indicates the next hop; L3 indicates the inner label. L9 and L10 are BGP LSP labels. **Figure 8-20** does not show advertisement of public IGP routes and distribution of public network labels.

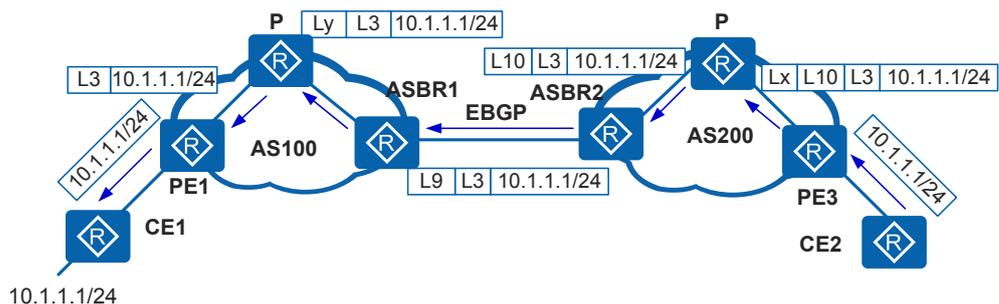
Figure 8-20 Route advertisement of Option C



- Packet forwarding

Figure 8-21 shows how packets are forwarded over the LSPs, which serve as the tunnels on the public network. L3 is the inner label; L9 and L10 are BGP LSP labels; Lx and Ly are outer tunnel labels.

Figure 8-21 Packet forwarding of Option C



Before forwarding a packet to PE1, PE2 adds three labels to the packet: VPN route label, BGP LSP label, and public LSP label. When the packet reaches ASBR2, two labels are left: VPN route label and BGP LSP label. When the packet reaches ASBR1, the BGP LSP label is terminated. Then common MPLS VPN forwarding is performed.

- Characteristics
 - VPN routes are directly exchanged between the ingress PE and the egress PE. The routes do not need to be stored and forwarded by intermediate devices.
 - Only PE devices need to exchange VPN routes. P devices and ASBRs are only responsible for packet forwarding. The intermediate devices need to support only MPLS forwarding, and do not need to support MPLS VPN services. ASBRs are unlikely to encounter bottlenecks. Option C is suitable for the VPNs that span multiple ASs.
 - MPLS VPN load balancing is easy to carry out in Option C.
 - Managing an end-to-end connection between PE devices has high costs.

8.2.5 MCE

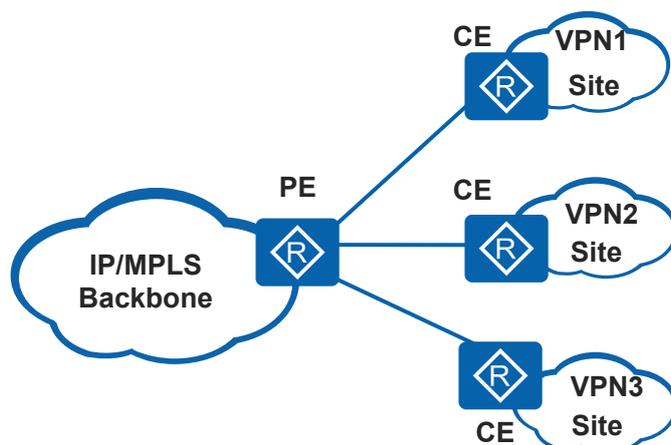
Definition

A multi-VPN-instance CE (MCE) device can function as a CE device for multiple VPN instances in BGP/MPLS IP VPN networking. The MCE function helps reduce expenses of network devices.

Background

BGP/MPLS IP VPN uses tunnels to transmit data of private networks on a public network. In the traditional BGP/MPLS IP VPN architecture, each VPN instance must use a CE device to connect to a PE device, as shown in [Figure 8-22](#).

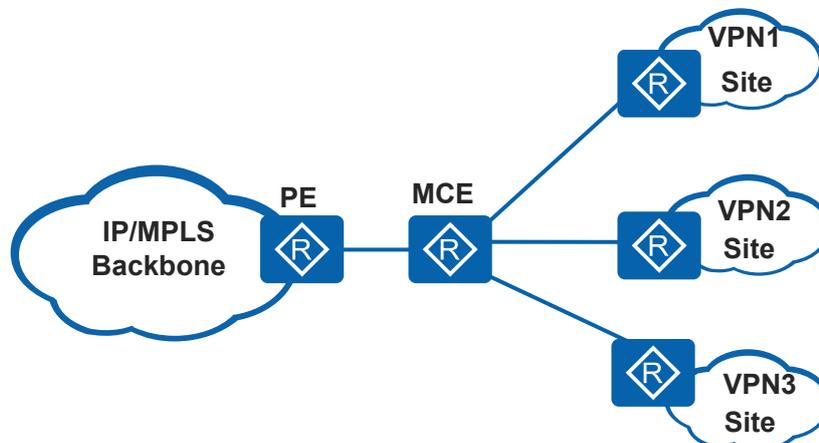
Figure 8-22 Networking without an MCE device



In many cases, a private network must be divided into multiple VPNs to implement fine-grained service management and enhance security. Services of users in different VPNs must be completely isolated. Deploying a CE device for each VPN increases the cost of device procurement and maintenance. If multiple VPNs share one CE device, data security cannot be ensured because all the VPNs use the same routing and forwarding table.

MCE technology ensures data security between different VPNs while reducing network construction and maintenance costs. Figure 8-23 shows MCE networking.

Figure 8-23 Networking with an MCE device



An MCE device has some PE functions. By binding each VPN instance to a different interface, an MCE device creates and maintains an independent VRF for each VPN. This application is also called multi-VRF application. The MCE device isolates forwarding paths of different VPNs on a private network and advertises routes of each VPN to the peer PE device, ensuring that VPN packets are correctly transmitted on the public network.

Implementation

An MCE device maintains a VRF for each VPN and binds each VPN instance to an interface. When the MCE device receives a route, it checks the receiving interface to determine the origin of the route and adds the route to the VRF of the VPN instance bound to the interface.

The PE interfaces connected to the MCE device must also be bound to the VPN instances. The bindings between interfaces and VPN instances on the PE device must be the same as those on the MCE device. When the PE device receives a packet, it checks the receiving interface to determine to which VPN the packet belongs, and then transmits the packet in the corresponding tunnel.

In **Figure 8-23**:

- The MCE device saves routes learned from VPN1 in VRF1.
- The PE device saves routes of VPN1 learned from the MCE device in VRF1.
- Routes of VPN2 and VPN3 are isolated from routes of VPN1, and are not saved in VRF1.

The MCE device exchanges routes with VPN sites and PE device in the following ways:

- Route exchange with VPN sites

Route Exchange Method	Implementation
Static routes	Static routes are bound to VPN instances on the MCE device. Static routes of different VPNs are isolated even if VPNs use overlapping address spaces.
Routing Information Protocol (RIP)	Each VPN instance is bound to a RIP process on the MCE device so that routes of different VPNs are exchanged between the MCE device and VPN sites using different RIP processes. This isolates routes of different VPNs and ensures security of VPN routes.
Open Shortest Path First (OSPF)	Each VPN instance is bound to an OSPF process on the MCE device to isolate routes of different VPNs.
Intermediate System to Intermediate System (IS-IS)	Each VPN instance is bound to an IS-IS process on the MCE device to isolate routes of different VPNs.
Border Gateway Protocol (BGP)	Each VPN instance is configured with a BGP peer on the MCE device. The MCE imports IGP routes of each VPN to the BGP routing table of the VPN.

- Route exchange with the PE device

Routes of different VPN instances are isolated on the MCE device. The MCE and PE devices identify packets of different VPN instances according to bindings between interfaces and VPN instances. An administrator only needs to perform simple routing configuration on the MCE and PE devices, and to import the VPN routes of the MCE device to the routing protocol running between the MCE and PE devices.

The MCE and PE devices can use static routes, RIP, OSPF, IS-IS, or BGP to exchange routes.

8.2.6 HoVPN

Definition

Hierarchy of VPN (HoVPN) is a multi-layer VPN architecture that deploys PE functions on multiple PE devices. In this architecture, multiple PE devices play different roles and fulfill the functions of one PE. HoVPN is also called hierarchy of PE (HoPE).

Background

As key devices on a BGP/MPLS IP VPN network, PE devices provide must provide a large number of interfaces for user access, and provide large-capacity memory and high forwarding capabilities to manage and advertise VPN routes, and process user packets.

Most networks use typical hierarchical architecture. For example, a MAN uses a three-layer architecture consisting of the core, aggregation, and access layers. From the core layer to the access layer, the requirements for device performance decreases, but the network scale increases.

BGP/MPLS IP VPN uses a plane model, which has the same performance requirement for all the PE devices. If some PE devices do not provide high performance or scalability, the entire network is affected.

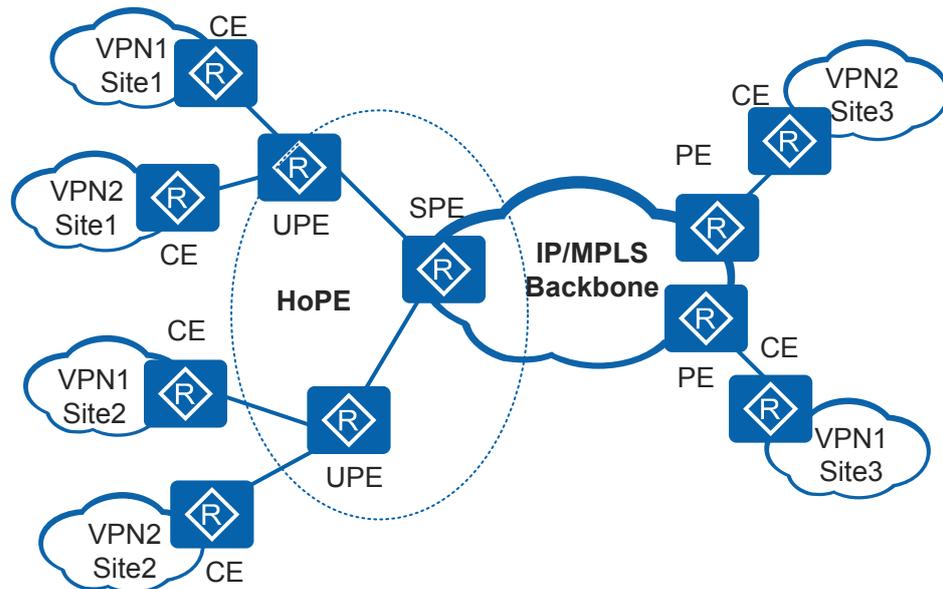
Because the plane model of BGP/MPLS IP VPN is different from the typical hierarchical architecture, deployment of new PE devices at each layer is difficult due to low scalability. This plane model hinders large-scale VPN deployment. The HoVPN solution is developed to address this issue.

In the HoVPN model, devices at higher layers must have high routing and forwarding capabilities, whereas devices at lower layers can have lower capabilities.

Implementation

- HoVPN architecture

Figure 8-24 HoVPN architecture



As shown in [Figure 8-24](#), the devices directly connected to user devices are called underlayer PE or user-end PE (UPE) devices. The device that is deployed within the backbone network and connected to UPE devices is called a superstratum PE or service provider-end PE (SPE) device.

Multiple UPE devices and an SPE device form a hierarchy of PE and provide functions of a traditional PE device.

- Relationship between the UPE and SPE
 - The UPE device provides user access. It maintains routes of directly connected VPN sites, but does not maintain routes of remote VPN sites or only maintains summarized routes of remote VPN sites. Each UPE device assigns an inner label to routes of directly connected sites and uses MP-BGP to advertise the label with the VPN routes to the SPE device.
 - The SPE device manages and advertises VPN routes. It maintains all the routes of the VPN sites connected through the UPE devices, including routes of local and remote sites. However, the SPE does not advertise routes of remote sites to the UPE devices. Instead, it advertises only default routes of VPN instances with labels.
 - The UPE and SPE devices use label forwarding. The SPE device uses only one interface to connect to each UPE device and does not need to provide many interfaces for access users. An UPE device can connect to the SPE device through a physical interface, a sub-interface, or a tunnel interface. If a tunnel interface is used, the UPE and SPE devices can communicate across an IP or MPLS network. Labeled packets are transmitted between the UPE and SPE devices through a tunnel. If a GRE tunnel is used, GRE must support encapsulation of MPLS packets.
- As an SPE device and a UPE device play different roles, requirements for them are different:
- An SPE device has a large routing table, high forwarding performance, but few interfaces.
 - A UPE device has a small routing table, low forwarding performance, and high access capabilities.

A PE device is a SPE device for a lower-layer PE device and is a UPE device for an upper-layer PE device.

An HoPE can coexist with common PE devices on an MPLS network.

- SPE-UPE

If a UPE device and an SPE device belong to the same AS, MP-BGP running between them is MP-IBGP. If they belong to different ASs, MP-BGP running between them is MP-EBGP.

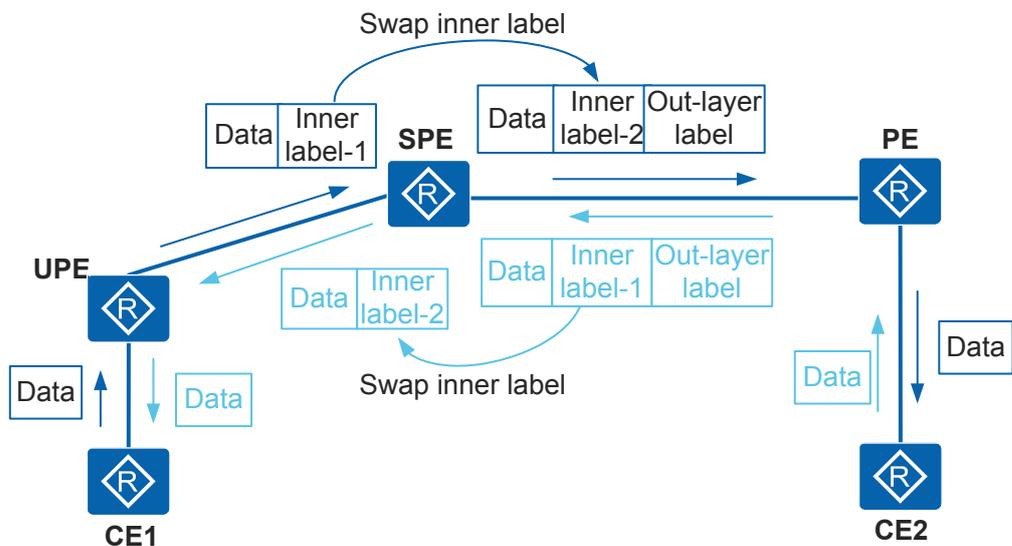
When MP-IBGP is used, an SPE device can function as an RR of multiple UPE devices to advertise routes between the IBGP peers. To reduce the number of routes on the UPE devices, do not use the SPE as an RR for other PE devices.

A UPE device can connect to multiple SPE devices. This networking is called UPE multi-homing. In this networking, the SPE devices advertise the VRF default routes to the UPE device. The UPE device selects one route as the optimal route or selects multiple routes to perform load balancing. The UPE device advertises all the VPN routes to the SPE devices or advertises some of VPN routes to each SPE to implement load balancing.

- Label operation in HoVPN

Figure 8-25 shows an example of label operation in HoVPN. In this example, an LSP tunnel is set up between the SPE and PE devices.

Figure 8-25 Label operation in HoVPN



- CE1 → CE2 (marked by the black line)
 - After receiving a packet from CE1, the UPE device adds an inner label to the packet and forwards the packet to the SPE device.
 - After receiving the labeled packet, the SPE device swaps the inner label, adds an outer LSP label to the packet, and sends the packet to the PE device.
 - After the packet arrives at the previous hop of the PE device, this hop pops the outer LSP label. The process is called penultimate hop popping.
 - After the PE device receives the packet, it pops the inner label.

- CE2 → CE1 (marked by the blue line)
 - After receiving a packet from CE2, the PE device adds an inner label and an outer LSP label to the packet, and then forwards the packet to the SPE device.
 - After the packet arrives at the previous hop of the SPE device, this hop pops the outer LSP label.
 - The SPE device swaps the inner label for a new one and forwards the packet to the UPE device.
 - After the UPE device receives the packet, it pops the inner label.
- HoVPN embedding and extension
 HoVPN supports HoPE embedding.
 - An HoPE can function as a UPE device and compose a new HoPE with an SPE device.
 - An HoPE can function as an SPE device and compose a new HoPE with multiple UPE devices.
 - HoPEs can be embedded multiple times in the preceding two modes.

HoPE embedding can infinitely extend a VPN.

Figure 8-26 HoPE embedding

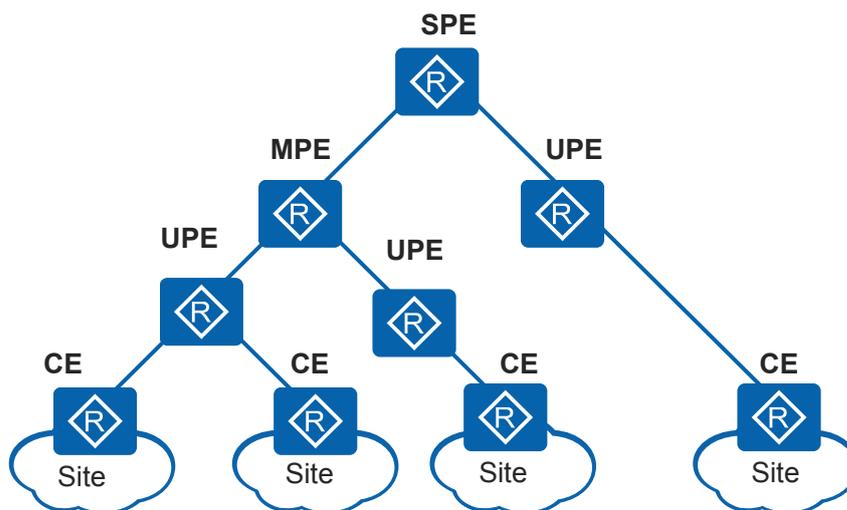


Figure 8-26 shows a three-layer HoPE, and the PE device in the middle is called the middle-level PE (MPE) device. MP-BGP runs between the SPE and MPE devices, and between the MPE and UPE devices.

NOTE

Actually, the MPE device does not exist in an HoVPN model. The concept is used just for the convenience of description.

MP-BGP advertises all the VPN routes of the UPE devices to the SPE device, but advertises only the default VPN routes of the SPE device to the UPE devices.

The SPE device maintains the routes of all VPN sites connected to the PE devices, whereas the UPE devices maintain only the VPN routes of the directly connected VPN sites. The quantities of routes maintained by the SPE, MPE, and UPE devices are in descending order.

Advantages of HoVPN

The HoVPN model has the following advantages:

- A BGP/MPLS IP VPN network can be divided into different hierarchies. If the performance of UPE devices does not satisfy service requirements, an SPE device can be added above UPE devices. When access capabilities of an SPE device are insufficient, UPE devices can be added below the SPE device.
- Label forwarding is performed between UPE and SPE devices. Therefore, a UPE device and an SPE device are interconnected through only a pair of interfaces or sub-interfaces. This saves interface resources.
- If a UPE device and an SPE device are separated by an IP or MPLS network, they can set up a GRE or LSP tunnel. A layered MPLS VPN has enhanced scalability.
- The UPE devices maintain only the local VPN routes, and all the remote routes are represented by a default or summarized route. This reduces loads on the UPE devices.
- SPE and UPE devices use MP-BGP to exchange routes and advertise labels. Each UPE device sets up only one MP-BGP peer, reducing the protocol cost and configuration workload.

8.2.7 VPN FRR

Definition

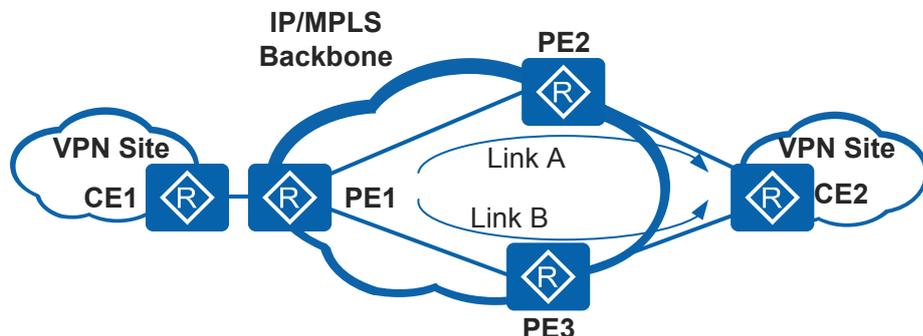
As networks develop rapidly, the time used for end-to-end service convergence if a fault occurs on a carrier's network has been used as an indicator to measure bearer network performance. MPLS TE Fast Reroute (FRR) is one of the commonly used fast switching technologies. The solution is to create an end-to-end TE tunnel between two PEs and a backup LSP that protects a primary LSP. When either of the PE devices detects that the primary LSP is unavailable because of a node or link failure, the PE switches the traffic to the backup LSP.

MPLS TE FRR protects services in the case of a link or node failure between two PE devices at both ends of a TE tunnel; however, MPLS TE FRR cannot protect services in the case of a PE device failure. If a fault occurs on the ingress or egress, services can only be restored through end-to-end route convergence and LSP convergence. The service convergence time is closely related to the number of routes inside an MPLS VPN and the number of LSP hops on the bearer network. The more VPN routes, the longer the service convergence time, and the more traffic is lost.

VPN FRR sets in advance on a remote PE device forwarding entries pointing to the active and standby PE devices, respectively. In collaboration with fast PE fault detection, VPN FRR can reduce end-to-end service convergence time if a fault occurs on an MPLS VPN where a CE device is dual-homed to two PE devices. In VPN FRR, service convergence time depends on only the time required to detect remote PE device faults and change tunnel status. VPN FRR enables the service convergence time to be irrelevant to the number of VPN routes on the bearer network.

Implementation

Figure 8-27 Typical VPN FRR networking



As shown in [Figure 8-27](#), normally, CE1 accesses CE2 over Link A. If PE2 is Down, CE1 accesses CE2 over Link B.

- Based on the traditional BGP/MPLS VPN technology, both PE2 and PE3 advertise routes destined for CE2 to PE1, and assign VPN labels to these routes. PE1 then selects a preferred VPNv4 route based on the routing policy. In this example, the preferred route is the one advertised by PE2, and only the routing information, including the forwarding prefix, inner label, selected LSP, advertised by PE2 is filled in the forwarding entry of the forwarding engine to guide packet forwarding.
- When PE2 fails, PE1 detects the fault of PE2 (the BGP peer relationship becomes Down or the outer LSP is unavailable). Then PE1 selects the route advertised by PE3 and updates the forwarding entry to complete end-to-end convergence. Before PE1 delivers the forwarding entry matching the route advertised by PE3, CE1 cannot communicate with CE2 for a certain period because the destination of the outer LSP, PE2, is Down. As a result, end-to-end services are interrupted.
- VPN FRR is an improvement on the traditional reliability technology. VPN FRR enables PE1 to add the optimal route advertised by PE2 and the secondary optimal route advertised by PE3 to a forwarding entry. The optimal route is used for traffic forwarding, and the secondary optimal route is used as a backup route.
- If a fault occurs on PE2, the MPLS LSP between PE1 and PE2 becomes unavailable. After detecting the fault, PE1 marks the corresponding entry in the LSP status table as unavailable, and delivers the setting to the forwarding table. After selecting a forwarding entry, the forwarding engine examines the status of the LSP corresponding to the forwarding entry. If the LSP is unavailable, the forwarding engine uses the second-best route carried in the forwarding entry to forward packets. After being tagged with the inner labels assigned by PE3, packets are transmitted to PE3 over the LSP between PE1 and PE3 and then forwarded to CE2. In this manner, fast end-to-end service convergence is implemented and traffic from CE1 to CE2 is restored.

VPN FRR performs fast switching based on inner labels. Outer tunnels can be LDP LSPs or RSVP TE tunnels. When the forwarding engine detects that the outer tunnel is unavailable, it triggers fast switching based on inner labels.

8.2.8 VPN GR

NOTE

The AR3260-S can function as both the GR restarter and GR helper, and other devices can only function as the GR helper.

Definition

VPN GR is an application of GR technology on a VPN. VPN GR ensures uninterrupted VPN traffic forwarding when an active/standby switchover is performed on a device transmitting VPN services. The purposes of VPN GR are as follows:

- Reduce the impact of route flapping on the entire network during the switchover.
- Reduce the impact on important VPN services.
- Reduce single-point failures on PE or CE devices to improve VPN network reliability.

Prerequisites for VPN GR

The device where an active/standby switchover occurs and its connected devices must have GR capabilities. They must retain forwarding information of all VPN routes within a period to ensure uninterrupted VPN traffic forwarding. That is, the devices must support IGP GR, BGP GR, and LDP GR. If TE tunnels are deployed on the backbone network, the devices must support RSVP GR.

Implementation

On a common BGP/MPLS VPN network, active/standby switchovers may occur on any PE, CE, or P device.

- Active/standby switchover on a PE device

The GR process on a PE device is the same as that on the GR restarter in IGP GR, BGP GR, or LDP GR.

When a CE device connected to the PE device detects the restart of the PE device, the CE device acts the same as the GR helper IGP GR or BGP GR and retains all IPv4 routes in a period.

When the P device connected to the PE device detects the restart of the PE device, the P device acts the same as the GR helper in IGP GR, BGP GR, or LDP GR and retains all public IPv4 routes in a period.

When other PE devices (including those functioning as ASBRs) and the RR reflecting VPNv4 routes detect the restart of the PE device, they act the same as the GR helper in BGP GR, and retain all the public IPv4 routes and VPNv4 routes in a period.

- Active/standby switchover on a P device

The GR process on a P device is the same as that on the GR restarter in IGP GR, BGP GR, or LDP GR.

When a P or PE device connected to this P device detects the restart, the P or PE device acts the same as the GR helper in IGP GR, BGP GR or LDP GR and retains all the public IPv4 routes in a period.

- Active/standby switchover on a CE device

The GR process on a CE device is the same as that on the GR restarter in IGP GR or BGP GR.

When the PE device connected to the CE device detects the restart of the CE device, the PE device acts the same as the GR helper in IGP GR or BGP GR and retains all the private IPv4 routes in a period.

For details about IGP GR and BGP GR, see "GR" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

For details about LDP GR and RSVP GR, see "GR" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS*.

8.2.9 VPN NSR

The BGP/MPLS IP VPN supports Non-stop Routing (NSR), which ensures uninterrupted VPN operating during an active/standby switchover. For details about NSR, see "NSR" in the *Feature Description - ReliabilityHuawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Reliability*.

NSR backs up the following data to ensure uninterrupted BGP/MPLS IP VPN operating:

- VPN forwarding table
- Labels

8.2.10 VPN Tunnel Policy

Introduction to VPN Tunnels

VPN data is transmitted over tunnels, including LSP tunnels, GRE tunnels, and Traffic Engineering (TE) tunnels. TE tunnels are constraint-based routed label switched path (CR-LSP) tunnels.

- GRE tunnel

If PE devices support MPLS functions but P devices on the backbone network provide only IP functions, LSPs cannot serve as tunnels. In this situation, GRE tunnels can be used as the tunnels of the VPN backbone network.

For details about GRE, see [3 GRE Configuration](#) in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - VPN*.

- LSP

An LSP forwards packets through label switching and is often used in BGP/MPLS IP VPN. If LSPs are used as public network tunnels, only PE devices need to analyze IP packet headers, and other devices that VPN packets pass do not need to analyze IP packet headers. This reduces VPN packet processing time and packet transmission delay. In addition, MPLS labels are supported by all link layers. An LSP is similar to an ATM virtual circuit (VC) or FR VC in functions and security. If all the devices on the backbone network support MPLS, it is recommended that LSP tunnels or MPLS TE tunnels be used as public network tunnels.

For details about LSPs, see MPLS LDP Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS*.

- MPLS TE tunnel

As a combination of MPLS and TE technologies, MPLS TE can balance network traffic by setting up LSPs along specified nodes and steering traffic away from congested nodes. LSPs in MPLS TE are called MPLS TE tunnels, which are also widely used in BGP/MPLS IP VPN.

Besides advantages of LSP, MPLS TE tunnels is capable of handling network congestion. Using MPLS TE tunnels, SPs can fully utilize existing network resources to provide diversified services. MPLS TE tunnels also allow SPs to optimize network resources and manage resources.

Usually, carriers are required to provide VPN users with end-to-end QoS for various services, such as voice, video, key-data services, and Internet access. MPLS TE tunnels can offer users with QoS guarantee.

Using MPLS TE tunnels, carriers can also provide required QoS guaranteed services for different VPN users based on policies.

For details about MPLS TE, see MPLS TE Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS*.

Tunnel Policy

VPN services are transmitted over tunnels. By default, LSPs are preferred in VPN service transmission, and only one LSP serves one VPN service.

When VPN services need to be transmitted over a specified TE tunnel or when load balancing needs to be performed among multiple tunnels to fully use network resources, tunnel policies need to be applied to VPNs. Tunnel policies are classified into two types, which cannot be configured simultaneously:

- Tunnel type prioritization policy: specifies the sequence in which each type of tunnel is selected and the number of tunnels participating in load balancing. Tunnels defined in a tunnel type prioritization policy are selected in sequence: The tunnels of the type specified first are selected as long as the tunnels are in Up state, regardless of whether they are in use. The tunnels of the type specified later are not selected unless load balancing is required or the tunnels of the type specified first are all Down.

For example, a tunnel policy defines the following rules: Both CR-LSPs and LSPs can be used, CR-LSPs are prior to LSPs, and the number of tunnels participating in load balancing is 3. Tunnels are selected as follows:

- CR-LSPs in Up state are preferred. If three or more CR-LSPs are in Up state, the three CR-LSPs listed earlier are selected.
- If there are less than three CR-LSPs in Up state, LSPs are selected. For example, if only one CR-LSP is in Up state, two LSP tunnels can be selected. If only one LSP or none is in Up state, the existing tunnels in Up state are used. If more than two LSPs are in Up state, only the first two LSPs are selected.

NOTE

If a TE tunnel is reserved for tunnel binding, the TE tunnel cannot be selected.

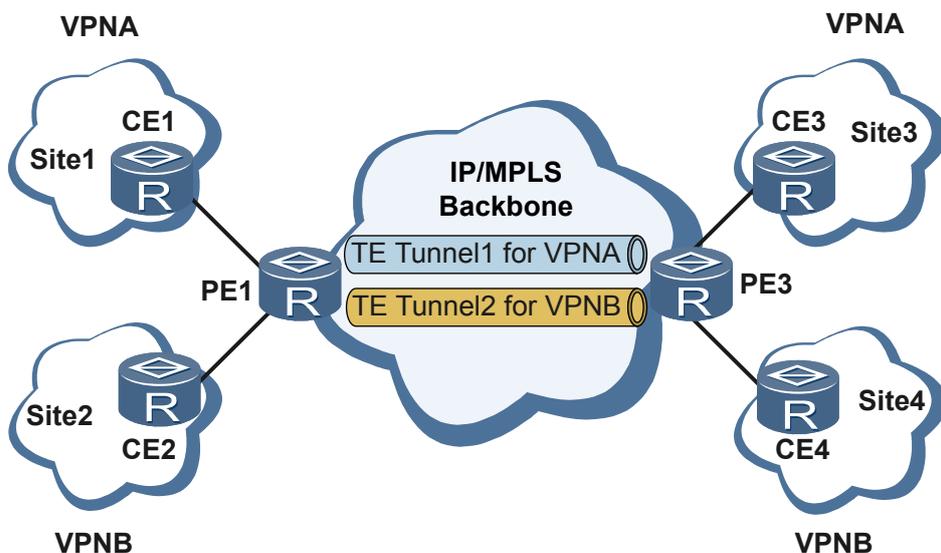
The tunnel type prioritization policy cannot specify the desired tunnels to use when multiple tunnels of the same type are available.

- Tunnel binding policy: specifies TE tunnels for carrying services of a VPN. You can specify multiple TE tunnels to the same destination for load balancing. You can also determine whether to use other tunnels to prevent traffic interruption when the specified tunnels are all unavailable. The rules for tunnel selection are as follows:

- Specified TE tunnels in Up state are selected to perform load balancing.
- If all the specified TE tunnels are unavailable, no other tunnel is selected by default. If you enable a PE device to select other tunnels in this situation, the PE device selects an available tunnel in the order of LSP and CR-LSP.

A tunnel binding policy can specify accurate TE tunnels over which VPN services are transmitted. TE tunnels have high reliability and guaranteed bandwidth, so tunnel binding policies can be used for VPN services requiring QoS guarantee. As shown in **Figure 8-28**, two MPLS TE tunnels, Tunnel1 and Tunnel2, are set up between PE1 and PE3.

Figure 8-28 Networking diagram of VPN tunnel binding



If you bind VPN A to Tunnel1 and VPN B to Tunnel2, VPN A and VPN B use different TE tunnels. Tunnel1 serves only VPN A, and Tunnel2 serves only VPN B. In this manner, services of VPN A and VPN B are isolated from each other and also from other services. The bandwidth for VPN A and VPN B is ensured. This facilitates subsequent QoS deployment.

Tunnel Selector

In HoVPN or inter-AS VPN Option B, SPE devices or ASBRs accept VPNv4 routes from all the UPE or PE devices. Currently, PE devices iterate LSP tunnels for VPNv4 routes. Sometimes, TE tunnels need to be iterated for VPNv4 routes to provide guaranteed bandwidth; the PE devices cannot provide this function by default.

In inter-AS VPN Option C, PE devices select LSP tunnels for BGP-IPv4 labeled routes. To provide guaranteed bandwidth, TE tunnels need to be iterated for VPNv4 routes, which cannot be implemented on the PE devices by default.

Tunnel selector addresses this issue.

The tunnel selector can filter VPNv4 routes or BGP-IPv4 labeled routes and apply a tunnel policy to the routes that pass the filtering criteria. In this way, expected tunnels can be selected based on the tunnel policy.

8.3 Application Scenarios for BGP/MPLS IP VPN

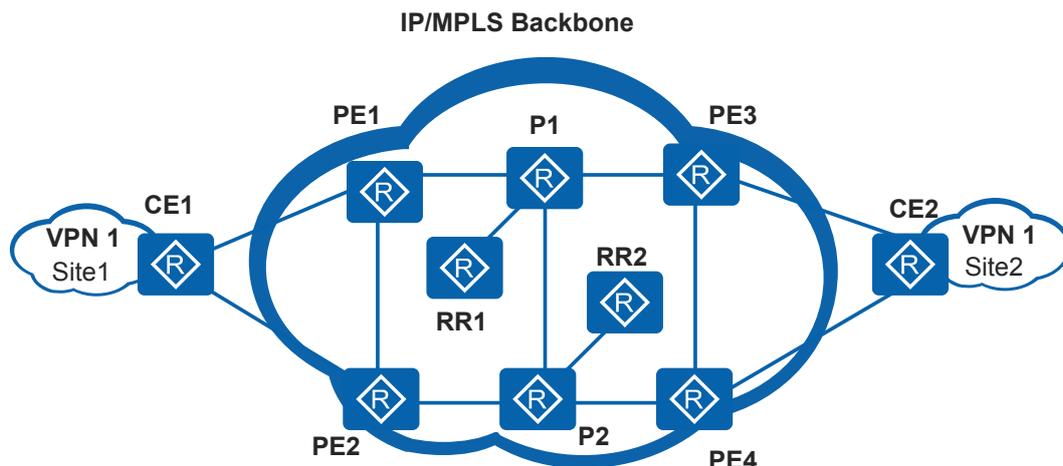
This section describes the application scenarios for BGP/MPLS IP VPN.

8.3.1 BGP/MPLS IP VPN Application

Service Overview

Figure 8-29 shows a typical networking diagram for a carrier. Site1 and Site2 represent two networks in different cities. The two networks may be networks for two branches of a company, or networks for municipal governments of the two cities. During communication between Site1 and Site2, data security must be ensured. The two networks must be separated from other networks and packets exchanged between the two networks must be transparently transmitted over the carrier's backbone network. BGP/MPLS VPN technology can meet such service requirements. VPN labels assigned using MP-BGP enable packets to enter the correct VPN site and MPLS enables packets to be transparently transmitted over tunnels on the carrier's backbone network.

Figure 8-29 BGP/MPLS IP VPN application



Networking Description

PE and P devices on the carrier's backbone network must be used to transmit routes and packets between Site1 and Site2 from the two networks to communicate. CE devices can be dual-homed to PE devices to ensure high network availability. Generally, a carrier deploys route reflectors (RRs) on the backbone network to reflect VPNv4 and VPNv6 routes.

Feature Deployment

In BGP/MPLS IP VPN networking, the following configurations must be performed:

- Configure static routes between CE devices and PE devices or configure RIP, OSPF, IS-IS, or BGP on CEs and PEs for them to exchange routing information.

- Configure MP-BGP peer relationships between all PE devices and RR1 and between all PE devices and RR2. Configure all PE devices as the clients of RR1 and RR2 and configure RR1 and RR2 to back up each other. These configurations ensure network reliability.
- Configure MPLS and an IGP on PE and P devices and establish MPLS tunnels for traffic forwarding.
- Adjust IGP costs of links to:
 - Ensure that the two links between CE1 and CE2 work in active/standby mode. If one link fails, traffic is switched to the other link for transmission.
 - Adjust the costs of links between RRs and the backbone network. Ensure that RRs are used only for route reflection, not for traffic forwarding.
- Configure VPN FRR for services that have high requirements on real-time transmission to enhance network reliability.

8.3.2 Hub and Spoke Networking Application

Service Overview

Financial enterprises such as banks can use the Hub&Spoke networking mode to ensure financial data security. Hub&Spoke networking allows branches to exchange data only through the headquarters. In this manner, data transmission between branches is under effective supervision.

In Hub&Spoke networking, the site where the access control device of the headquarters is located is called a Hub site; other sites where branches are located are called Spoke sites. At the Hub site, a device that connects to the VPN backbone network is called a Hub-CE device. At a Spoke site, a device that connects to the VPN backbone network is called a Spoke-CE device. On the VPN backbone network, a device that connects to the Hub site is called a Hub-PE device, and a device that connects to a Spoke site is called a Spoke-PE device.

A Spoke site advertises routes to the Hub site. The Hub site then advertises the routes to other Spoke sites. Spoke sites do not advertise routes to each other. The Hub site controls communication between all the Spoke sites.

Networking Description

In Hub and Spoke networking, the following solutions can be used:

- EBGp running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices
- IGP running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices
- EBGp running between the Hub-CE and Hub-PE devices, and IGP running between Spoke-PE and Spoke-CE devices

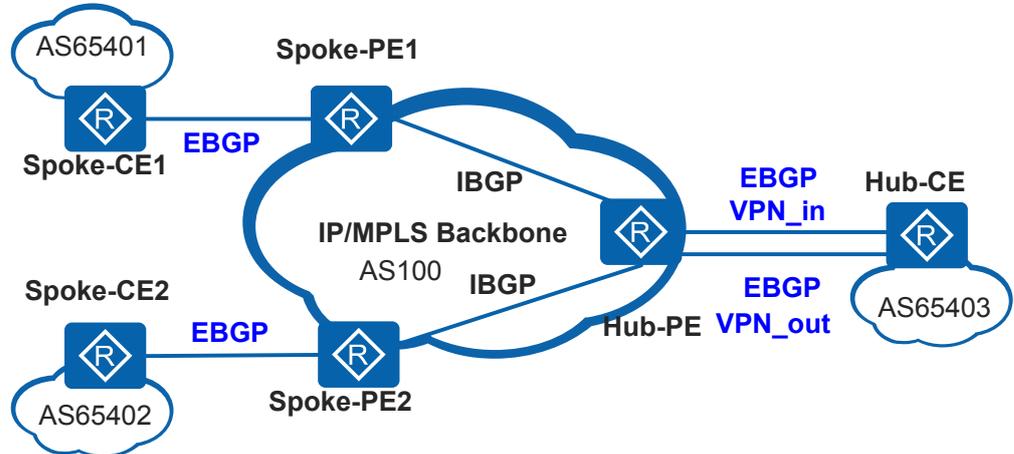
The following describes these networking solutions in detail:

- EBGp running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices

As shown in [Figure 8-30](#), a route advertised by a Spoke-CE device is forwarded to the Hub-CE and Hub-PE device before being transmitted to other Spoke-PE devices. If EBGp runs between the Hub-PE and the Hub-CE device, the Hub-PE device performs an

AS-Loop check on the route. When the Hub-PE device detects its own AS number in the route, it discards the route. To implement Hub and Spoke networking, the Hub-PE device must be configured to allow repeated AS numbers.

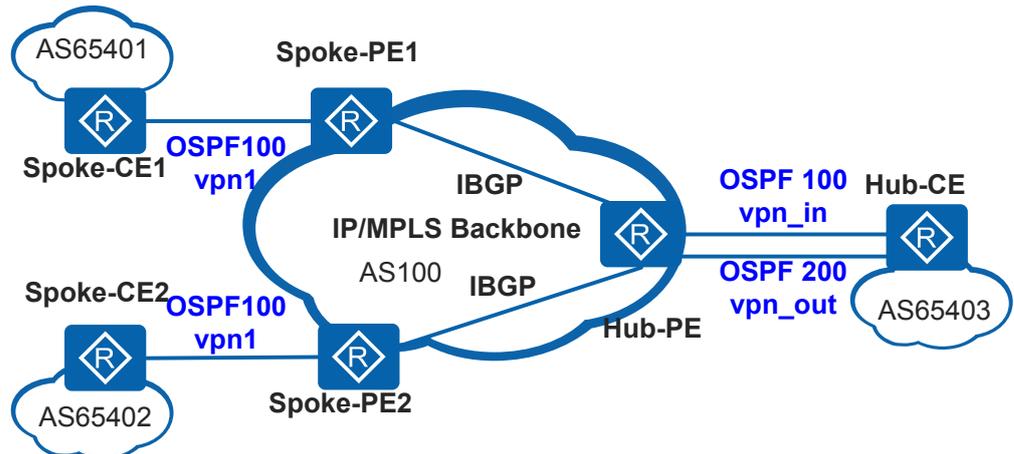
Figure 8-30 EBGP running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices



- IGP running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices

As shown in **Figure 8-31**, all PE and CE devices exchange routes using an IGP, and IGP routes do not contain the AS_Path attribute. Therefore, the AS_Path field of BGP VPNv4 routes is empty.

Figure 8-31 IGP running between the Hub-CE and Hub-PE devices, and between Spoke-PE and Spoke-CE devices

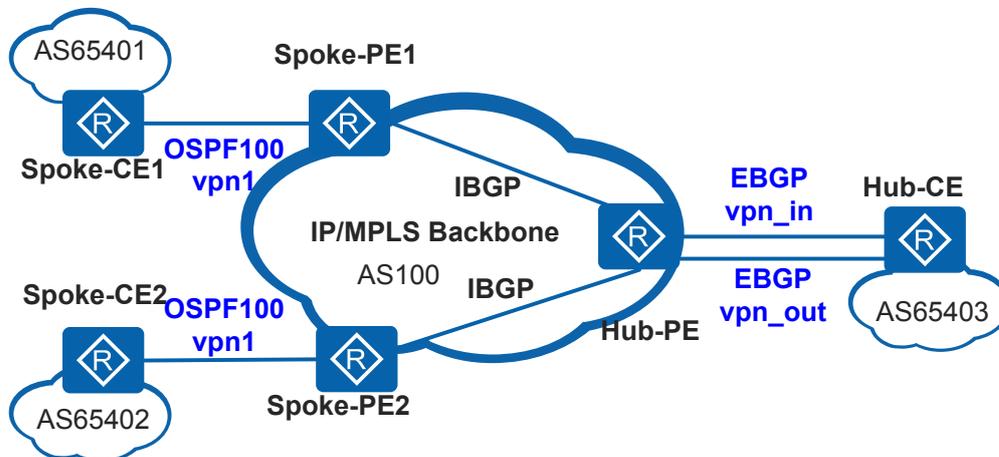


- EBGP running between the Hub-CE and Hub-PE devices, and IGP running between Spoke-PE and Spoke-CE devices

As shown in **Figure 8-32**, the network topology is similar to that shown in **Figure 8-30**. The AS_Path attribute of the routes forwarded by the Hub-CE device to the Hub-PE

device contains the AS number of the Hub-PE device. Therefore, the Hub-PE device must be configured to allow repeated AS numbers.

Figure 8-32 EBGP running between the Hub-CE and Hub-PE devices, and IGP running between Spoke-PE and Spoke-CE devices



8.3.3 Interconnection Between VPNs and the Internet

Generally, users within a VPN can only communicate with other users in the same VPN. They cannot communicate with users on the Internet or connect to the Internet. However, VPN sites may need to access the Internet. To implement interconnection between a VPN and the Internet, the following conditions must be met:

- The devices in the VPN that need to access the Internet have reachable routes to the Internet.
- Routes are available from the Internet to the devices in the VPN.
- Similar to interconnection between non-VPN users and the Internet, security mechanisms such as firewalls must be used.

Interconnection between a VPN and the Internet can be implemented in the following ways:

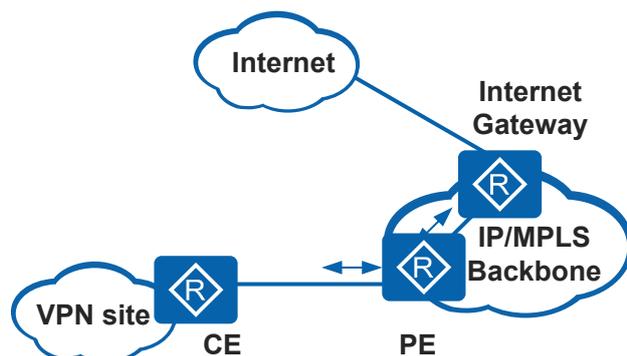
- **Interconnection implemented on a PE device:** PE devices of the backbone network identify data streams destined for the VPN and those destined for the Internet, and then forward the data to the Internet and the VPN respectively. PE devices provide the firewall function between the VPN and the Internet.
- **Interconnection implemented on an Internet gateway:** Internet gateways are carrier devices connected to the Internet. They must support VPN route management. For example, a PE device that has no VPN user attached can function as an Internet gateway.
- **Interconnection implemented on a CE device:** CE devices of the private network identify data streams destined for the VPN and those destined for the Internet, and then direct the data to two areas. One area connects to the VPN through a PE device, and the other area connects to the Internet through an ISP router that does not belong to the VPN. The CE devices provide the firewall function.

Interconnection Implemented on a PE Device

Generally, default static routes are used.

- The PE device sends a default route destined for the Internet to the CE device.
- The PE device adds a default route destined for the Internet gateway to the VPN routing table.
- To ensure that the Internet has a route to the VPN, the PE device must have a static route to the CE in the public routing table and advertise this route to the Internet. The static route is manually added to the public routing table of the PE device. In the static route, the destination address is the address of the VPN user, and the outbound interface is the PE interface that connects to the CE device. The PE uses an IGP to advertise the route to the Internet.

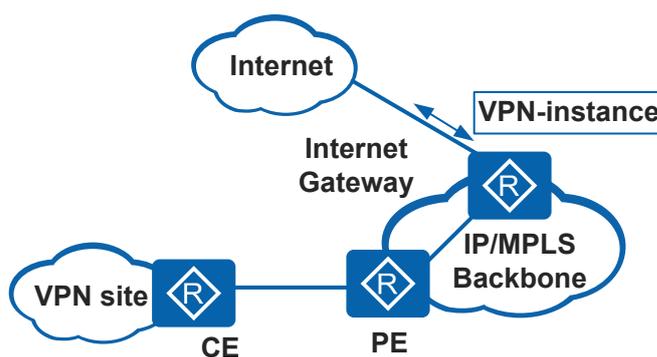
Figure 8-33 Interconnection implemented on a PE Device



Interconnection Implemented on an Internet Gateway

An instance is configured for each VPN on the Internet gateway. Each VPN uses one interface to access the Internet, and the interface is bound to the VPN instance.

Figure 8-34 Interconnection implemented on an Internet gateway

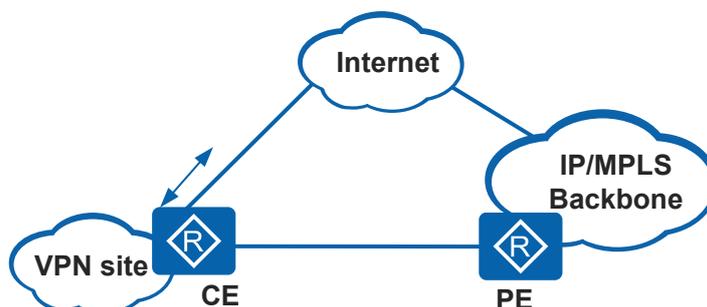


Interconnection Implemented on a CE Device

Interconnection between a VPN and the Internet can be implemented on a CE device in the following ways:

- The CE device directly connects to the Internet, as shown in [Figure 8-35](#).
A direct connection with the Internet can be achieved in the following modes:
 - One of sites (for example, central site) connects to the Internet. The CE device in the central site has a default route to the Internet, which is advertised to other sites through the backbone network. The firewalls are deployed only in the central site. In this mode, all the traffic to the Internet passes through the VPN backbone network except the traffic of the central site. A typical application of this mode is connections between the Internet and Hub sites in Hub and Spoke networking.
 - Each site connects to the Internet. Each CE device has a default route to the Internet and configured with the firewall functions. None of traffic to the Internet passes through the VPN backbone network.

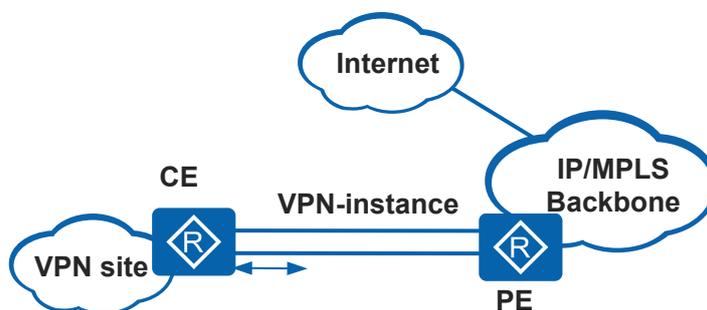
Figure 8-35 A CE device directly connects to the Internet



- A single CE interface or sub-interface connects to a PE device. The PE device injects routes of the CE device into the public routing table and advertises the routes to the Internet. Then the PE device advertises the default route or the Internet routes to the CE device. The interface that connects to the PE device does not belong to any VPN and is not associated with any VPN instance. That is, the interface can act as a VPN user and a non-VPN user to connect to the PE device, as shown in [Figure 8-36](#).

It is recommended that a tunnel be set up between the VPN backbone device connected to the Internet and the PE device connected to the CE device. Internet routes are transmitted through the tunnel, and P devices do not accept the Internet routes.

Figure 8-36 A single CE interface connects to a PE device



Comparison Between the Three Solutions

Interconnection implemented on a PE device can save interface resources and allow different VPNs to share one public IP address. However, the configuration on the PE device is

complex, and security cannot be guaranteed. Denial of Service (DoS) attacks from the Internet may occur on the PE device. When this occurs, the link between the PE and CE devices is occupied by a large amount of attack traffic, and cannot transmit valid VPN packets.

Interconnection implemented on an Internet gateway provides higher security than that on a PE device. An Internet gateway, however, must be configured with multiple VPN instances, which may overburden the gateway. In addition, an Internet gateway has multiple interfaces connected to the Internet, and each interface has a public network IP address. Each VPN uses an interface on the gateway and one public network IP address.

Interconnection implemented on a CE device is simple to deploy. This solution has high security and reliability because public routes are separated from VPN routes. However, this solution consumes interface resources and each VPN needs a public network address.

Table 8-1 Comparison between three solutions

Solution	Security	Used Interface	Used Public IP Address	Easiness of Deployment
Interconnection implemented on a PE device	Low	The PE device reserves only one interface for both VPN access and Internet access. This solution saves interface resources.	Multiple VPNs on the PE device share a public IP address.	Difficult
Interconnection implemented on an Internet gateway	High	The Internet gateway must reserve an interface for each VPN to access the Internet. This solution consumes interface resources of the gateway.	Each VPN uses a public IP address.	Difficult
Interconnection implemented on a CE	High	The CE device must reserve an interface for each VPN to access the Internet. This solution consumes interface resources of the CE.	Each VPN uses a public IP address.	Easy

8.4 Summary of BGP/MPLS IP VPN Configuration Tasks

After basic BGP/MPLS IP VPN configurations are complete, a simple VPN network can be established using MPLS technology. To deploy special BGP/MPLS IP VPN networking, perform other configuration tasks according to the reference sections provided in the following table.

Table 8-2 lists the BGP/MPLS IP VPN configuration tasks.

Table 8-2 BGP/MPLS IP VPN configuration tasks

Scenario	Description	Task
Configure basic BGP/MPLS IP VPN functions	This configuration establishes a simple BGP/MPLS IP L3VPN network with basic functions.	8.7.1 Configuring Basic BGP/MPLS IP VPN Functions
Configure BGP/MPLS IP VPN in various networking modes	<p>You adjust the basic BGP/MPLS IP L3VPN configurations in different networking mode to implement flexible communication and isolation between VPNs:</p> <ul style="list-style-type: none"> ● Intranet VPN and extranet VPN networking: The configurations are same as the configurations in basic BGP/MPLS IP VPN networking except for the VPN target setting. ● Hub and Spoke networking: configure the Hub and Spoke. 	8.7.1 Configuring Basic BGP/MPLS IP VPN Functions 8.7.2 Configuring Hub and Spoke
Configure inter-AS VPN	<p>Configure inter-AS VPN if the backbone network spans multiple ASs. Three inter-AS VPN solutions are available, applicable to different scenarios:</p> <ul style="list-style-type: none"> ● Inter-AS VPN Option A: Use this solution when only a few VPNs are configured on the PE devices. The ASBRs must support VPN instances. ● Inter-AS VPN Option B: Use this solution when many VPNs are configured on the PE devices, and the ASBRs do not have enough interfaces to reserve an interface for each inter-AS VPN. The ASBRs must be able to maintain and advertise VPN-IPv4 routes. ● Inter-AS VPN Option C: Use this solution when a large number of VPN routes need to be exchanged between ASs. This solution mitigates the loads on ASBRs so that they will not become the bottleneck on the network. 	8.7.3 Configuring Inter-AS VPN Option A 8.7.4 Configuring Inter-AS VPN Option B 8.7.5 Configuring Inter-AS VPN Option C (Solution 1) 8.7.6 Configuring Inter-AS VPN Option C (Solution 2)

Scenario	Description	Task
Configure an MCE device	An MCE device can connect to multiple VPNs. The MCE solution isolates services of different VPNs while reducing cost of CE devices.	8.7.7 Configuring an MCE Device
Configure HoVPN	HoVPN can reduce loads on PE devices. In an HoVPN networking, aggregation and access devices function as user-end provider edge (UPE) devices and work with the superstratum provider edge (SPE) devices on the backbone to provide PE functions.	8.7.8 Configuring HoVPN
Configure OSPF sham links	To ensure that VPN traffic is forwarded over the backbone network but not through backdoor routes, configure OSPF sham links between PE devices. Then routes on the MPLS VPN backbone network change into intra-area OSPF routes and can be preferred in VPN traffic forwarding.	8.7.10 Configuring an OSPF Sham Link

Scenario	Description	Task
Configure BGP/MPLS IP VPN reliability	<p>To improve VPN network reliability, you can deploy a VPN networking with full-mesh connections on the backbone network, nested PE devices on the MPLS network, and CE dual-homing (or multi-homing) on the access layer. In this networking, a BGP route reflector (RR) can be configured to reduce the number of MP-IBGP connections. This configuration mitigates loads on the network devices and facilitates device maintenance and management.</p> <p>The following technologies can also be used to improve VPN network reliability:</p> <ul style="list-style-type: none"> ● IP fast reroute (IP FRR) for VPN routes: enables traffic to be quickly switched to another PE-CE link between when the primary route is unreachable. This technology reduces the IP service interruption time. ● VPN fast reroute (VPN FRR): enables traffic to be quickly switched to another PE-PE link the primary link between them fails. This technology implements end-to-end fast convergence of VPN services. ● VPN graceful restart (VPN GR): ensures uninterrupted VPN traffic forwarding during an active/standby switchover on a PE, P, or CE device. This technology minimizes the impact of PE or CE failures on VPN services. The AR3260-S can function as both the GR restarter and GR helper, and other devices can only function as the GR helper. 	<p>8.7.11 Configuring Route Reflection to Optimize the VPN Backbone Layer</p> <p>8.7.12 Configuring IP FRR for VPN Routes</p> <p>8.7.13 Configuring VPN FRR</p> <p>8.7.14 Configuring VPN GR</p>

Scenario	Description	Task
Configure VPN tunnel policies	When VPN services need to be transmitted over a specified traffic engineering (TE) tunnel or when load balancing needs to be performed among multiple tunnels to fully use network resources, configure VPN tunnel policies.	8.7.15 Configuring Tunnel Policies
Connect VPNs to the Internet	If users in a VPN need to connect to the Internet, configure interconnection between the VPN and the Internet.	8.7.16 Connecting a VPN to the Internet

8.5 Licensing Requirements and Limitations for BGP/MPLS IP VPN

Involved Network Elements

None

License Requirements

BGP/MPLS IP VPN is a basic feature of the device and is not under license control.

Feature Limitations

When configuring BGP/MPLS IP VPN on the router, pay attention to the following points:

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S series do not supports BGP/MPLS IP VPN, only supports MCE.

8.6 Default Settings for BGP/MPLS IP VPN

This section describes the default settings for BGP/MPLS IP VPN.

[Table 8-3](#) lists the default settings for BGP/MPLS IP VPN.

Table 8-3 Default settings for BGP/MPLS IP VPN

Parameter	Default Setting
BGP/MPLS IP VPN feature	Disabled
Alarm function for BGP/MPLS IP VPN events	Disabled

Parameter	Default Setting
Number of local AS number repetitions allowed (applicable to Hub and Spoke networking)	0
Label allocation mode on PE devices	Label per route
Label allocation mode on ASBRs (inter-AS VPN)	Label per route

8.7 Configuring BGP/MPLS IP VPN

This section describes the procedures for configuring BGP/MPLS IP VPN functions.

8.7.1 Configuring Basic BGP/MPLS IP VPN Functions

Basic BGP/MPLS VPN applies to the scenario in which there is only one carrier, the MPLS backbone network belong to the same AS, and PEs, Ps, and CEs are not multi-role hosts. After basic BGP/MPLS VPN is configured, different sites in a VPN can communicate with each other.

8.7.1.1 Configuration Tasks

Table 8-4 Basic BGP/MPLS IP VPN configuration tasks

Configuration Task	Sub-task	Configuration
Configure the MPLS VPN backbone network.	Confirm requirements of VPN users.	<ol style="list-style-type: none"> Determine number of devices and interfaces based on the network scale, including: <ul style="list-style-type: none"> ● Number of users ● Number of VPNs for each user ● Number of VPN instances for each VPN Routing protocol used on the backbone network <p>NOTE When RIP-1 runs on the backbone network, you need to enable LDP to search for routes to establish LSPs based on the longest match rule. For details, see Configuring LDP Extensions for Inter-Area LSPs.</p>

Configuration Task	Sub-task	Configuration
	Configure routing between backbone devices.	<p>Configure an Interior Gateway Protocol (IGP) on the PE and P devices of the MPLS backbone network to achieve IP connectivity on the backbone network.</p> <p>For detailed configuration, see the <i>Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing</i>.</p>
	Enable MPLS on backbone devices.	<p>Enable MPLS and configure a Label Distribution Protocol (LDP) to set up public network tunnels. The LDP can be MPLS LDP or Resource Reservation Protocol-Traffic Engineering (RSVP-TE).</p> <ul style="list-style-type: none"> ● For detailed configuration, see the MPLS LDP Configuration in the <i>Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS</i>. ● For detailed configuration, see the RSVP-TE Configuration in the <i>Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS</i>. <p>You also need to configure VPN tunnel policies when VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources. For detailed configuration, see 8.7.15 Configuring Tunnel Policies.</p>
	Configure MP-IBGP between PE devices.	See 8.7.1.2 Establishing MP-IBGP Peer Relationships Between PE Devices .
Connect MPLS VPN users.	Configure VPN instances on PE devices.	See 8.7.1.3 Configuring a VPN Instance on a PE Device .
	Bind VPN instances to interfaces.	See 8.7.1.4 Binding a VPN Instance to an Interface .
	Configure route exchange between PE and CE devices.	See 8.7.1.5 Configuring Route Exchange Between PE and CE Devices .

8.7.1.2 Establishing MP-IBGP Peer Relationships Between PE Devices

Context

Perform the following steps on the PE devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** *ipv4-address* **as-number** *as-number*

The peer PE is configured as a BGP peer.

Step 4 Run **peer** *ipv4-address* **connect-interface** **loopback** *interface-number*

An interface is used to set up a Transmission Control Protocol (TCP) connection with the BGP peer.

 **NOTE**

A PE must use a loopback interface address with a 32-bit mask to set up an MP-IBGP peer relationship with the peer PE so that VPN routes can be iterated to tunnels. The route to the local loopback interface is advertised to the peer PE using an IGP on the MPLS backbone network.

Step 5 Run **ipv4-family vpnv4** [**unicast**]

The BGP-VPNv4 address family view is displayed.

Step 6 Run **peer** *ipv4-address* **enable**

The ability to exchange VPN IPv4 routes with the BGP peer is enabled.

----End

Related Tasks

When a large number of PE devices on the backbone network need to establish MP-IBGP peer relationships to exchange VPN routes, configure a route reflector (RR) to reduce the number of MP-IBGP connections between PE devices. The PE devices only need to establish MP-IBGP peer relationships with the RR. For detailed configuration, see [8.7.11 Configuring Route Reflection to Optimize the VPN Backbone Layer](#).

8.7.1.3 Configuring a VPN Instance on a PE Device

Context

In BGP/MPLS IP VPN application, each VPN has an instance to maintain forwarding information of the local VPN. Such an instance is called a VPN instance or VPN routing and forwarding table (VRF).

VPN instances isolate VPN routes from routes on the public network and isolate the routes of different VPN instances. VPN instances must be configured in all types of BGP/MPLS IP VPN networking.

Perform the following steps on each PE device.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ip vpn-instance** *vpn-instance-name*

A VPN instance is created, and its view is displayed.

NOTE

A VPN instance name is case sensitive. For example, "vpn1" and "VPN1" are different VPN instances.

Step 3 (Optional) Run **description** *description-information*

The description is configured for the VPN instance.

Step 4 (Optional) Run **service-id** *service-id*

A service ID is created for the VPN instance.

A service ID is unique on a device. It distinguishes a VPN service from other VPN services on the network.

Step 5 Run **ipv4-family**

The IPv4 address family is enabled for the VPN instance, and the VPN instance IPv4 address family view is displayed.

VPN instances support both the IPv4 and IPv6 address families. Configurations in a VPN instance can be performed only after an address family is enabled for the VPN instance based on the advertised route and forwarding data type.

Step 6 Run **route-distinguisher** *route-distinguisher*

An RD is configured for the VPN instance IPv4 address family.

A VPN instance IPv4 address family takes effect only after being configured with an RD. The RDs of different VPN instances on a PE must be different.

NOTE

- An RD can be modified or deleted only after the VPN instance is deleted or the VPN instance IPv4 address family is disabled.
- If you configure an RD for the VPN instance IPv4 address family in the created VPN instance view, the VPN instance IPv4 address family is enabled and the VPN instance IPv4 address family is displayed.

Step 7 Run **vpn-target** *vpn-target* <1-8> [**both** | **export-extcommunity** | **import-extcommunity**]

A VPN target is configured for the VPN instance IPv4 address family.

A VPN target is a BGP extended community attribute. It is used to control the receiving and advertisement of VPN routing information. A maximum of eight VPN targets can be configured using a **vpn-target** command.

Step 8 (Optional) Restrict the number of routes in a VRF.

The configuration restricts the number of routes or route prefixes imported from the attached CE devices and peer PE devices into a VPN instance on a PE device. It is recommended that you use only one of the following commands.

By default, the number of routes in a VRF is not limited as long as the total number of routes does not exceed the maximum number of unicast routes supported by the PE device.

- To set the maximum number of routes in the VPN instance IPv4 address family, run **routing-table limit** *number* { *alert-percent* | **simply-alert** }.

NOTE

The **routing-table limit** command enables the system to display a message when the number of routes added to the routing table of VPN instance IPv4 address family exceeds the limit. If you run the **routing-table limit** command to increase the maximum number of routes in the VPN instance IPv4 address family or run the **undo routing-table limit** command cancel the limit, the system adds newly received routes of various protocols to the private network IP routing table.

- To set the maximum number of route prefixes in the VPN instance IPv4 address family, run **prefix limit** *number* { *alert-percent* [**route-unchanged**] | **simply-alert** }.

NOTE

If the **prefix limit** command is run, the system gives a prompt when the number of route prefixes added to the routing table of the VPN instance IPv4 address family exceeds the limit. After the **prefix limit** command is run to increase the allowed maximum number of route prefixes in a VPN instance IPv4 address family or the **undo prefix limit** command is run to cancel the limit, the system adds newly received route prefixes of various protocols to the private network IP routing table.

After the number of route prefixes exceeds the maximum limit, direct and static routes can still be added to the IPv4 address family routing table of VPN instances.

Step 9 (Optional) Run **limit-log-interval** *interval*

The interval for logging the event that the number of routes exceeds the threshold is set for the VPN instance IPv4 address family.

If the routes or prefixes in the IPv4 address family of a VPN instance reach the maximum, the system will generate logs at intervals (defaulting to 5 seconds). To prevent logs from being displayed frequently, run this step to prolong the interval of log generation.

Step 10 (Optional) Configure a routing policy for the VPN instance.

In addition to using VPN targets to control VPN route advertisement and reception, you can configure a routing policy for the VPN instance to better control VPN routes.

- An import routing policy filters routes before they are imported into the VPN instance IPv4 address family.
- An export routing policy filters routes before they are advertised to other PE devices.

NOTE

Before applying a routing policy to a VPN instance, create the routing policy. For details about how to configure a routing policy, see Routing Policy Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Run the following command as required:

- To configure an import routing policy for the VPN instance IPv4 address family, run **import route-policy** *policy-name*.

- To configure an export routing policy for the VPN instance IPv4 address family, run **export route-policy** *policy-name*.

Step 11 (Optional) Run one of the following commands to configure the label allocation mode in the VPN instance IPv4 address family.

- Run **apply-label per-instance**

MPLS label allocation based on the VPN instance IPv4 address family (known as label per instance) is configured. One label is assigned to all the routes of the VPN instance IPv4 address family.

When a large number of VPN routes on the PE exhausts MPLS label resources, the label per instance mode saves label resources on the PE and lowers the requirement for the PE capacity.

- Run **apply-label per-route**

MPLS label allocation based on each route (known as label per route) is configured. The VPN instance address family assigns a unique label to each route to be sent to the peer PE.

When only a small number of VPN routes exists on the PE and MPLS label resources are sufficient, the label per route mode improves system security. In this way, downstream devices can load balance VPN traffic based on the inner labels of packets.

By default, the VPN instance IPv4 address family assigns the same label to all routes to be sent to the peer PE.

----End

8.7.1.4 Binding a VPN Instance to an Interface

Prerequisites

A VPN instance has been created and the IPv4 address family has been enabled for the VPN instance.

Context

- After configuring a VPN instance on a PE device, bind the VPN instance to the interface that belongs to the VPN. Otherwise, the interface functions as a public network interface and cannot forward VPN data.
- An interface becomes a private network interface after a VPN instance is bound to it. You must configure an IP address for the interface so that the PE device can exchange routing information with its attached CE device.
- After a VPN instance is bound to an interface, configuration of the Layer 3 features (IPv4 and IPv6 features) including IP addresses and routing protocols is deleted from the interface.
- When you disable an address family (IPv4 or IPv6 address family) in a VPN instance, configuration of the address family is deleted from the interface. No interface is bound to a VPN instance if no address family configuration exists in the VPN instance.

Perform the following steps on the PE devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The interface view is displayed.

Step 3 Run **ip binding vpn-instance** *vpn-instance-name*

A VPN instance is bound to the interface.

By default, an interface is a public network interface and is not associated with any VPN instance.

Step 4 Run **ip address** *ip-address { mask | mask-length }*

An IP address is configured for the interface.

----End

8.7.1.5 Configuring Route Exchange Between PE and CE Devices

Context

In BGP/MPLS IP VPN, a routing protocol or static routes must be configured between a PE and a CE to allow them to communicate and allow the CE to obtain routes to other CEs. The routing protocol can be EBGP (External/Exterior BGP), IBGP (Internal/Interior BGP), RIP (Routing Information Protocol), OSPF (Open Shortest Path First), or IS-IS (Intermediate System to Intermediate System). Choose one of the following configurations as needed:

- [Configure EBGP between a PE and a CE](#)
- [Configure IBGP between a PE and a CE](#)
- [Configure static route between a PE and a CE](#)
- [Configure RIP between a PE and a CE](#)
- [Configure OSPF between a PE and a CE](#)
- [Configure IS-IS between a PE and a CE](#)

The routing protocol configurations on the CE and PE are different:

- The CE is located at the client side. It does not know the existence of a VPN. Therefore, you do not need to configure VPN parameters when configuring a routing protocol on the CE device.
- The PE device is located at the edge of the carrier's network. It connects to a CE device and exchanges VPN routing information with other PE devices. If the CE devices that access a PE device belong to different VPNs, the PE must maintain different VRF tables. When configuring a routing protocol on the PE device, specify the name of the VPN instance to which the routing protocol applies and configure the routing protocol and MP-BGP to import routes from each other.

Configure EBGP Between a PE and a CE

Perform the following configuration on the PE device.

Table 8-5 PE configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
(Optional) Configure a unique AS number for the VPN instance IPv4 address family.	as-number <i>as-number</i>	A VPN instance uses the AS number of BGP by default. To smoothly re-assign a device to another AS or transmit different services in different instances, run this command to configure a different AS number for each VPN instance IPv4 address family. NOTE The AS number configured in the BGP-VPN instance IPv4 address family view must be different from the AS number configured in the BGP view.
Configure a CE device as a VPN peer.	peer <i>ipv4-address as-number as-number</i>	-
Set the maximum number of hops of an EBGP connection.	peer { <i>ipv4-address</i> <i>group-name</i> } ebgp-max-hop [<i>hop-count</i>]	Generally, EBGP peers are connected by a directly physical link. If no directly physical link is available, this command must be used to allow EBGP peers to establish a multi-hop TCP connection. The default value of <i>hop-count</i> is 255. If the maximum number of hops is set to 1, the PE cannot establish an EBGP connection with a peer if they are not directly connected.

Action	Command	Description
(Optional) Import direct routes destined for the local CE device into the routing table of the IPv4 VPN instance.	Use either of the following commands: <ul style="list-style-type: none"> ● import-route direct [<i>med med</i> route-policy <i>route-policy-name</i>] * ● network <i>ipv4-address</i> [<i>mask</i> <i>mask-length</i>] [route-policy <i>route-policy-name</i>] 	The PE device needs to import the routes destined for the local CE device into its VPN routing table so that it can advertise the routes to the remote PE device. NOTE The PE device can automatically learn the direct routes destined for the local CE device. The learned routes take precedence over the direct routes advertised from the local CE device using EBGp. If this step is not performed, the PE does not use MP-BGP to advertise the direct routes destined for the local CE device to the remote PE device.
(Optional) Configure the Site-of-Origin (SoO) attribute for a CE device.	peer { <i>group-name</i> <i>ipv4-address</i> } soo <i>site-of-origin</i>	Several CE devices at a VPN site may establish BGP connections with different PE devices. The VPN routes advertised from the CE devices to the PE devices may be re-advertised to the same VPN site after the routes traverse the backbone network. This may cause route loops at the VPN site. If the SoO attribute is configured for a specified CE device, the PE device adds the attribute to a route sent from the CE device and advertises the route to the remote PE. The remote PE device checks the SoO attribute of the route before sending it to its attached CE device. If the SoO attribute is the same as the local SoO attribute on the remote PE device, the remote PE device does not send the route to its attached CE device.

Action	Command	Description
(Optional) Enable BGP AS number substitution	peer <i>ipv4-address</i> substitute-as	BGP uses AS numbers to detect routing loops. Sites located at different geographical locations must be assigned different AS numbers to ensure correct transmission of routing information. If CE devices scattered at different geographical locations use the same AS number, configure BGP AS number substitution on the PE devices. NOTICE Enabling BGP AS number substitution may cause route loops in a CE multi-homing network.
(Optional) Prohibit BGP private routes from being delivered to the private IP routing table.	routing-table rib-only [route-policy <i>route-policy-name</i>]	If the BGP routing table has large numbers of VPN routes, these routes will consume large numbers of memory resources after being delivered to the IP VPN routing table. If these routes are not used in traffic forwarding, you can run the routing-table rib-only command to prevent these routes from being added to the IP VPN routing table. If some of these routes are not used in traffic forwarding, you can run the routing-table rib-only route-policy command to prevent this part of routes from being added to the IP VPN routing table. NOTICE If traffic is interrupted after the routing-table rib-only command is run, you can configure a static route or default route to guide traffic forwarding.

Perform the following configurations on the CE device.

Table 8-6 CE configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Configure the PE device as a VPN peer.	peer <i>ipv4-address</i> as-number <i>as-number</i>	-
Set the maximum number of hops of an EBGP connection.	peer { <i>ipv4-address</i> <i>group-name</i> } ebgp-max-hop [<i>hop-count</i>]	Generally, EBGP peers are connected by a directly physical link. If no directly physical link is available, this command must be used to allow EBGP peers to establish a multi-hop TCP connection. The default value of <i>hop-count</i> is 255. If the maximum number of hops is set to 1, the PE cannot establish an EBGP connection with a peer if they are not directly connected.
Import routes of the local sites.	import-route <i>protocol</i> [<i>process-id</i>] [med <i>med</i> route-policy <i>route-policy-name</i>] *	The CE device advertises the routes of its own VPN network segment to the connected PE device. The PE device forwards the routes to the remote CE device. The type of routes imported at this step may vary according to the networking mode.

Configure IBGP Between a PE and a CE

Perform the following configuration on the PE device.

Table 8-7 PE configuration

Action	Command	Description
Enter the system view.	system-view	-

Action	Command	Description
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
(Optional) Configure a unique AS number for the VPN instance IPv4 address family.	as-number <i>as-number</i>	A VPN instance uses the AS number of BGP by default. To smoothly re-assign a device to another AS or transmit different services in different instances, run this command to configure a different AS number for each VPN instance IPv4 address family. NOTE The AS number configured in the BGP-VPN instance IPv4 address family view must be different from the AS number configured in the BGP view.
Configure a CE device as a VPN peer.	peer <i>ipv4-address as-number as-number</i>	-
(Optional) Import direct routes destined for the local CE device into the routing table of the IPv4 VPN instance.	Use either of the following commands: <ul style="list-style-type: none"> ● import-route direct [<i>med med</i> route-policy <i>route-policy-name</i>] * ● network <i>ipv4-address</i> [<i>mask</i> <i>mask-length</i>] [route-policy <i>route-policy-name</i>] 	The PE device needs to import the routes destined for the local CE device into its VPN routing table so that it can advertise the routes to the remote PE device. NOTE The PE device can automatically learn the direct routes destined for the local CE device. The learned routes take precedence over the direct routes advertised from the local CE device using IBGP. If this step is not performed, the PE does not use MP-BGP to advertise the direct routes destined for the local CE device to the remote PE device.

Action	Command	Description
(Optional) Prohibit BGP private routes from being delivered to the private IP routing table.	routing-table rib-only [route-policy route-policy-name]	<p>If the BGP routing table has large numbers of VPN routes, these routes will consume large numbers of memory resources after being delivered to the IP VPN routing table. If these routes are not used in traffic forwarding, you can run the routing-table rib-only command to prevent these routes from being added to the IP VPN routing table. If some of these routes are not used in traffic forwarding, you can run the routing-table rib-only route-policy command to prevent this part of routes from being added to the IP VPN routing table.</p> <p>NOTICE If traffic is interrupted after the routing-table rib-only command is run, you can configure a static route or default route to guide traffic forwarding.</p>

Perform the following configurations on the CE device.

Table 8-8 CE configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Configure the PE device as a VPN peer.	peer <i>ipv4-address</i> as-number <i>as-number</i>	-

Action	Command	Description
Import routes of the local sites.	import-route <i>protocol</i> [<i>process-id</i>] [med <i>med</i> route-policy <i>route-policy-name</i>] *	The CE device advertises the routes of its own VPN network segment to the connected PE device. The PE device forwards the routes to the remote CE device. The type of routes imported at this step may vary according to the networking mode.

When many CE devices connect to a PE device, the PE device can function as an RR and the CE devices function as clients. This reduces the number of IBGP connections between CE devices and facilitates route maintenance and management.

Configure Static Routes Between a PE and a CE

Perform the following configuration on the PE device. The procedure for configuring static routes on the CE device is not provided here. For details about how to configure a static route, see Static Route Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 8-9 PE configuration

Action	Command	Description
Enter the system view.	system-view	-
Configure a static route for a VPN instance.	ip route-static vpn-instance <i>vpn-source-name destination-address</i> { <i>mask</i> <i>mask-length</i> } <i>interface-type interface-number</i> [<i>nexthop-address</i>] [preference <i>preference</i> tag <i>tag</i>] *	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-

Action	Command	Description
Import the configured static route to the routing table of the BGP-VPN instance IPv4 address family.	import-route static [med med route-policy route-policy-name] *	After this command is run in the BGP-VPN instance IPv4 address family view, the PE will import the VPN routes learned from the attached CE into the BGP routing table and advertise VPNv4 routes to the remote PE.

Configure RIP between a PE and a CE

Perform the following configuration on the PE device. Configure RIPv1 or RIPv2 on the CE, and the CE configuration details are not provided here. For details on how to configure RIP, see RIP Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.



NOTICE

Deleting a VPN instance or disabling a VPN instance IPv4 address family will delete all the RIP processes bound to the VPN instance or the VPN instance IPv4 address family on the PE device.

Table 8-10 PE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create a RIP process running between the PE and CE devices and enter the RIP view.	rip process-id vpn-instance vpn-instance-name	A RIP process can be bound to only one VPN instance. If a RIP process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.

Action	Command	Description
Enable RIP on the network segment of the interface to which the VPN instance is bound.	network <i>network-address</i>	-
Import BGP routes to the RIP routing table.	import-route bgp [cost { <i>cost</i> transparent } route-policy <i>route-policy-name</i>] *	After this command is executed in the RIP view, the PE device can import the VPNv4 routes learned from the remote PE device into the RIP routing table and advertise them to the attached CE device.
Return to system view.	quit	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
Import RIP routes into the routing table of the BGP-VPN instance IPv4 address family.	import-route rip <i>process-id</i> [med <i>med</i> route-policy <i>route-policy-name</i>] *	After this command is run in the BGP-VPN instance IPv4 address family view, the PE will import the VPN routes learned from the attached CE into the BGP routing table and advertise VPNv4 routes to the remote PE.

Configure OSPF Between a PE and a CE

Configure OSPF on the CE, and the CE configuration details are not provided here. Perform the following configuration on the PE device. For details on how to configure OSPF, see OSPF Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

 **NOTICE**

Deleting a VPN instance or disabling a VPN instance IPv4 address family will delete all the OSPF processes bound to the VPN instance or the VPN instance IPv4 address family on the PE device.

Table 8-11 PE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an OSPF process running between the PE and CE device and enter the OSPF view.	ospf process-id [router-id router-id] vpn-instance vpn-instance-name	An OSPF process can be bound to only one VPN instance. If an OSPF process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance. A router ID needs to be specified when an OSPF process is started after it is bound to a VPN instance. The router ID must be different from the public network router ID configured in the system view. If the router ID is not specified, OSPF selects the IP address of one of the interfaces bound to the VPN instance as the router ID based on a certain rule.

Action	Command	Description
(Optional) Configure a domain ID for the OSPF process.	domain-id <i>domain-id</i> [secondary]	The domain ID of an OSPF process is contained in the routes generated by the process. When OSPF routes are imported into BGP, the domain ID is added to the BGP VPN routes and forwarded as the BGP extended community attribute. There are no restrictions on the domain IDs of the OSPF processes of different VPNs on a PE device. The OSPF processes of the same VPN must be configured with the same domain ID to ensure proper route advertisement. The default domain ID is 0.
(Optional) Configure a VPN route tag.	route-tag <i>tag</i>	The VPN route tag prevents loops of Type-5 LSAs in CE dual-homing networking. By default, the VPN route tag is calculated using the BGP AS number. If BGP is not configured, the VPN route tag is 0.
Import BGP routes to the OSPF routing table.	import-route bgp [permit-ibgp] [cost <i>cost</i> route-policy <i>route-policy-name</i> tag <i>tag</i> type <i>type</i>] *	After this command is executed in the OSPF view, the PE can import the VPNv4 routes learned from the remote PE into the OSPF routing table and advertise them to the attached CE.
Enter the OSPF area view.	area <i>area-id</i>	-
Enable OSPF on the network segment of the interface to which the VPN instance is bound.	network <i>ip-address wildcard-mask</i>	-

Action	Command	Description
Return to the OSPF view.	quit	-
Return to system view.	quit	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
Import OSPF routes into the routing table of the BGP-VPN instance IPv4 address family.	import-route ospf <i>process-id</i> [med <i>med</i> route-policy <i>route-policy-name</i>] *	After this command is run in the BGP-VPN instance IPv4 address family view, the PE will import the VPN routes learned from the attached CE into the BGP routing table and advertise VPNv4 routes to the remote PE.

Configure IS-IS Between a PE and a CE

Configure IS-IS on the CE, and the CE configuration details are not provided here. Perform the following configuration on the PE device. For details on how to configure IS-IS, see "IPv4 IS-IS Configuration" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

NOTICE

Deleting a VPN instance or disabling a VPN instance IPv4 address family will delete all the IS-IS processes bound to the VPN instance or the VPN instance IPv4 address family on the PE device.

Table 8-12 PE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an IS-IS process running between the PE and CE devices and enter the IS-IS view.	isis process-id vpn-instance vpn-instance-name	An IS-IS process can be bound to only one VPN instance. If an IS-IS process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Set a network entity title (NET) for the IS-IS process.	network-entity net	A NET specifies the current IS-IS area address and the system ID of the router. An IS-IS process on one router can be configured with a maximum of three NETs.
(Optional) Set the level of the PE device.	is-level { level-1 level-1-2 level-2 }	By default, the IS-IS level of the router is Level-1-2.
Import BGP routes to the IS-IS routing table.	<ul style="list-style-type: none"> ● import-route bgp [cost-type { external internal } cost cost tag tag route-policy route-policy-name [level-1 level-2 level-1-2]] * ● import-route bgp inherit-cost [{ level-1 level-2 level-1-2 } tag tag route-policy route-policy-name] * 	<p>If the IS-IS level is not specified in the command, BGP routes will be imported into the Level-2 IS-IS routing table.</p> <p>After this command is executed in the ISIS view, the PE can import the VPNv4 routes learned from the remote PE into the IS-IS routing table and advertise them to the attached CE.</p>
Return to system view.	quit	-
Enter the view of the interface to which the VPN instance is bound.	interface interface-type interface-number	-

Action	Command	Description
Enable IS-IS on the interface.	isis enable [<i>process-id</i>]	-
Return to system view.	quit	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
Import IS-IS routes into the routing table of the BGP-VPN instance IPv4 address family.	import-route isis <i>process-id</i> [med <i>med</i> route-policy <i>route-policy-name</i>] *	After this command is run in the BGP-VPN instance IPv4 address family view, the PE will import the VPN routes learned from the attached CE into the BGP routing table and advertise VPNv4 routes to the remote PE.

8.7.1.6 Verifying the Configuration of Basic BGP/MPLS IP VPN Functions

Prerequisites

All configurations for a basic BGP/MPLS IP VPN are complete.

Procedure

- Run the following commands on the PE to check information about the created VPN instance IPv4 address family, including the RD and other attributes.
 - Run the **display ip vpn-instance** *vpn-instance-name* command to check brief information about a specified VPN instance.
 - Run the **display ip vpn-instance verbose** *vpn-instance-name* command to check detailed information about a specified VPN instance.
 - Run the **display ip vpn-instance import-vt** *ivt-value* command to check information about the VPN instances with the specified import VPN target.
 - Run the **display ip vpn-instance** [*vpn-instance-name*] **interface** command to view information about the interface bound to a specified VPN instance.

- Run the following commands on the PE and CE to check information about the IPv4 VPN routes to the local and remote sites.
 - Run the **display ip routing-table vpn-instance** *vpn-instance-name* command on the PE to check the routing information of a specified VPN instance IPv4 address family.
 - Run the **display ip routing-table** command on the CE to check routing information.

----End

8.7.2 Configuring Hub and Spoke

In Hub and Spoke networking, a central site is deployed and all the other sites communicate through the central site. The central site controls communication between sites.

Pre-configuration Tasks

Before configuring Hub and Spoke, complete the following tasks:

- Configuring IGP on PE devices and P devices in the MPLS backbone network



When RIP-1 runs on the backbone network, you need to enable LDP to search for routes to establish LSPs based on the longest match rule. For details, see [Configuring LDP Extensions for Inter-Area LSPs](#).

- Configuring basic MPLS capabilities and MPLS LDP (or RSVP-TE) on PE devices and P devices in the MPLS backbone network
- Configuring the IP addresses, through which the CE devices access the PE devices, on the CE devices



You also need to configure VPN tunnel policies when VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources. For detailed configuration, see [8.7.15.1 Configuring and Applying a Tunnel Policy](#).

Configuration Procedure

All the following tasks are mandatory. Perform these tasks in this sequence to complete the Hub and Spoke configuration.

8.7.2.1 Configuring MP-IBGP Between Hub-PE and Spoke-PE

Context

The Hub-PE must set up the MP-IBGP peer with all the Spoke-PE devices. Spoke-PE devices do not need to set up the MP-IBGP peer between each other.

Perform the following steps on the Hub-PE and Spoke-PE devices.

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** *ipv4-address* **as-number** *as-number*

The peer PE is configured as a BGP peer.

Step 4 Run **peer** *ipv4-address* **connect-interface** **loopback** *interface-number*

An interface is used to set up a Transmission Control Protocol (TCP) connection with the BGP peer.

 **NOTE**

A PE must use a loopback interface address with a 32-bit mask to set up an MP-IBGP peer relationship with the peer PE so that VPN routes can be iterated to tunnels. The route to the local loopback interface is advertised to the peer PE using an IGP on the MPLS backbone network.

Step 5 Run **ipv4-family** **vpn4** [**unicast**]

The BGP-VPNv4 address family view is displayed.

Step 6 Run **peer** *ipv4-address* **enable**

The ability to exchange VPN IPv4 routes with the BGP peer is enabled.

----End

8.7.2.2 Configuring VPN Instances on PE Devices

Context

Configure VPN instances on each Spoke-PE device and the Hub-PE device. This section provides only the mandatory configuration for a VPN instance. For the optional configuration of a VPN instance, see [8.7.1.3 Configuring a VPN Instance on a PE Device](#).

Procedure

- Configure VPN instances on the Hub-PE device.

Configure the following two VPN instances for the Hub-PE device:

- VPN-in: accepts and maintains all the VPNv4 routes advertised by all the Spoke-PE devices.
- VPN-out: maintains the routes of the Hub site and all the Spoke sites and advertises those routes to all the Spoke-PE devices.

a. Run **system-view**

The system view is displayed.

b. Run **ip vpn-instance** *VPN-in*

The *VPN-in* instance is created and the *VPN-in* instance view is displayed.

c. Run **ipv4-family**

The IPv4 address family is enabled for the *VPN-in* instance, and the *VPN-in* instance IPv4 address family view is displayed.

- d. Run **route-distinguisher** *route-distinguisher*
The RD of the *VPN-in* instance IPv4 address family is configured.
- e. Run **vpn-target** *vpn-target1* &<1-8> **import-extcommunity**
The VPN target extended community for the *VPN-in* instance IPv4 address family is created to import the VPNv4 routes advertised by all the Spoke-PE devices.
vpn-target1 lists the Export VPN targets advertised by all the Spoke-PE devices.
- f. Run **quit**
The VPN instance view is displayed.
- g. Run **quit**
Return to the system view.
- h. Run **ip vpn-instance** *VPN-out*
The *VPN-out* instance is created and the *VPN-out* instance view is displayed.
- i. Run **ipv4-family**
The IPv4 address family is enabled for the *VPN-out* instance, and the *VPN-out* instance IPv4 address family view is displayed.
- j. Run **route-distinguisher** *route-distinguisher*
The RD of the *VPN-out* instance IPv4 address family is configured.
- k. Run **vpn-target** *vpn-target2* &<1-8> **export-extcommunity**
The VPN target extended community for the *VPN-out* instance IPv4 address family is created to advertise the routes of all the Hubs and Spokes.
vpn-target2 lists the Import VPN targets advertised by all the Spoke-PE devices.
- Configure a Spoke-PE device.
Every Spoke-PE device is configured with a VPN instance.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **ip vpn-instance** *vpn-instance-name*
The VPN instance view of VPN-in is displayed.
 - c. Run **ipv4-family**
The VPN instance IPv4 address family view is displayed.
 - d. Run **route-distinguisher** *route-distinguisher*
The RD of the *VPN-in* instance is configured.
 - e. Run **vpn-target** *vpn-target2* &<1-8> **import-extcommunity**
The VPN target extended community is configured for the VPN instance IPv4 address family to receive the VPNv4 routes advertised by the Hub-PE device.
vpn-target2 must be in the export VPN target list configured on the Hub-PE device.
 - f. Run **vpn-target** *vpn-target1* &<1-8> **export-extcommunity**
The VPN target extended community is configured for the VPN instance IPv4 address family to advertise the routes of Spoke sites.

vpn-target1 must be in the import VPN target list configured on the Hub-PE device.

----End

8.7.2.3 Binding a VPN Instance to an Interface

Prerequisites

A VPN instance has been created and the IPv4 address family has been enabled for the VPN instance.

Context

The configuration on the Hub-PE involves two interfaces or sub-interfaces: one is bound with the VPN-in and receives the routes advertised by the Spoke-PE; the other is bound with the VPN-out and advertises the routes of the Hub and all the Spokes.

- After configuring a VPN instance on a PE device, bind the VPN instance to the interface that belongs to the VPN. Otherwise, the interface functions as a public network interface and cannot forward VPN data.
- An interface becomes a private network interface after a VPN instance is bound to it. You must configure an IP address for the interface so that the PE device can exchange routing information with its attached CE device.
- After a VPN instance is bound to an interface, configuration of the Layer 3 features (IPv4 and IPv6 features) including IP addresses and routing protocols is deleted from the interface.
- When you disable an address family (IPv4 or IPv6 address family) in a VPN instance, configuration of the address family is deleted from the interface. No interface is bound to a VPN instance if no address family configuration exists in the VPN instance.

Perform the following steps on the Hub-PE and all the Spoke-PE devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The interface view is displayed.

Step 3 Run **ip binding vpn-instance** *vpn-instance-name*

A VPN instance is bound to the interface.

By default, an interface is a public network interface and is not associated with any VPN instance.

Step 4 Run **ip address** *ip-address { mask | mask-length }*

An IP address is configured for the interface.

----End

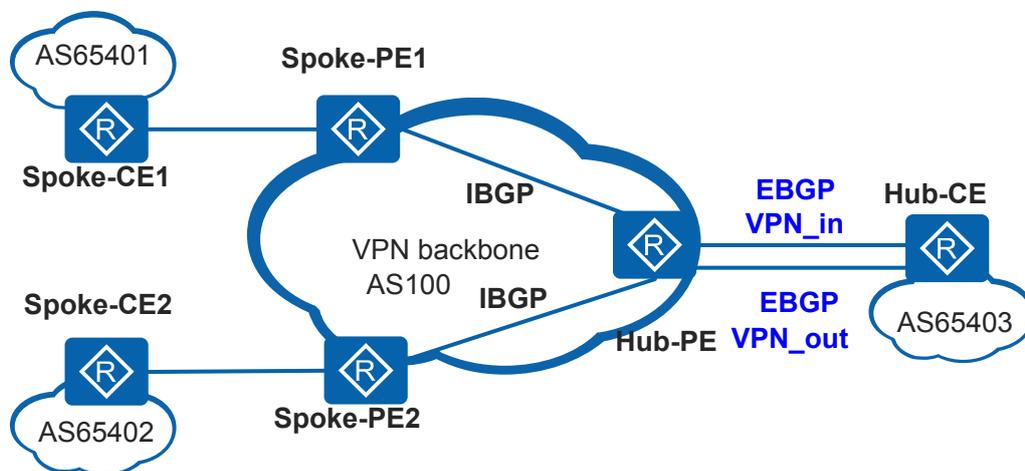
8.7.2.4 Configuring Route Exchange Between PE device and CE Devices

Context

The Hub-PE and Hub-CE devices can use IGP or EBGP to exchange routing information. When they use EBGP, you must configure the Hub-PE device to allow repeated local AS number.

As shown in **Figure 8-37**, the routing information advertised by a Spoke-CE is forwarded to the Hub-CE and Hub-PE device before being transmitted to other Spoke-PE devices. If EBGP runs between the Hub-PE device and the Hub-CE, the Hub-PE device performs the AS-Loop detection on the route. If the Hub-PE device detects its own AS number in the route, it discards the route. In this case, to implement the Hub and Spoke networking, the Hub-PE device must be configured to permit the existence of repeated local AS numbers.

Figure 8-37 EBGP running between the Hub-CE and Hub-PE devices



Procedure

- Configure EBGP between the Hub-PE and Hub-CE devices.

For detailed configuration procedures, see [Configuring a Routing Protocol Between PE device and CE](#).

The Spoke-PE and Spoke-CE devices can use EBGP, IGP, or static routes.

To set up an EBGP peer relationship between the Hub-PE and Hub-CE devices and between a Spoke-PE device and a Spoke-CE device, perform the following steps on the Hub-PE device:

- Run **system-view**
 The system view is displayed.
- Run **bgp { as-number-plain | as-number-dot }**
 The BGP view is displayed.
- Run **ipv4-family vpn-instance vpn-instance-name**

The BGP-VPN instance IPv4 address family view is displayed.

- d. Run **peer ip-address allow-as-loop** [*number*]

The Hub-PE is configured to allow the routing loop. Here the value of *number* is set as 1, which means the route with the AS repeated once can be sent.

- Configure an IGP between the Hub-PE and Hub-CE devices.

For detailed configuration procedures, see [Configuring a Routing Protocol Between PE and CE](#).

In this way, instead of BGP, IGP or static routes are adopted between the Spoke-PE and the Spoke-CE. If BGP is used, the source BGP route's AS number will get lost when the route is transmitted through the IGP running between the Hub-PE and Hub-CE. The Spoke-PE will receive both the source BGP route sent by the Spoke-CE and the source BGP route with no AS number forwarded by the Hub-PE. The source BGP route sent by the Spoke-CE has an AS number and is therefore not preferred by the Spoke-PE. After the route is withdrawn, the Spoke-PE prefers the source BGP route received from the Spoke-CE again and advertises this route again. As this process repeats, route flapping occurs.

- Configure static routes between the Hub-PE and the Hub-CE devices.

For detailed configuration procedures, see [Configuring a Routing Protocol Between PE device and CE](#).

EBGP, IGP, or static routes can be used between the Spoke-PE and the Spoke-CE devices.

If the Hub-CE device uses the default route to access the Hub-PE device, perform the following steps on the Hub-PE device to advertise the default route to all the Spoke-PE devices:

- a. Run **system-view**

The system view is displayed.

- b. Run **ip route-static vpn-instance vpn-source-name 0.0.0.0 0.0.0.0 nexthop-address** [**preference preference** | **tag tag**]* [**description text**]

Here, *vpn-instance-name* refers to the VPN-out. *nexthop-address* is the IP address of the Hub-CE interface that is connected with the PE device interface bound with the VPN-out instance.

- c. Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

- d. Run **ipv4-family vpn-instance vpn-instance-name**

The BGP-VPN instance IPv4 address family view is displayed. *vpn-instance-name* refers to the VPN-out instance.

- e. Run **network 0.0.0.0 0**

The default route is advertised to all the Spoke-PE devices through MP-BGP.

----End

8.7.2.5 Verifying the Hub and Spoke Configuration

Prerequisites

The configurations of the Hub and Spoke function are complete.

Procedure

- Run the **display ip routing-table vpn-instance** *vpn-instance-name* command to check routing information about the VPN-in and VPN-out on the Hub-PE.

If the VPN-in routing table has routes to all the Spoke stations, and the VPN-out routing table has routes to the Hub and all the Spoke stations, it means the configuration is successful.

- Run the **display ip routing-table** command to check routing information on the Hub-CE and all the Spoke-CE devices.

The Hub-CE and all the Spoke-CE devices have routes to the Hub and all the Spoke sites.

----End

8.7.3 Configuring Inter-AS VPN Option A

If the MPLS VPN backbone network spans multiple ASs, inter-AS VPN is required. Inter-AS VPN-Option A can be used when each PE device has a few VPNs and VPN routes.

Procedure

To implement inter-AS VPN Option A, complete basic BGP/MPLS IP VPN configuration in each AS and configure the ASBR-PE devices as the CE device of each other. You need to configure VPN instances for a PE device and an ASBR-PE device respectively. The PE device connects to CE devices, and the ASBR-PE device connects to the remote ASBR-PE device. For details about basic BGP/MPLS IP VPN configuration, see [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#).

NOTE

In inter-AS VPN Option A, the VPN targets of VPN instances on the ASBR and PE devices in the same AS must match for the same VPN. This is not required for the PE devices in different ASs.

Verifying the Configuration

After inter-AS VPN Option A is configured, run the following commands to check previous configurations.

- Run the **display bgp vpnv4 all peer** command on the PE or ASBR, and you can view that the status of the BGP VPNv4 peer relationship between the PE and ASBR in the same AS is "Established".
- Run the **display bgp vpnv4 all routing-table** command on the PE or ASBR, and you can view the VPNv4 routes.
- Run the **display ip routing-table vpn-instance** *vpn-instance-name* command on the PE or ASBR, and you can view that the VPN routing table of the PE or ASBR has related VPN routes.

8.7.4 Configuring Inter-AS VPN Option B

If virtual private network (VPN) routes need to be established over a Multiprotocol Label Switching (MPLS) backbone network spanning multiple autonomous areas (ASs), inter-AS

VPN is required. If the provider edge (PE) devices connect to many VPNs but the autonomous area border routers (ASBRs) do not have enough interfaces to reserve an interface for each inter-AS VPN, the inter-AS VPN Option B solution can be used on the network.

Pre-configuration Tasks

Before configuring inter-AS VPN Option B, complete the following tasks:

- Configuring an Interior Gateway Protocol (IGP) for the MPLS backbone network of each AS to ensure IP connectivity on the backbone network within each AS
- Configuring the basic MPLS functions and MPLS Label Distribution Protocol (LDP) or Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for the MPLS backbone network of each AS
- In each AS, configuring VPN instances on the PE devices connected to CE devices and associating the VPN instances with PE interfaces connected to CE devices
- Configuring route exchange between the PE and CE devices in each AS

For details about the configurations, see [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#).

Configuration Procedure

[8.7.4.4 \(Optional\) Configuring Routing Policies to Control VPN Route Advertisement and Acceptance](#) and [8.7.4.5 \(Optional\) Enabling Next-Hop-based Label Allocation on the ASBR](#) are optional, and other tasks are mandatory. Perform these tasks in this sequence to complete inter-AS VPN Option B configuration.

When VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources, you also need to complete the task of [8.7.15 Configuring Tunnel Policies](#).

NOTE

In inter-AS VPN Option B, the ASBRs maintain and advertise VPNv4 routes of inter-AS VPNs, and they can also work as PE devices. When the ASBRs work as PE devices, configure VPN instances on the ASBRs to enable them to exchange routing information with CE devices. The configuration is the same as that on common PE devices.

8.7.4.1 Configuring MP-IBGP Between PE and ASBR in the Same AS

Context

Perform the following steps on the PE and ASBR in the same AS.

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run `bgp { as-number-plain | as-number-dot }`

The BGP view is displayed.

Step 3 Run **peer** *ipv4-address* **as-number** *as-number*

The peer ASBR is specified as the IBGP peer.

Step 4 Run **peer** *ipv4-address* **connect-interface loopback** *interface-number*

The loopback interface is specified as the outgoing interface of the BGP session.

 **NOTE**

The 32-bit mask IP addresses of the loopback interfaces must be used to establish the MP-IBGP peer relationship between PEs. This can ensure that the tunnel can be iterated. The route destined to the loopback interface is advertised to the remote PE based on IGP on the MPLS backbone network.

Step 5 Run **ipv4-family vpnv4 [unicast]**

The BGP-VPNv4 address family is displayed.

Step 6 Run **peer** *ipv4-address* **enable**

The exchange of VPNv4 routes between the PE and ASBR in the same AS is enabled.

 **NOTE**

When the ASBR sends a VPNv4 route to a PE, the ASBR can automatically change the next hop in the VPNv4 route to the IP address of itself.

----End

8.7.4.2 Configuring MP-EBGP Between ASBRs in Different ASs

Context

In inter-AS VPN Option B, you need not create VPN instances on ASBRs. The ASBR does not filter the VPNv4 routes received from the PE in the same AS based on VPN targets. Instead, it advertises the received VPNv4 routes to the peer ASBR through MP-EBGP.

In the AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S, an ASBR can only change the next-hop address of a VPNv4 route to the ASBR's address before advertising the route to a PE.

Perform the following steps on the ASBR.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type* *interface-number*

The view of the interface connected with the ASBR interface is displayed.

Step 3 Run **ip address** *ip-address* { *mask* | *mask-length* }

The interface IP address is configured.

Step 4 Run **mpls**

The MPLS capability is enabled.

Step 5 Run **quit**

Return to the system view.

Step 6 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 7 Run **peer** *ipv4-address* **as-number** *as-number*

The peer ASBR is specified as the EBGP peer.

Step 8 (Optional) Run **peer** { *ipv4-address* | *group-name* } **ebgp-max-hop** [*hop-count*]

The maximum number of hops is configured for the EBGP connection.

Generally, one or multiple directly connected physical links exist between EBGP peers. If the directly connected physical link(s) are not available, run this command to ensure that the TCP connection can be set up between the EBGP peers through multiple hops.

Step 9 Run **ipv4-family vpnv4** [**unicast**]

The BGP-VPNv4 address family is displayed.

Step 10 Run **peer** *ipv4-address* **enable**

The exchange of IPv4 VPN routes with the peer ASBR is enabled.

----End

8.7.4.3 Disabling an ASBR from Filtering VPNv4 Routes by VPN Targets

Context

By default, the PE performs VPN target filtering on the received IPv4 VPN routes. The routes passing the filter are added to the routing table, and the others are discarded. If the PE is not configured with VPN instance, or the VPN instance is not configured with the VPN target, the PE discards all the received VPN IPv4 routes.

In Inter-AS VPN Option B, you do not need to configure VPN instances on the ASBRs. An ASBR must save all the VPNv4 routes and advertises the VPNv4 routes to the remote ASBR. In this case, the ASBR must accept all the VPNv4 routing information without the VPN target filtering.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **ipv4-family vpnv4** [**unicast**]

The BGP-VPNv4 address family is displayed.

Step 4 Run **undo policy vpn-target**

The VPN IPv4 routes are not filtered by the VPN target.

---End

8.7.4.4 (Optional) Configuring Routing Policies to Control VPN Route Advertisement and Acceptance

Context

The ASBRs accept all VPNv4 routes after they are configured not to filter VPNv4 routes by VPN targets. When there are many VPN routes on the network, the ASBRs are overburdened.

If only some of VPNs or sites need to communicate across ASs, you can configure a routing policy on the ASBRs to restrict the VPNv4 routes that can be accepted by the ASBRs. This reduces loads on the ASBRs.

This section describes how to configure the following filtering policies to control VPNv4 route advertisement and acceptance:

- Filtering policy based on VPN targets
- Filtering policy based on RDs

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run either of the following command to configure a route filter.

1. To configure an extended community filter, run **ip extcommunity-filter** *extcomm-filter-number* { **permit** | **deny** } { **rt** { *as-number:nn* | *ipv4-address:nn* } } &<1-16>.
2. To configure an RD filter, run **ip rd-filter** *rd-filter-number* { **deny** | **permit** } *route-distinguisher* &<1-10>.

Step 3 Run **route-policy** *route-policy-name* **permit** **node** *node*

A routing policy is configured.

Step 4 Run either of the following command to configure an if-match clause in the configured route filter:

1. If you configured an extended community filter in [Step 2](#), run the **if-match extcommunity-filter** { { *basic-extcomm-filter-num* | *advanced-extcomm-filter-num* } &<1-16> | *advanced-extcomm-filter-name* | *basic-extcomm-filter-name* } command to configure an if-match clause based on the extended community filter in the routing policy.
2. If you configured an RD filter in [Step 2](#), run the **if-match rd-filter** *rd-filter-number* command to configure an if-match clause based on the RD filter in the routing policy.

Step 5 Run **quit**

Return to the system view.

Step 6 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 7 Run `ipv4-family vpnv4 [unicast]`

The BGP-VPNv4 address family is displayed.

Step 8 Run `peer ipv4-address route-policy route-policy-name { export | import }`

The routing policy is applied to controlling the VPN IPv4 routing information.

---End

8.7.4.5 (Optional) Enabling Next-Hop-based Label Allocation on the ASBR

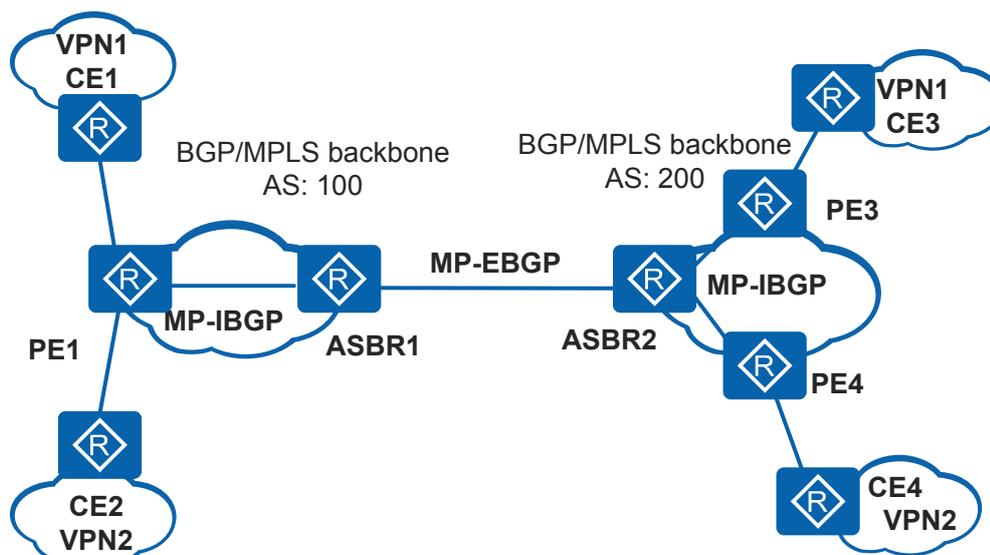
Context

In an inter-AS VPN Option B scenario, ASBRs can be enabled to allocate labels to VPN routes based on next hops. This saves labels on the ASBRs.

Next-hop-based label allocation means to allocate the same label for the routes with the same forwarding behavior. In other words, VPN routes with the same forwarding path and outbound label are assigned the same label. Different from the prefix-based label allocation mode that is used by default, next-hop-based label allocation enrich the label allocation modes and allows for flexible label allocation. In addition, when an ASBR functions as a PE device, next-hop-based label allocation can be used together with one label per instance mode to save labels on the ASBR.

As shown in [Figure 8-38](#), the inter-AS VPN Option B networking is established; two VPN instances, VPN1 and VPN2, are configured on PE1; the label allocation mode is one label per VPN instance. CE1 in VPN1 and CE2 in VPN2 are respectively imported with 1 thousand VPN routes. When the next-hop-based label allocation feature is not enabled for VPN routes on ASBRs, the 2 thousand routes of PE1 advertised by ASBR1 to ASBR2 use 2 thousand labels; after the next-hop-based label allocation feature is enabled for VPN routes on ASBR1, ASBR1 only assigns one label for VPN routes of the same next hop and outgoing label. As a result, ASBR1 needs to allocate only two labels for 2 thousands routes.

Figure 8-38 Next-hop-based label allocation for VPN routes on ASBR





NOTICE

After next-hop-based label allocation is enabled or disabled, the label allocated by the ASBR for a route changes, which leads to packet loss.

Perform the following steps on the ASBR.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp { as-number-plain | as-number-dot }**

The BGP view is displayed.

Step 3 Run **ipv4-family vpnv4**

The BGP-VPNv4 view is displayed.

Step 4 Run **apply-label per-nexthop**

The next-hop-based label allocation for IPv4 VPN routes is enabled on the ASBR.

----End

8.7.4.6 Verifying the Inter-AS VPN Option B Configuration

Prerequisites

The configuration of inter-AS VPN Option B is complete.

Procedure

- Run the **display bgp vpnv4 all peer** command on the PE or ASBR. If the status of the IBGP peer between the PE and ASBR in the same AS is "Established", and the status of the EBGP peer between ASBRs in the different AS is "Established", the configuration is successful.
- Run the **display bgp vpnv4 all routing-table** command on the ASBR. If the VPN IPv4 routes are displayed, the configuration is successful.
- Run the **display ip routing-table vpn-instance vpn-instance-name** command on the PE device. If the VPN routes are displayed, the configuration is successful.
- Run the **display mpls lsp** command on the ASBR. If information about the LSP and label is displayed, it means that the configuration succeeds. If the ASBR is enabled with the next-hop-based label allocation, only one label is allocated for the VPN routes with the same next hop and outgoing label.
- Run the **display ip extcommunity-filter** command on an ASBR to check the configured extended community filters.
- Run the **display ip rd-filter** command on an ASBR to check the configured RD filters.

----End

8.7.5 Configuring Inter-AS VPN Option C (Solution 1)

If virtual private network (VPN) routes need to be established over a Multiprotocol Label Switching (MPLS) backbone network spanning multiple autonomous areas (ASs), inter-AS VPN is required. If each AS needs to exchange a large number of VPN routes, inter-AS VPN Option C is a good choice to prevent the autonomous area border routers (ASBRs) from becoming bottlenecks that impede network expansion.

Pre-configuration Tasks

Before configuring inter-AS VPN Option C, complete the following tasks:

- Configuring an Interior Gateway Protocol (IGP) for the MPLS backbone network of each AS to ensure IP connectivity on the backbone network within each AS
- Configuring the basic MPLS functions and MPLS Label Distribution Protocol (LDP) or Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for the MPLS backbone network of each AS
- In each AS, configuring VPN instances on the PE devices connected to CE devices and associating the VPN instances with PE interfaces connected to CE devices
- Configuring route exchange between the PE and CE devices in each AS

For details about the configurations, see [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#).

Context

The following solutions can be used to implement inter-AS VPN Option C:

- Solution 1: After learning the labeled BGP routes of the public network in the remote AS from the remote ASBR, the local ASBR allocates labels for these routes, and advertises these routes to the IBGP peer that supports the label switching capability. In this manner, a complete LSP is set up.
- Solution 2: The IBGP peer relationship between the PE and ASBR is not needed. In this solution, an ASBR learns the labeled public BGP routes of the remote AS from the peer ASBR. Then these labeled public BGP routes are imported to an IGP to trigger the establishment of an LDP LSP. In this manner, a complete LDP LSP can be established between the two PEs.

Solution 1 is described in this section, and solution 2 is described in [8.7.6 Configuring Inter-AS VPN Option C \(Solution 2\)](#).

Configuration Procedure

All the following tasks are mandatory. Perform these tasks in this sequence to complete inter-AS VPN Option C configuration.

When VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources, you also need to complete the task of [8.7.15 Configuring Tunnel Policies](#).

NOTE

In inter-AS VPN Option C mode, do not enable LDP between ASBRs. If LDP is enabled on the interfaces between ASBRs, LDP sessions are then established between the ASBRs. When a lot of BGP routes exist, many LDP labels are occupied.

8.7.5.1 Enabling the Labeled IPv4 Route Exchange

Context

In inter-AS VPN Option C, establish an inter-AS VPN LSP. The related PEs and ASBRs exchange public network routes with the MPLS labels.

The public network routes with the MPLS labels are advertised by the MP-BGP. Based on RFC 3107 (Carrying Label Information in BGP-4), the label mapping information of a route is carried by advertising BGP updates. This feature is implemented through BGP extension attributes, which requires BGP peers to process the labeled IPv4 routes.

By default, BGP peers cannot process labeled IPv4 routes.

Procedure

- Configure a PE device.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **bgp { as-number-plain | as-number-dot }**

The BGP view is displayed.
 - c. Run **peer ipv4-address as-number as-number**

An IBGP peer relationship is established between the local PE and ASBR in the same AS.
 - d. Run **peer ipv4-address connect-interface loopback interface-number**

A loopback interface is specified as the outbound interface of the BGP session.
 - e. Run **peer ipv4-address label-route-capability**

Exchange of the labeled IPv4 routes with the ASBR in the same AS is enabled.
- Configure an ASBR.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **interface interface-type interface-number**

The view of the interface connected with the peer ASBR is displayed.
 - c. Run **ip address ip-address { mask | mask-length }**

The interface IP address is configured.
 - d. Run **mpls**

The MPLS capability is enabled.
 - e. Run **quit**

Return to the system view.
 - f. Run **bgp { as-number-plain | as-number-dot }**

The BGP view is displayed.
 - g. Run **peer ipv4-address as-number as-number**

An IBGP peer relationship is established between the local PE and the remote PE in the same AS.

- h. Run **peer *ipv4-address* connect-interface loopback *interface-number***

A loopback interface is specified as the outbound interface of the BGP session.

- i. Run **peer *ipv4-address* label-route-capability**

Exchange of the labeled IPv4 routes with the remote PE in the same AS is enabled.

- j. Run **peer *ipv4-address* as-number *as-number***

The peer ASBR is specified as the EBGP peer.

- k. (Optional) Run **peer { *ipv4-address* | *group-name* } ebgp-max-hop [*hop-count*]**

The maximum number of hops is configured for the EBGP connection.

Generally, one or multiple directly connected physical links exist between EBGP peers. If the directly connected physical links are not available, run the **peer ebgp-max-hop** command to ensure that the TCP connection can be set up between the EBGP peers through multiple hops.

If BGP uses a loopback interface to establish an EBGP peer relationship, you must run the **peer ebgp-max-hop** command and set the hop count to a value larger than or equal to 2. Otherwise, the peer relationship cannot be established. If *hop-count* is not specified, the default value 255 is used.

- l. Run **peer *ipv4-address* label-route-capability [check-tunnel-reachable]**

The exchange of the labeled IPv4 routes with the peer ASBR is enabled.

- If tunnel reachability checking is enabled, BGP advertises IPv4 unicast routes to peers when routed tunnels are unreachable or advertises labeled routes to peers when routed tunnels are reachable. This eliminates the risk of establishing an MP-EBGP peer relationship between PEs over a faulty LSP because this will cause data forwarding failures.
- If tunnel reachability checking is disabled, BGP advertises labeled routes to peers whether the tunnels for imported routes are reachable or not.

---End

8.7.5.2 Configuring a Routing Policy to Control Label Distribution

Context

You need to configure a routing policy to control label allocation for each inter-AS BGP LSP. If labeled IPv4 routes are advertised to a PE of the local AS, you need to re-allocate MPLS labels to these routes. If routes sent by a PE of the local AS are advertised to the peer ASBR, you need to allocate MPLS labels to these routes.

Procedure

- Step 1** Create a routing policy.

Perform the following steps on the ASBR.

1. Run **system-view**

- The system view is displayed.
2. Run **route-policy** *policy-name1* **permit node** *node*
The routing policy applied to the local PE is created.
For the labeled IPv4 routes received from peer ASBRs, and sent to the PEs in the same AS, this policy ensures that a new MPLS label is allocated.
 3. Run **if-match** **mpls-label**
The IPv4 routes with labels are matched.
 4. Run **apply** **mpls-label**
The label is allocated to the IPv4 route.
 5. Run **quit**
Return to the system view.
 6. Run **route-policy** *policy-name2* **permit node** *node*
The routing policy applied to the peer ASBR is created.
For the labeled IPv4 routes received from PE in the local AS, and sent to the remote ASBR, this policy ensures that a new MPLS label is allocated.
 7. Run **apply** **mpls-label**
The label is allocated to the IPv4 route.

Step 2 Apply the routing policy.

Perform the following steps on the ASBR.

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **peer** *ipv4-address* **route-policy** *policy-name1* **export**
The routing policy adopted when the route is advertised to the local PE is created.
4. Run **peer** *ipv4-address* **route-policy** *policy-name2* **export**
The routing policy adopted when the route is advertised to the peer ASBR is created.

Step 3 (Optional) Control the creation of ingress LSPs for labeled BGP routes based on routing policies.

Perform the following steps on each PE.

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **ingress-lsp trigger** **route-policy** *route-policy-name*
The function to create ingress LSPs for labeled BGP routes based on routing policies is configured.

On a MAN where the hybrid access mode is used, a large number of labeled BGP routes are used to establish end-to-end LSPs. On certain intermediate nodes where VPN services do not need to be supported, excessive ingress LSPs are created, wasting network resources. In this case, you can run the **ingress-lsp trigger** command to create ingress LSPs based on a routing policy to save network resources.

---End

8.7.5.3 Establishing an MP-EBGP Peer Relationship Between PE Devices

Context

By introducing extended community attributes into BGP, MP-EBGP can advertise VPNv4 routes between PEs.

Procedure

- Configure a PE device to advertise its loopback interface IP addresses used for peer relationship establishment to the ASBRs of other ASs and peer PE devices. You can also configure an ASBR to send the loopback interface IP addresses of a PE device used for peer relationship establishment to the ASBRs of other ASs and peer PE devices.

- a. Run **system-view**

The system view is displayed.

- b. Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

- c. Run **network** *ip-address* [*mask* | *mask-length*] [**route-policy** *route-policy-name*]

The loopback address of the PE in the local AS is advertised to the remote ASBR.

- (Optional) Disable an ASBR from advertising BGP supernet labeled routes.

In an inter-AS VPN Option C scenario, a PE uses a routing policy to assign a label to its loopback address route and advertises this route as a BGP labeled route. When an ASBR receives the route, the route is a BGP supernet labeled route in which the destination address and next hop address are the same or the destination address is more detailed than the next hop address. In V2R3C00 or earlier, the ASBR does not advertise the received BGP supernet labeled route. After the ASBR is upgraded to a version later than V2R3C00, the ASBR can advertise the received BGP supernet labeled route to other BGP peers. This advertisement may change the traffic path on the network before and after the upgrade. To ensure that the traffic path remains unchanged, disable the ASBR from advertising BGP supernet labeled routes.

- a. Run **system-view**

The system view is displayed.

- b. Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

- c. Run **supernet label-route advertise disable**

The ASBR is disabled from advertising BGP supernet labeled routes.

After you disable the ASBR from advertising BGP supernet labeled routes, to advertise the loopback address route of a PE in the local AS to a PE in another AS,

run the **network** command on the ASBR to advertise the BGP route to the loopback address of the PE in the same AS.

- Perform the following steps on the PE device:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
 - c. Run **peer** *ipv4-address* **as-number** { *as-number-plain* | *as-number-dot* }
The peer PE is specified as the EBGP peer.
 - d. Run **peer** *ipv4-address* **connect-interface loopback** *interface-number*
The source interface that sends BGP packets is specified.
 - e. Run **peer** *ipv4-address* **ebgp-max-hop** [*hop-count*]
The maximum hop of the EBGP peer is configured.

PEs of different ASs are generally not directly connected. To set up the EBGP peer between PEs of different ASs, configure the maximum hop between PEs and ensure the PEs are reachable.

- f. (Optional) Run **peer** { *group-name* | *ipv4-address* } **mpls-local-ifnet disable**

The ability to establish an MPLS local IFNET tunnel between PEs is disabled.

In the Option C scenario, PEs establish an MP-EBGP peer relationship. Therefore, an MPLS local IFNET tunnel between PEs is established over the MP-EBGP peer relationship. The MPLS local IFNET tunnel fails to transmit traffic because PEs are indirectly connected.

If a fault occurs on the BGP LSP between PEs, traffic is iterated to the MPLS local IFNET tunnel, not an FRR bypass tunnel. As the MPLS local IFNET tunnel cannot forward traffic, traffic is interrupted. To prevent the traffic interruption, run this command to disable the establishment of an MPLS local IFNET tunnel between PEs.

- g. Run **ipv4-family vpnv4** [**unicast**]
The BGP VPNv4 address family is displayed.
- h. Run **peer** *ipv4-address* **enable**
The exchange of VPN IPv4 routes with the peer PE is enabled.

----End

Related Tasks

To improve scalability, specify an RR in each AS and establish MP-EBGP peer relationships between the RRs in ASs to save all VPNv4 routes on the RRs. Then configure PEs in each AS as the RR's clients to exchange VPNv4 routing information with the RR. The configuration is as follows:

- Configure a PE device to advertise its loopback interface IP addresses used for peer relationship establishment to the ASBRs of other ASs and peer PE devices. You can also configure an ASBR to send the loopback interface IP addresses of a PE device used for

peer relationship establishment to the ASBRs of other ASs and peer PE devices. The configuration procedure is the same as the above mentioned procedure.

- Establish an MP-EBGP peer relationship between the RRs. The configuration procedure is similar to the procedure for establishing an MP-EBGP peer relationship between two PE devices, except that you need to run the **peer ipv4-address next-hop-invariable** command in the BGP-VPNv4 address family view of the RRs to configure them not to change the next hop when advertising routes to the EBGP peers.
- Configure PE devices as the clients of the RR in the local AS to exchange VPNv4 routing information with the RR. For details about the configurations, see [8.7.11 Configuring Route Reflection to Optimize the VPN Backbone Layer](#).

8.7.5.4 Verifying the Inter-AS VPN Option C Configuration (Solution 1)

Prerequisites

The configuration of inter-AS VPN Option C (Solution 1) is complete.

Procedure

- Run the **display bgp vpnv4 all peer** command to check the BGP peers on the PE device. You can find the status of the EBGP peer between PEs is "Established".
- Run the **display bgp vpnv4 all routing-table** command to check the VPN IPv4 routing table on the PE or ASBR. You can view that the PE has the VPN IPv4 routes while the ASBR has no VPN IPv4 route.
- Run the **display bgp routing-table label** command to check information about the label of the IPv4 route on the ASBR.
- Run the **display ip routing-table vpn-instance vpn-instance-name** command to check the VPN routing table on the PE device. The command displays all VPN routes to all the CE devices in the VPN routing table of the PE device.

---End

8.7.6 Configuring Inter-AS VPN Option C (Solution 2)

If virtual private network (VPN) routes need to be established over a Multiprotocol Label Switching (MPLS) backbone network spanning multiple autonomous areas (ASs), inter-AS VPN is required. If each AS needs to exchange a large number of VPN routes, inter-AS VPN-Option C is a good choice to prevent the autonomous area border routers (ASBRs) from becoming bottlenecks that impede network expansion.

Pre-configuration Tasks

Before configuring inter-AS VPN Option C, complete the following tasks:

- Configuring an Interior Gateway Protocol (IGP) for the MPLS backbone network of each AS to ensure IP connectivity on the backbone network within each AS
- Configuring the basic MPLS functions and MPLS Label Distribution Protocol (LDP) or Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for the MPLS backbone network of each AS
- In each AS, configuring VPN instances on the PE devices connected to CE devices and associating the VPN instances with PE interfaces connected to CE devices

- Configuring route exchange between the PE and CE devices in each AS

For details about the configurations, see [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#).

Context

The following solutions can be used to implement inter-AS VPN-Option C:

- Solution 1: After learning the labeled BGP routes of the public network in the remote AS from the remote ASBR, the local ASBR allocates labels for these routes, and advertises these routes to the IBGP peer that supports the label switching capability. In this manner, a complete LSP is set up.
- Solution 2: The IBGP peer relationship between the PE and ASBR is not needed. In this solution, an ASBR learns the labeled public BGP routes of the remote AS from the peer ASBR. Then these labeled public BGP routes are imported to an IGP to trigger the establishment of an LDP LSP. In this manner, a complete LDP LSP can be established between the two PEs.

If an ASBR is ready to access a large number of PEs, solution 2 is recommended for its easy configuration.

NOTE

In inter-AS VPN Option C mode, do not enable LDP between ASBRs. If LDP is enabled on the interfaces between ASBRs, LDP sessions are then established between the ASBRs. When a lot of BGP routes exist, many LDP labels are occupied.

Configuration Procedure

All the following tasks are mandatory. Perform these tasks in this sequence to complete inter-AS VPN Option C configuration.

When VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources, you also need to complete the task of [8.7.15 Configuring Tunnel Policies](#).

8.7.6.1 Establishing the EBGP Peer Relationship Between ASBRs

Context

An EBGP peer relationship is established between ASBRs to advertise routes destined for the loopback interfaces on PEs.

Perform the following steps on ASBRs.

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run `interface interface-type interface-number`

The view of the interface that connects the remote ASBR is displayed.

Step 3 Run **ip address** *ip-address* { *mask* | *mask-length* }

The IP address is configured.

Step 4 Run **quit**

Return to the system view.

Step 5 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 6 Run **peer** *ipv4-address* **as-number** *as-number*

The remote ASBR is configured as the EBGP peer.

Step 7 (Optional) Run **peer** { *ipv4-address* | *group-name* } **ebgp-max-hop** [*hop-count*]

The maximum number of hops is configured for the EBGP connection.

Generally, one or multiple directly connected physical links exist between EBGP peers. If the directly connected physical link(s) are not available, run the **peer ebgp-max-hop** command to ensure that the TCP connection can be set up between the EBGP peers through multiple hops.

----End

8.7.6.2 Advertising the Routes of the PE in the Local AS to the Remote PE

Context

After the routes of the loopback interface on a PE in an AS are advertised to the remote PE in another AS, the MP-EBGP peer relationship is established between PEs.

Procedure

- The loopback address of the PE in the local AS is advertised to the remote ASBR.
Perform the following steps on the local ASBR:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
 - c. Run **network** *ip-address* [*mask* | *mask-length*]
The loopback address of the PE in the local AS is advertised to the remote ASBR.
- The BGP routes are imported to IGP.
Perform the following steps on the peer ASBR:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **ospf** *process-id*
The OSPF view is displayed.
 - c. Run **import-route bgp** [*cost cost*] [**route-policy** *route-policy-name*]

The BGP routes are imported to IGP.

----End

8.7.6.3 Enabling the Capability of Exchanging Labeled IPv4 Routes

Context

To establish an inter-AS BGP LSP, you must enable ASBRs to exchange labeled IPv4 routes.

Perform the following steps on ASBRs.

Procedure

- Creating a routing policy.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **route-policy route-policy-name permit node node**
The routing policy applied to advertise routes to the remote ASBR is configured.
 - c. Run **apply mpls-label**
Labels for IPv4 routes are distributed.
 - d. Run **quit**
Return to the system view.
- Applying a Routing Policy
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp { as-number-plain | as-number-dot }**
The BGP view is displayed.
 - c. Run **peer ipv4-address route-policy route-policy-name export**
The routing policy applied to advertise routes to the remote ASBR is configured.
 - d. Run **quit**
Return to the system view.
- Enabling the function of labeled IPv4 route exchange.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **interface interface-type interface-number**
The view of the interface connecting the remote ASBR is displayed.
 - c. Run **mpls**
The MPLS function is enabled.
 - d. Run **quit**
Return to the system view.

- e. Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

----End

8.7.6.4 Establishing an LDP LSP for the Labeled BGP Routes of the Public Network

Context

By enabling LDP on ASBRs to allocate labels for BGP routes, you can establish LDP LSPs for labeled BGP routes of the public network that are filtered in the IP prefix list.

Perform the following steps on ASBRs.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mpls**

The MPLS view is displayed.

Step 3 Run **lsp-trigger bgp-label-route** [**ip-prefix** *ip-prefix-name*]

An LDP LSP is established for the labeled BGP routes of the public network that is filtered by the IP prefix list.

----End

8.7.6.5 Establishing the MP-EBGP Peer Relationship Between PEs

Prerequisites

By introducing extended community attributes into BGP, MP-IBGP can advertise VPNv4 routes between PEs. PEs of different ASs are generally not directly connected. To set up an EBGP connection between the PEs of different ASs, you must configure the permitted maximum number of hops between PEs.

Perform the following steps on PEs.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer ipv4-address as-number as-number**

The remote PE is specified as the EBGP peer.

Step 4 Run **peer *ipv4-address* connect-interface *interface-type* *interface-number* *ipv4-source-address***

The source interface that sends BGP packets is specified.

Step 5 Run **peer *ipv4-address* ebgp-max-hop [*hop-count*]**

The maximum number of hops permitted to establish the EBGP peer is specified.

Step 6 (Optional) Run **peer { *group-name* | *ipv4-address* } mpls-local-ifnet disable**

The ability to establish an MPLS local IFNET tunnel between PEs is disabled.

In the Option C scenario, PEs establish an MP-EBGP peer relationship. Therefore, an MPLS local IFNET tunnel between PEs is established over the MP-EBGP peer relationship. The MPLS local IFNET tunnel fails to transmit traffic because PEs are indirectly connected.

If a fault occurs on the BGP LSP between PEs, traffic is iterated to the MPLS local IFNET tunnel, not an FRR bypass tunnel. As the MPLS local IFNET tunnel cannot forward traffic, traffic is interrupted. To prevent the traffic interruption, run this command to disable the establishment of an MPLS local IFNET tunnel between PEs.

Step 7 Run **ipv4-family vpnv4**

The BGP VPNv4 sub-address family view is displayed.

Step 8 Run **peer *ipv4-address* enable**

The VPNv4 route exchange capability with the remote PE is enabled.

----End

8.7.6.6 Verifying the Inter-AS VPN Option C Configuration (Solution 2)

Prerequisites

The configurations of the Inter-AS VPN Option C (Solution 2) function are complete.

Procedure

- Run the **display bgp vpnv4 all peer** command to check information about the specified VPNv4 peer on a PE. You can find that the EBGP peer relationship between PEs is established.
- Run the **display bgp vpnv4 all routing-table** command to check information about the VPN-IPv4 routing table on a PE or an ASBR. You can find that BGP VPNv4 routes and BGP VPN instance routes are on the PE, rather than on the ASBR.
- Run the **display bgp routing-table label** command to check information about the labels of IPv4 routes on an ASBR.
- Run the **display ip routing-table vpn-instance *vpn-instance-name*** command to check the VPN routing table on a PE device. You can find that the VPN routing table of the PE has the VPN routes to the CE related to the specified VPN instance.
- Run the **display mpls route-state [{ *exclude* | *include* } { *idle* | *ready* | *settingup* } * | *destination-address mask-length*] [*verbose*]** command to check the matching relationship between routes and the LSP on an ASBR. You can find the routes with the type as **L**, that is, the labeled BGP routes of the public network.

- Run the **display ip routing-table** command to check information about the routing table on an ASBR. You can find that the routes to the remote PE are labeled BGP routes of the public network: The routing table is "Public", the protocol type is "BGP", and the label has a non-zero value.
- Run the **display mpls lsp [vpn-instance vpn-instance-name] [protocol ldp] [{ exclude | include } ip-address mask-length] [outgoing-interface interface-type interface-number] [in-label in-label-value] [out-label out-label-value] [lsr-role { egress | ingress | transit }] [verbose]** command to check whether an LDP LSP is established on an ASBR. You can find that an LDP LSP is established between the ASBR and the remote PE. Besides, the LDP ingress LSP to the remote PE can be found on the local PE.

---End

8.7.7 Configuring an MCE Device

A multi-VPN-instance CE (MCE) device can connect to multiple VPNs. The MCE solution isolates services of different VPNs while reducing cost of network devices.

Pre-configuration Tasks

Before configuring an MCE device, complete the following tasks:

- **Configuring a VPN Instance** on the multi-instance CE, and the PE that is accessed by it (each service with a VPN instance)
- Configuring the link layer protocol and network layer protocol for LAN interfaces and connecting the LAN to the multi-instance CE (each service using an interface to access the multi-instance CE)
- Binding related VPN instances to the interfaces of the multi-instance CE and PE interfaces through which the PE accesses the multi-instance and configuring IP addresses for those interfaces

Configuration Procedure

The following tasks are mandatory and can be performed in a random order.

8.7.7.1 Configure Route Exchange Between an MCE Device and VPN Sites

Context

Routing protocols that can be used between an MCE device and VPN sites are static routing, RIP (Routing Information Protocol), OSPF (Open Shortest Path First), IS-IS (Intermediate System to Intermediate System), or BGP (Border Gateway Protocol). Choose one of the following configurations as needed:

- **Configure static routes between an MCE device and a site.**
- **Configure RIP between an MCE device and a site.**
- **Configure OSPF between an MCE device and a site.**
- **Configure IS-IS between an MCE device and a site.**
- **Configure BGP between an MCE device and a site.**

The following configurations are performed on the MCE device. On the devices in the site, you only need to configure the corresponding routing protocol.

Configure Static Routes Between an MCE Device and a Site

Perform the following configurations on the MCE device. You only need to configure a static route to the MCE device in the site. The site configuration is not provided here. For detailed configuration of static routes, see Static Route Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 8-13 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Configure a static route to the site.	ip route-static vpn-instance <i>vpn-source-name destination-address { mask mask-length } { nexthop-address [public] interface-type interface-number [nexthop-address] } [preference preference tag tag] *</i>	You must specify the next hop address on the MCE device.

Configure RIP Between an MCE Device and a Site

Perform the following configurations on the MCE device. Configure RIPv1 or RIPv2 in the site. The site configuration is not provided here. For detailed RIP configuration, see RIP Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 8-14 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create a RIP process running between the MCE device and the site and enter the RIP view.	rip process-id vpn-instance <i>vpn-instance-name</i>	A RIP process can be bound to only one VPN instance. If a RIP process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.

Action	Command	Description
Enable RIP on the network segment of the interface to which the VPN instance is bound.	network <i>network-address</i>	-
(Optional) Import the routes to the remote sites advertised by the PE device in to the RIP routing table.	import-route { { static direct unr } { rip ospf isis } [<i>process-id</i>] } [cost <i>cost</i> route-policy <i>route-policy-name</i>] * import-route bgp [cost { <i>cost</i> transparent } route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.

Configure OSPF Between an MCE Device and a Site

Perform the following configurations on the MCE device. Configure OSPF in the site. The site configuration is not provided here. For detailed OSPF configuration, see OSPF Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR200-S&AR2200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 8-15 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an OSPF process running between the MCE device and the site and enter the OSPF view.	ospf [<i>process-id</i> router-id <i>router-id</i>] * vpn-instance <i>vpn-instance-name</i>	-
(Optional) Import the routes to the remote sites advertised by the PE device into the OSPF routing table.	import-route { bgp [permit-ibgp] direct unr rip [<i>process-id-rip</i>] static isis [<i>process-id-isis</i>] ospf [<i>process-id-ospf</i>] } [cost <i>cost</i> type <i>type</i> tag <i>tag</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.
Configure an OSPF area and enter the OSPF area view.	area { <i>area-id</i> <i>area-id-address</i> }	-

Action	Command	Description
Enable OSPF on the network segment of the interface to which the VPN instance is bound.	network <i>ip-address wildcard-mask</i>	-

Configure IS-IS Between an MCE Device and a Site

Perform the following configurations on the MCE device. You only need to configure IS-IS in the site. The site configuration is not provided here. For detailed IS-IS configuration, see IS-IS Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR200-S&AR2200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 8-16 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an IS-IS process running between the MCE device and the site and enter the IS-IS view.	isis process-id vpn-instance <i>vpn-instance-name</i>	An IS-IS process can be bound to only one VPN instance. If an IS-IS process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Set a network entity title (NET) for the IS-IS process.	network-entity <i>net</i>	A NET specifies the current IS-IS area address and the system ID of the router. A maximum of three NETs can be configured for one process on each router.

Action	Command	Description
Import the routes to the remote sites advertised by the PE device into the IS-IS routing table.	Use either of the following commands: <ul style="list-style-type: none"> ● import-route { direct static unr { ospf rip isis } [<i>process-id</i>] bgp } [cost-type { external internal } cost <i>cost</i> tag <i>tag</i> route-policy <i>route-policy-name</i> [level-1 level-2 level-1-2]] * ● import-route { { ospf rip isis } [<i>process-id</i>] bgp direct unr } inherit-cost [{ level-1 level-2 level-1-2 } tag <i>tag</i> route-policy <i>route-policy-name</i>] * 	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.
Return to system view.	quit	-
Enter the view of the interface to which the VPN instance is bound.	interface <i>interface-type interface-number</i>	-
Enable IS-IS on the interface.	isis enable [<i>process-id</i>]	-

Configure BGP between an MCE Device and a Site

Perform the following configurations on the MCE device.

Table 8-17 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
Configure the device connected to the MCE device in the site as a VPN peer.	peer <i>ipv4-address as-number as-number</i>	-

Action	Command	Description
Import the routes to the remote sites advertised by the PE device into the BGP routing table.	import-route <i>protocol</i> [<i>process-id</i>] [med <i>med</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.

Perform the following configurations on the device connected to the MCE device in the site.

Table 8-18 Site configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Configure the MCE device as a VPN peer.	peer <i>ipv4-address</i> as-number <i>as-number</i>	-
Import IGP routes of the VPN into the BGP routing table.	import-route <i>protocol</i> [<i>process-id</i>] [med <i>med</i> route-policy <i>route-policy-name</i>] *	The site must advertise routes to its attached VPN network segments to the MCE device.

8.7.7.2 Configure Route Exchange Between an MCE Device and a PE Device

Context

Routing protocols that can be used between an MCE device and a PE device are static routing, RIP, OSPF, IS-IS, and BGP. Choose one of the following configurations as needed:

- **Configure static routes between an MCE device and a PE device.**
- **Configure RIP between an MCE device and a PE device.**
- **Configure OSPF between an MCE device and a PE device.**
- **Configure IS-IS between an MCE device and a PE device.**
- **Configure BGP between an MCE device and a PE device.**

The following configurations are performed on the MCE device. The configurations on the PE device are similar to those on a PE device in the BGP/MPLS IP VPN networking. For detailed configuration, see [Configuring Route Exchange Between PE and CE Devices](#).

Configure Static Routes Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 8-19 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Configure a static route to the PE device.	ip route-static vpn-instance vpn-source-name destination-address { mask mask-length } vpn-instance vpn-destination-name nexthop-address [preference preference tag tag] *	You must specify the next hop address on the MCE device.

Configure RIP Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 8-20 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create a RIP process running between the MCE and PE devices and enter the RIP view.	rip process-id vpn-instance vpn-instance-name	A RIP process can be bound to only one VPN instance. If a RIP process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Enable RIP on the network segment of the interface to which the VPN instance is bound.	network network-address	-
(Optional) Import VPN routes of the site into the RIP routing table.	import-route { { static direct unr } { rip ospf isis } [process-id] } [cost cost route-policy route-policy-name] * import-route bgp [cost { cost transparent } route-policy route-policy-name] *	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.

Configure OSPF Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 8-21 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an OSPF process running between the MCE and PE devices and enter the OSPF view.	ospf [<i>process-id</i> router-id <i>router-id</i>] * vpn-instance <i>vpn-instance-name</i>	-
(Optional) Import VPN routes of the site into the OSPF routing table.	import-route { bgp [permit-ibgp] direct unr rip [<i>process-id-rip</i>] static isis [<i>process-id-isis</i>] ospf [<i>process-id-ospf</i>] } [cost <i>cost</i> type <i>type</i> tag <i>tag</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.
Disable routing loop detection in the OSPF process.	vpn-instance-capability simple	By default, routing loop detection is disabled in an OSPF process. If routing loop detection is not disabled in the OSPF process on the MCE device, the MCE device rejects OSPF routes sent from the PE device.
Configure an OSPF area and enter the OSPF area view.	area { <i>area-id</i> <i>area-id-address</i> }	-
Enable OSPF on the network segment of the interface to which the VPN instance is bound.	network <i>ip-address wildcard-mask</i>	-

Configure IS-IS Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 8-22 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an IS-IS process running between the MCE and PE devices and enter the IS-IS view.	isis process-id vpn-instance vpn-instance-name	An IS-IS process can be bound to only one VPN instance. If an IS-IS process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Set a network entity title (NET) for the IS-IS process.	network-entity net	A NET specifies the current IS-IS area address and the system ID of the router. A maximum of three NETs can be configured for one process on each router.
(Optional) Import VPN routes of the site into the IS-IS routing table.	Use either of the following commands: <ul style="list-style-type: none"> ● import-route { direct static unr { ospf rip isis } [process-id] bgp } [cost-type { external internal } cost cost tag tag route-policy route-policy-name [level-1 level-2 level-1-2]] * ● import-route { { ospf rip isis } [process-id] bgp direct unr } inherit-cost [{ level-1 level-2 level-1-2 } tag tag route-policy route-policy-name] * 	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.
Return to system view.	quit	-
Enter the view of the interface to which the VPN instance is bound.	interface interface-type interface-number	-
Enable IS-IS on the interface.	isis enable [process-id]	-

Configure BGP Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 8-23 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Enter the BGP view.	bgp { <i>as-number-plain</i> <i>as-number-dot</i> }	-
Enter the BGP-VPN instance IPv4 address family view.	ipv4-family vpn-instance <i>vpn-instance-name</i>	-
Configure the PE device as the VPN peer of the MCE device.	peer <i>ipv4-address</i> as-number <i>as-number</i>	-
Import the routes to the remote sites advertised by the PE device into the BGP routing table.	import-route <i>protocol</i> [<i>process-id</i>] [med <i>med</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.

8.7.7.3 Verifying the MCE Configuration

Prerequisites

The configurations of the Multi-VPN-Instance CE function are complete.

Procedure

- Run the **display ip routing-table vpn-instance** *vpn-instance-name* [**verbose**] command to check the VPN routing table on the multi-instance CE. If there are routes to the LAN and the remote nodes for each service, the configuration is successful.

----End

8.7.8 Configuring HoVPN

The HoVPN networking reduces the requirements for PE devices.

Pre-configuration Tasks

Before configuring HoVPN, complete the task of [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#).

Configuration Procedure

In addition to basic BGP/MPLS IP VPN configuration, you need to specify UPE devices on the SPE device and advertise default routes of VPN instances to the UPE devices.

When VPN services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources, you also need to complete the task of [8.7.15 Configuring Tunnel Policies](#).

NOTE

According to RFC 4382, the VPN instance status obtained from a management information base (MIB) or schema is Up only if at least one interface bound to the VPN instance is Up. On an HoVPN, VPN instances on SPEs are not bound to interfaces. As a result, the VPN instance status obtained from a MIB or schema is always Down. To solve this problem, run the **transit-vpn** command in the VPN instance view or VPN instance IPv4 address family view of an SPE. Then, the VPN instance status obtained from a MIB or schema is always Up, no matter whether the VPN instance is bound to interfaces.

Perform the following steps on the SPE device.

Procedure

Step 1 Specify a UPE device.

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **peer** { *ipv4-address* | *group-name* } **as-number** *as-number*
A UPE device is specified as the BGP peer of the SPE.
4. Run **ipv4-family vpnv4** [**unicast**]
The BGP-VPNv4 family is displayed.
5. Run **peer** { *ipv4-address* | *group-name* } **enable**
The capability of exchanging BGP VPNv4 routing information with the peer is enabled.
6. Run **peer** { *ipv4-address* | *group-name* } **upe**
The peer is specified as the UPE of the SPE.

Step 2 Advertise default routes of a VPN instance.

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **ipv4-family vpnv4**
The BGP-VPNv4 family view is displayed.
4. Run **peer** { *ipv4-address* | *group-name* } **default-originate vpn-instance** *vpn-instance-name*
The default routes of a specified VPN instance are advertised to the UPE device.

After running the command, the SPE advertises a default route to the UPE with its local address as the next hop, regardless of whether there is a default route in the local routing table.

---End

Verifying the Configuration

After completing the HoVPN configuration, run the **display ip routing-table** command on the CE devices. You can see that the local CE device does not have any route to the network segment of the remote CE interface but has a default route with the next hop as the UPE device.

8.7.9 Configuring PBR to an LSP for VPN Packets

Policy-based routing (PBR) to an LSP enables the device to forward VPN packets through LSPs on the MPLS backbone network through PBR, without the need to search the forwarding table of the VPN instance.

Context

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S supports PBR to an LSP for VPN packets, which can be used for VPN data forwarding.

If VPN packets do not match the PBR rules, they are forwarded according to common VPN data forwarding process. If VPN packets match the PBR rules, they are forwarded through the specified LSP.

NOTE

PBR to an LSP for VPN packets requires two or more LSPs. If PBR to an LSP for VPN packets are used together with LDP FRR, the LSPs must work in active/standby mode. In other situations, the LSPs can work in active/standby mode or load balancing mode.

Perform the following configuration on the ingress PE device.

Pre-configuration Tasks

Before configuring PBR to an LSP for VPN packets, complete the following tasks:

- Configuring an ACL to filter packets if you want to filter packets based on IP addresses
- Configuring at least two LSPs from the ingress PE device to the egress PE device
- Configuring LDP FRR if necessary

Procedure

Step 1 Configure PBR to an LSP for VPN packets.

1. Run **system-view**

The system view is displayed.

2. Run **policy-based-route** *policy-name* { **deny** | **permit** } **node** *node-id*

A routing policy or a policy node is created.

3. Run **if-match acl** *acl-number***if-match packet-length** *min-length max-length*

An if-match clause is configured to match the IP addresses of packets.

Or, run:

An if-match clause is configured to match the lengths of IP packets.

4. Run **apply lsp vpn** *vpn-instance-name ce-address* [*pe-address* [*p-address* | *interface-type interface-number* | **secondary**]]

PBR to an LSP are configured for VPN packets.

5. (Optional) Run **ip policy-based-route refresh-time** [*refresh-time-value*]

The interval at which local PBR updates LSPs is configured.

By default, the interval at which local PBR updates LSPs is 5000 ms.

Step 2 Apply PBR.

Enable PBR in the system (local PBR).

1. Run **system-view**

The system view is displayed.

2. Run **ip local policy-based-route** *policy-name*

Local PBR is enabled.

Local PBR takes effect only to locally originated packets and only one local PBR rule can be configured.

---End

Verifying the Configuration

After completing the configuration of PBR to an LSP, run the **tracert lsp** [*-a source-ip* | *-exp exp-value* | *-h ttl-value* | *-r reply-mode* | *-t time-out*] * { **ip** *destination-address mask-length* [*ip-address*] [**nexthop** *nexthop-address* | **draft6**] | **te tunnel** *interface-number* [**hot-standby**] [**draft6**] } command to check the VPN packet transmission path. The command output shows that VPN packets are transmitted through the specified LSP.

NOTE

Before running the **tracert lsp** command on a CE device to check the packet forwarding path, run the **ttl propagate vpn** command on the ingress and egress PE devices directly connected to the CE device to enable MPLS IP TTL replication.

8.7.10 Configuring an OSPF Sham Link

The sham link between two PE devices on an MPLS VPN backbone network is considered as an OSPF intra-area route. Then VPN traffic is transmitted through the route over the backbone network but not backdoor routes.

Pre-configuration Tasks

Before configuring an OSPF sham link, complete the following tasks:

- **8.7.1 Configuring Basic BGP/MPLS IP VPN Functions** (use OSPF between PE and CE)

- Configuring OSPF in the LANs where the CE devices are located

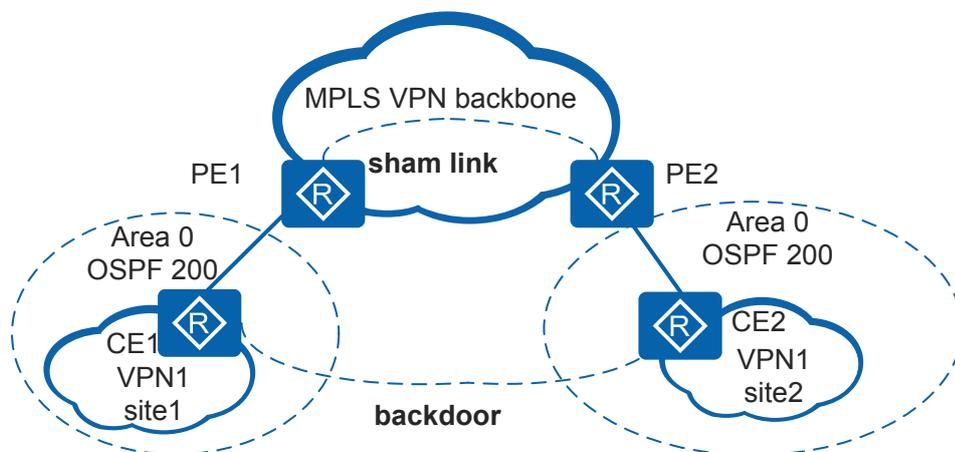
Context

OSPF sham links are IP unnumbered P2P links between two PE devices on an MPLS VPN backbone network.

Generally, BGP peers use BGP extended community attributes to carry routing information over the MPLS VPN backbone. OSPF running on a PE device can use the routing information to generate inter-area routes from the PE to CE devices.

As shown in **Figure 8-39**, if an intra-area OSPF link exists between the network segments of local and remote CE devices, this OSPF link is called a backdoor link.

Figure 8-39 OSPF sham link



The routes that pass through a backdoor link are intra-area routes and have a higher preference than the inter-area routes that pass through the MPLS VPN backbone network. As a result, VPN traffic is always forwarded through the backdoor routes instead of the backbone network. Generally, backdoor links are only used as backup links.

To avoid such a problem, an OSPF sham link can be established between the PE devices. In this way, the routes that pass through the MPLS VPN backbone network become OSPF intra-area routes and are preferred over the backdoor routes in VPN traffic forwarding.

Configure an OSPF sham link only when a backdoor link exists between two sites in the same OSPF area. If no backdoor link exists between sites in the same area, you do not need to configure any OSPF sham link.

Perform the following steps on the PE devices at both ends of a sham link.

Procedure

Step 1 Configure an endpoint address for the sham link.

Each VPN instance must have an endpoint address of the sham link. The endpoint address is a loopback interface address with a 32-bit mask in the VPN address space on a PE device. Multiple sham links of the same OSPF process share an endpoint address, but sham links of different OSPF processes cannot have the same endpoint address.

1. Run **system-view**
The system view is displayed.
2. Run **interface loopback interface-number**
A loopback interface is created and the loopback interface view is displayed.
3. Run **ip binding vpn-instance vpn-instance-name**
The loopback interface is bound to a VPN instance.
4. Run **ip address ip-address { mask | mask-length }**
An IP address is assigned to the loopback interface.

 **NOTE**

The loopback interface address must have a 32-bit mask, 255.255.255.255.

Step 2 Advertise routes of the sham link endpoint address.

1. Run **system-view**
The system view is displayed.
2. Run **bgp { as-number-plain | as-number-dot }**
The BGP view is displayed.
3. Run **ipv4-family vpn-instance vpn-instance-name**
The BGP-VPN instance IPv4 address family view is displayed.
4. Run **import-route direct**
Direct routes are imported to BGP. (The route of the sham link endpoint address is imported to BGP).

BGP advertises the sham link endpoint address as a VPN IPv4 address.

 **NOTE**

The route of the sham link endpoint address cannot be advertised to the peer PE through an OSPF process bound to a VPN instance.

If the route of the sham link endpoint address is advertised to the peer PE through an OSPF process bound to a VPN instance, the peer PE has two routes to the sham link endpoint address. One route is learned from the OSPF process, and the other is learned from MP-BGP. The OSPF route takes precedence over the BGP route, so the peer PE uses the OSPF route. As a result, the sham link fails to be established.

Step 3 Create a sham link.

1. Run **system-view**
The system view is displayed.
2. Run **ospf process-id [router-id router-id] vpn-instance vpn-instance-name**
The OSPF view is displayed.
3. Run **area area-id**
The OSPF area view is displayed.
4. Run **sham-link source-ip-address destination-ip-address [[simple [plain plain-text | [cipher] cipher-text] | { md5 | hmac-md5 | hmac-sha256 } [key-id { plain plain-text | [cipher] cipher-text }] | authentication-null | keychain keychain-name] | smart-**

```
discover | cost cost | dead dead-interval | hello hello-interval | retransmit retransmit-interval | trans-delay trans-delay-interval ] *
```

A sham link is configured.

The default settings of the parameters in the command are as follows:

- *cost* (sham link interface cost): 1
- *dead-interval* (sham link timeout interval): 40 seconds
- *hello-interval* (interval for sending Hello packets on the sham link interface): 10 seconds
- *retransmit-interval* (LSA packet retransmission interval on the sham link interface): 5 seconds
- *trans-delay-interval* (delay in sending LSA packets on the sham link interface): 1 second

Both ends of the sham link must use the same packet authentication method. If packet authentication is configured, the PE devices accept only the OSPF packets that pass the authentication. If packets fail the authentication, the neighbor relationship cannot be established between the PE devices.

If simple-text authentication (**simple**) is used, the authentication key type is **plain** by default. If the MD5 or HMAC-MD5 authentication (**md5** | **hmac-md5**) is used, the authentication key type is **cipher** by default.

NOTE

If **plain** is selected, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text.

MD5 and HMAC-MD5 authentication cannot ensure security. Keychain authentication is recommended.

To forward VPN traffic over the MPLS backbone network, ensure that the cost of the sham link is smaller than the cost of the OSPF route used for forwarding VPN traffic over the customer network. A commonly used method is to set the cost of the forwarding interface on the customer network to be larger than the cost of the sham link.

---End

Verifying the Configuration

After configuring an OSPF sham link, you can check the routing table on a CE, trace the nodes that data packets pass through from local CE to the remote CE, and check whether the sham link is successfully established on the PE.

- Run the **display ip routing-table vpn-instance** *vpn-instance-name* command on the PE to check the VPN routing table. You can see from the VPN routing table that the route from the PE to the remote CE is an OSPF route that passes through the customer network but not a BGP route that passes through the backbone network.
- Run the **display ip routing-table** and **tracert** *host* commands on a CE, and you can find that the VPN traffic from the local CE to the remote CE is forwarded through the backbone network.
- Run the **display ospf process-id sham-link** [**area** *area-id*] command on the PE to check whether the sham link is established successfully. You can find that the OSPF neighbor relationship between the PE and the remote CE is Full.
- Run the **display ospf routing** on the CE, and you can find that the route to the remote CE is an intra-area route.

8.7.11 Configuring Route Reflection to Optimize the VPN Backbone Layer

Using an RR can reduce the number of MP IBGP connections between PEs. This not only reduces the burden of PEs, but also facilitates network maintenance and management.

Pre-configuration Tasks

Before configuring route reflection to optimize the VPN backbone layer, complete the following tasks:

- Configuring the routing protocol for the MPLS backbone network to implement IP interworking between devices on the backbone network
- Establishing tunnels (LSPs, GRE, or MPLS TE tunnels) between the RR and all client PE devices

Configuration Procedure

All the following configuration tasks are mandatory. An RR can be any device such as P, PE, and ASBR.

8.7.11.1 Configuring the Client PEs to Establish MP IBGP Connections with the RR

Context

Perform the following steps on all Client PEs.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** *ipv4-address* **as-number** *as-number*

The RR is specified as the BGP peer.

Step 4 Run **peer** *ipv4-address* **connect-interface** *interface-type* *interface-number*

The interface is specified as an interface to establish the TCP connection.

Step 5 Run **ipv4-family vpnv4**

The BGP VPNv4 address family view is displayed.

Step 6 Run **peer** *ipv4-address* **enable**

The capability of exchanging VPNv4 routes between the PE and RR is enabled.

----End

8.7.11.2 Configuring the RR to Establish MP IBGP Connections with the Client PEs

Context

Choose one of the following schemes to configure the RR.

Procedure

- Configuring the RR to establish MP IBGP connections with the peer group
Add all the client PEs to the peer group and establish MP-IBGP connections with the peer group.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
 - c. Run **group** *group-name* [**internal**]
An IBGP peer group is created.
 - d. Run **peer** *group-name* **connect-interface** *interface-type interface-number*
The interface is specified as an interface to establish the TCP connection.
 - e. Run **ipv4-family vpnv4**
The BGP VPNv4 address family view is displayed.
 - f. Run **peer** *group-name* **enable**
The capability of exchanging IPv4 VPN routes between the RR and the peer group is enabled.

By default, only the peer in the BGP IPv4 unicast address family view is automatically enabled.
 - g. Run **peer** *ip-address* **group** *group-name*
The peer is added to the peer group.
- Configuring the RR to establish an MP IBGP connection with each client PE
Repeat the following steps on the RR to establish an MP IBGP connection with each client PE.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
 - c. Run **peer** *ipv4-address* **as-number** *as-number*
The client PE is specified as the BGP peer.
 - d. Run **peer** *ipv4-address* **connect-interface** *interface-type interface-number*
The interface is specified as an interface to establish the TCP connection.

e. Run **ipv4-family vpnv4**

The BGP VPNv4 address family view is displayed.

f. Run **peer ipv4-address enable**

The capability of exchanging VPNv4 routes between the RR and the client PE is enabled.

----End

8.7.11.3 Configuring Route Reflection for BGP IPv4 VPN Routes

Context

Perform the following steps on the RR.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp { as-number-plain | as-number-dot }**

The BGP view is displayed.

Step 3 Run **ipv4-family vpnv4**

The BGP VPNv4 address family view is displayed.

Step 4 Enable route reflection for BGP VPNv4 routes on the RR.

- Run the **peer group-name reflect-client** command to enable route reflection if the RR establishes the MP IBGP connection with the peer group consisting of client PEs.
- Run the **peer ipv4-address reflect-client** command repeatedly to enable route reflection if the RR establishes the MP IBGP connection with each PE rather than peer group.

Step 5 Run **undo policy vpn-target**

The filtering of VPNv4 routes based on the VPN target is disabled.

Step 6 (Optional) Run **rr-filter { extcomm-filter-number | extcomm-filter-name }**

The reflection policy is configured for the RR. Only the IBGP route of which route-target extended community attribute meets the matching rules can be reflected. This allows load balancing among RRs.

In the command, the extended community filter specified by *extcomm-filter-number* or *extcomm-filter-name* must have been configured using the **ip extcommunity-filter** command.

Step 7 (Optional) Run **undo reflect between-clients**

Route reflection is disabled between clients.

If the clients of an RR have established full-mesh connections with each other, the **undo reflect between-clients** command can be used to disable route reflection between clients in order to reduce the link cost. By default, route reflection is enabled between the clients of an RR.

This command can only be configured on the RR.

Step 8 (Optional) Run **reflector cluster-id** *cluster-id*

The RR cluster ID is set.

If a cluster has multiple RRs, you can use this command to set the same cluster ID for these RRs to prevent routing loops. By default, the cluster ID is the router ID.

---End

8.7.11.4 Verifying the Configuration of Route Reflection to Optimize the VPN Backbone Layer

Prerequisites

The configurations of the reflection to optimize the VPN backbone layer function are complete.

Procedure

- Run the **display bgp vpnv4 all peer** [[*ipv4-address*] **verbose**] command to check information about the BGP VPNv4 peer on the RR or the Client PEs. You can find that the status of the MP IBGP connections between the RR and all Client PEs is "Established".
- Run the **display bgp vpnv4 all routing-table peer** *ipv4-address* { **advertised-routes** | **received-routes** } command or **display bgp vpnv4 all routing-table statistics** command to check information about the routes received from the peer or the routes advertised to the peer on the RR or the Client PEs. You can find that the RR and each Client PE can receive and send VPNv4 routing information between each other.
- Run the **display bgp vpnv4 all group** [*group-name*] command to check information about the VPNv4 peer group on the RR. You can view information about the group members and find that the status of the BGP connections between the RR and the group members is "Established".

---End

8.7.12 Configuring IP FRR for VPN Routes

When multiple CE devices in a VPN site connect to the same PE, you can configure IP FRR for VPN routes. IP FRR enables VPN traffic to be fast switched to another PE-CE link when the next hop of the primary route is unreachable.

Pre-configuration Tasks

Before configuring IP FRR for VPN routes, complete the following tasks:

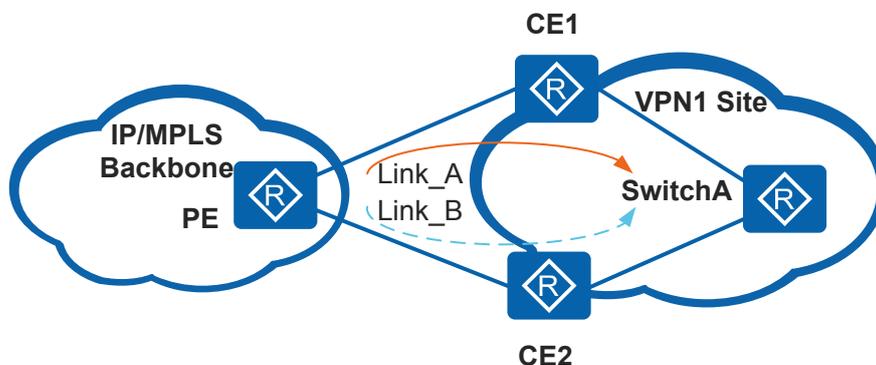
- [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#)
- Ensuring that the PE has learned VPN routes with the same prefix from the attached CE devices

Context

IP FRR for VPN routes is used in scenarios where multiple CE devices connect to one PE device. As shown in [Figure 8-40](#), the PE device forwards data to the site of vpn1 through

Link_A, and Link_B is a backup link. When the PE device detects that the route to CE1 is unreachable, it immediately switches traffic to Link_B and then performs other operations to trigger VPN route convergence. This minimizes impact of the link failure on VPN services.

Figure 8-40 IP FRR for VPN routes



Configuration Procedure

The router supports IP FRR for VPN routes.

Perform the following steps on a PE device.

Procedure

- configuring IP FRR
 - a. Run **system-view**

The system view is displayed.
 - b. Run **route-policy route-policy-name { permit | deny } node node**

A node is configured for a route-policy, and the route-policy view is displayed.
 - c. Run **apply backup-interface interface-type interface-number**

A backup outbound interface is specified.
 - d. (Optional) Run **apply backup-nexthop ip-address**

A backup next hop is specified.

The backup next hop is optional for a P2P link and mandatory for a non-P2P link.
 - e. Run **quit**

Return to the system view.
 - f. Run **ip vpn-instance vpn-instance-name**

The VPN instance view is displayed.
 - g. Run **ipv4-family**

The VPN instance IPv4 address family view is displayed.

- h. Run **ip frr route-policy** *route-policy-name*

IP FRR is enabled for the VPN instance IPv4 address family.

----End

Verifying the Configuration

Run the **display ip routing-table vpn-instance** *vpn-instance-name* [*ipv4-address*] **verbose** command to check the backup next hops and backup outbound interfaces of VPN-IPv4 routes in the routing table.

8.7.13 Configuring VPN FRR

In the networking of CE dual-homing, you can configure VPN FRR to ensure VPN service switchover to a secondary link when the primary link between PEs fails.

Pre-configuration Tasks

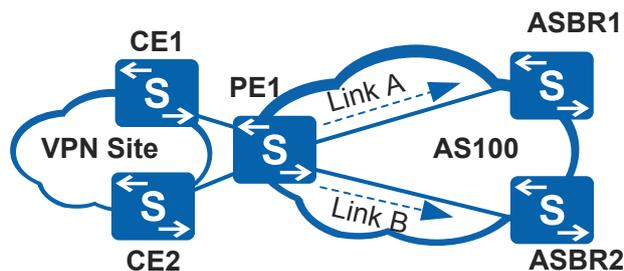
Before configuring VPN FRR, complete the following tasks:

- [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#)
- Generating two unequal-cost routes on the PE by setting different costs or metrics

Context

VPN FRR is used in PE multi-homing scenarios to enhance network reliability. As shown in [Figure 8-41](#), if the primary link (Link A) between PE1 and ASBR1 fails, VPN FRR quickly switches traffic to the backup link (Link B) between PE1 and ASBR2 to minimize the impact of the link failure on VPN services.

Figure 8-41 VPN FRR networking



You can configure VPN FRR in either of the following modes:

- Manual VPN FRR: Information such as the backup next hop is specified.
- Auto VPN FRR: The backup next hop is unspecified, but a proper next hop is selected for the VPN route.

You can select either mode as required. If both of them are configured, manual VPN FRR has a higher priority. When manual VPN FRR fails, auto VPN FRR takes effect.

 NOTE

- Configuring the **lsp-trigger** command on the P is not recommended when an LSP is created on the VPN backbone network. Use the default configuration on the P. Otherwise, VPN FRR switchback may fail.
- To implement fast switching within milliseconds, configure BFD for LSPs. For details about BFD, see Configuring Static BFD to Detect an LDP LSP, Configuring Dynamic BFD for LDP LSPs and Configuring Static BFD for TE Tunnels in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS*. Perform the BFD configuration based on the tunnel used for forwarding VPN services.
- In the L3VPN over GRE scenario, the device does not support VPN FRR function.

Perform the following steps on a PE device.

Procedure

- Configure manual VPN FRR.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **route-policy route-policy-name { permit | deny } node node**
The routing policy node is created and the routing policy view is displayed.
 - c. Run **apply backup-nexthop ip-address**
The backup next hop is configured.
 - d. Run **ip vpn-instance vpn-instance-name ipv4-family vpn frr route-policy route-policy-name**
quit
Return to the system view.
 - e. Run
The VPN instance view is displayed.
 - f. Run
The VPN instance IPv4 address family view is displayed.
 - g. Run
The VPN FRR is enabled.
- Enable VPN auto FRR using a routing policy.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **route-policy route-policy-name { permit | deny } node node**
The routing policy node is created and the routing policy view is displayed.
 - c. Run **apply backup-nexthop auto**
The auto mode is used.
 - d. Run **ip vpn-instance vpn-instance-name ipv4-family vpn frr route-policy route-policy-name**
quit

- Return to the system view.
- e. Run
The VPN instance view is displayed.
- f. Run
The VPN instance IPv4 address family view is displayed.
- g. Run
The VPN FRR is enabled.
- (Optional) Add multiple VPNv4 routes to the VPN instance with a different RD from these routes' RDs.

By default, if the RD of the VPN instance on the local PE is different from the RDs of the VPN instances on multiple remote PEs, and the RDs of the VPN instances on remote PEs are the same, the local PE adds only the optimal route to the VPN instance after receiving VPNv4 or VPNv6 routes with the same destination address from the remote PEs. As a result, load balancing or VPN FRR does not take effect. To resolve this problem, run the **vpn-route cross multipath** command on the local PE.

- a. Run **system-view**
The system view is displayed.
- b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
- c. Run **ipv4-family vpn-instance** *vpn-instance-name*
The BGP-VPN instance IPv4 address family view is displayed.
- d. Run **vpn-route cross multipath**
Multiple VPNv4 routes are added to the VPN instance with a different RD from these routes' RDs.
- (Optional) Disable VPN FRR in all VPN instances.

To disable VPN FRR in a VPN instance, run the **undo vpn frr** command in the VPN instance view. However, if multiple VPN instances are configured on a PE and VPN FRR is enabled for each VPN instance, it is complex to disable VPN FRR one by one in these VPN instances.

To address this problem, the device allows you to disable VPN FRR in all VPN instances using one command.

- a. Run **system-view**
The system view is displayed.
- b. Run **undo vpn frr all**
VPN FRR is disabled from all VPN instances.

----End

Verifying the Configuration

All VPN FRR configurations are complete, run the **display ip routing-table vpn-instance vpn-instance-name [ip-address] verbose** command to check information about the backup next-hop PE, backup tunnel, and backup label.

8.7.14 Configuring VPN GR

In the process of active/standby control board switchover or the system upgrade, you can configure VPN GR to ensure that VPN traffic is not interrupted on the PE, CE, or P device.

Context

In GR process, two roles are defined according to their functions, that is, GR restarter and GR helper.

- GR restarter: performs active/standby control board switchover or the system upgrade.
- GR helper: helps the GR restarter to implement uninterrupted service forwarding.

NOTE

The AR3260-S can function as both the GR restarter and GR helper, and other devices can only function as the GR helper.

VPN GR is the collection of GR capabilities of various protocols running on devices on VPN networks. You need to configure IGP GR, BGP GR, MPLS LDP GR, or MPLS TE GR based on the related protocol running on the GR restarter. You also need to configure neighboring devices of the GR restarter as the GR helper to help the GR restarter implement uninterrupted service forwarding.

Configure specified VPN GR on the PE, CE, and P as follows:

- Configure IGP GR, BGP GR and MPLS LDP (or MPLS TE) GR on the PE device.
- Configure IGP GR and the MPLS LDP (or MPLS TE) GR on the P device.
- Configure IGP GR or the BGP GR on the CE device.
- If a VPN spans multiple ASs, you must configure the IGP GR, BGP GR and MPLS LDP GR on the ASBR.

NOTE

The GR capability cannot ensure uninterrupted traffic forwarding when the neighboring device performs an active/standby switchover at the same time.

Pre-configuration Tasks

Before configuring VPN GR, complete the following tasks:

- [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#)
- Enabling GR Helper on all the devices on the network

Procedure

- Configure IS-IS GR or OSPF GR.
 - For details about how to configure IS-IS GR, see section "Enabling IS-IS GR" in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers *Configuration Guide - IP Routing*.

- Configure MPLS LDP GR or MPLS TE GR.
 - For details about how to configure MPLS LDP GR, see section "" in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers *Configuration Guide - MPLS*.

---End

8.7.15 Configuring Tunnel Policies

This section describes how to configure a tunnel policy and tunnel selector. By default, VPN services are transmitted through LSP tunnels. To use TE tunnels to transmit VPN services or load balance VPN traffic on multiple tunnels, configure a tunnel policy.

Pre-configuration Tasks

Before configuring a tunnel policy, complete the following tasks:

- Creating GRE or LSP or MPLS TE tunnels used to transmit VPN services

NOTE

For details on how to create a GRE tunnel, see [GRE Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - VPN](#).

For details on how to create an LSP tunnel, see [MPLS LDP Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS](#).

For details on how to create a TE tunnel, see [MPLS TE Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS](#).

- Establishing the basic VPN network (For details about BGP/MPLS IP VPN configuration, see [Configuring Basic BGP/MPLS IP VPN Functions](#))

Before configuring and applying a tunnel selector, complete the following tasks:

- Configuring a tunnel policy (see [8.7.15.1 Configuring and Applying a Tunnel Policy](#))
- Configuring an RD filter if routes need to be filtered based on RDs
- Configuring an ACL or IPv4 prefix if routes need to be filtered based on the next hop IPv4 address

Configuration Procedure

When VPN services need to be transmitted over TE or GRE tunnels, or when multiple tunnels need to perform load balancing to fully use network resources, complete the task of [8.7.15.1 Configuring and Applying a Tunnel Policy](#).

To select TE or GRE tunnels to transmitted VPN services in HoVPN, inter-AS VPN Option B, or inter-AS VPN Option C networking, complete the task of [8.7.15.2 Configuring and Applying a Tunnel Selector](#) on the SPE, ASBR, and PE devices.

NOTE

By default, if you specify a nonexistent tunnel policy in a command, the command does not take effect.

If you need the nonexistent tunnel policy can be specified in a command, run the **tunnel-policy nonexistent-config-check** command.

8.7.15.1 Configuring and Applying a Tunnel Policy

Context

VPN data is transmitted over tunnels. By default, LSP tunnels are used to transmit data, and each service is transmitted by only one LSP tunnel.

If the default tunnel configuration cannot meet VPN service requirements, apply tunnel policies to VPNs. You can configure either of the following types of tunnel policies according to service requirements:

- Tunnel type prioritization policy: This policy can change the type of tunnels selected for VPN data transmission or select multiple tunnels for load balancing.
- Tunnel binding policy: This policy can bind multiple TE tunnels to provide QoS guarantee for a VPN.

Perform the following steps on the PE devices that need to use a tunnel policy.

Procedure

Step 1 Configure a tunnel policy.

Use either of the following methods to configure a tunnel policy.

Configure a tunnel type prioritization policy.

By default, no tunnel policy is configured. LSP tunnels are used to transmit VPN data and each VPN service is transmitted over one LSP tunnel.

1. Run **system-view**
The system view is displayed.
2. Run **tunnel-policy** *policy-name*
A tunnel policy is created, and tunnel policy view is displayed.
3. (Optional) Run **description** *description-information*
The description of the tunnel policy is configured.
4. Run **tunnel select-seq** { **gre** | **lsp** | **cr-lsp** } * **load-balance-number** *load-balance-number*
The sequence in which each type of tunnel is selected and the number of tunnels participating in load balancing are set.

Configure a tunnel binding policy.

1. Run **system-view**
The system view is displayed.
2. Run **interface tunnel** *interface-number*
A tunnel interface is created and the tunnel interface view is displayed.
3. Run **tunnel-protocol mpls te**
MPLS TE is configured as a tunnel protocol.
4. Run **mpls te reserved-for-binding**
The binding capability of the TE tunnel is enabled.

5. Run **mpls te commit**

The MPLS TE configuration is committed for the configuration to take effect.

6. Run **quit**

Return to the system view.

7. Run **tunnel-policy policy-name**

A tunnel policy is created.

8. (Optional) Run **description description-information**

The description of the tunnel policy is configured.

9. Run **tunnel binding destination dest-ip-address te { tunnel interface-number } &<1-16> [ignore-destination-check] [down-switch]**

Bind specified TE tunnels in the policy.

 **NOTE**

- If the PE device has multiple peers, you can run the **tunnel binding** command multiple times to specify different destination IP addresses in a tunnel policy.
- If **down-switch** is specified in the command, the system selects available tunnels in an order of LSP, CR-LSP when the bound tunnels are unavailable.

Step 2 Apply the tunnel policy.

1. Run **system-view**

The system view is displayed.

2. Run **ip vpn-instance vpn-instance-name**

The VPN instance view is displayed.

3. Run **ipv4-family**

The VPN instance IPv4 address family view is displayed.

4. Run **tnl-policy policy-name**

A tunnel policy is applied to the VPN instance IPv4 address family.

---End

Verifying the Configuration

After configuring a tunnel policy and apply it to a VPN instance, you can check information about the tunnel policy applied to the VPN instance and tunnels in the system.

- Run the **display tunnel-info { tunnel-id tunnel-id | all | statistics [slots] }** command to check information about tunnels in the system.
- Run the **display interface tunnel interface-number** command to check detailed information about a specified tunnel interface.
- Run the **display tunnel-policy [tunnel-policy-name]** command to check information about the specified tunnel policy.
- Run the **display ip vpn-instance verbose [vpn-instance-name]** command to check the tunnel policy applied to the specified VPN instance.

8.7.15.2 Configuring and Applying a Tunnel Selector

Context

By configuring a tunnel selector, you can set route filtering conditions to iterate expected routes to the specified tunnels. A tunnel consists of two parts:

- **if-match** clause: matches an attribute of routes, for example, RD and next hop.
If no **if-match** clause is configured in a tunnel selector, all routes match the tunnel selector.
- **apply** clause: applies a tunnel policy to the routes matching the filtering rules.

After a tunnel selector is applied to routes on a PE, ASBR, or SPE device, the device filters routes using the specified filtering rules and iterates the matching routes to specified tunnels.

A tunnel selector takes effect for the following routes:

- VPNv4 routes: When a tunnel selector is applied to a BGP-VPNv4 address family on an SPE device in HoVPN networking or an ASBR in inter-AS VPN Option B networking, the SPE device or ASBR applies the tunnel policy to VPNv4 routes and iterates the matching routes to expected tunnels.
- Labeled BGP-IPv4 routes: When a tunnel selector is applied to the BGP-IPv4 unicast address family on a PE device or an ASBR in inter-AS VPN Option C networking, the PE device or ASBR applies the tunnel policy to labeled BGP-IPv4 routes.

Procedure

Step 1 Create a tunnel selector.

1. Run **system-view**

The system view is displayed.

2. Run **tunnel-selector** *tunnel-selector-name* { **permit** | **deny** } **node** *node*

A tunnel selector is created, and tunnel selector view is displayed.

3. (Optional) Configure **if-match** clauses.

If no **if-match** clause is configured in a tunnel selector, all routes match the tunnel selector.

- To configure an **if-match** clause that filters routes based on router distinguishers (RDs), run **if-match rd-filter** *rd-filter-number*.
- To configure an **if-match** clause that filters routes based on next-hop IPv4 addresses, run **if-match ip next-hop** { **acl** { *acl-number* | *acl-name* } | **ip-prefix** *ip-prefix-name* }.
- To configure an **if-match** clause that filters routes based on next-hop IPv6 addresses, run **if-match ipv6 next-hop prefix-list** *ipv6-prefix-name*.

4. Run **apply tunnel-policy** *tunnel-policy-name*

An apply clause is configured to specify a tunnel policy for the routes matching the if-match clause.

Step 2 Apply the tunnel selector.

Perform the following steps on an SPE device in HoVPN networking or an ASBR in inter-AS VPN Option B networking:

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **ipv4-family vpnv4**
The BGP-VPNv4 address family view is displayed.
4. Run **tunnel-selector** *tunnel-selector-name*
The tunnel selector is applied to VPNv4 routes on the local device. The tunnel policy specified in the apply clause is applied to the VPNv4 routes that matching the if-match clause. The VPNv4 routes that are filtered out by the if-match clause are iterated to LSP tunnels.

Step 3 Apply the tunnel selector.

Apply the tunnel selector to VPNv4 routes.

Perform the following steps on an SPE device in HoVPN networking or an ASBR in inter-AS VPN Option B networking:

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **ipv4-family vpnv4**
The BGP-VPNv4 address family view is displayed.
4. Run **tunnel-selector** *tunnel-selector-name*
The tunnel selector is applied to VPNv4 routes on the local device. The tunnel policy specified in the apply clause is applied to the VPNv4 routes that matching the if-match clause. The VPNv4 routes that are filtered out by the if-match clause are iterated to LSP tunnels.

Apply the tunnel selector to labeled BGP-IPv4 routes.

Perform the following steps on a PE device or an ASBR in inter-AS VPN Option C networking:

1. Run **system-view**
The system view is displayed.
2. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
3. Run **tunnel-selector** *tunnel-selector-name*
The tunnel selector is applied to labeled BGP-IPv4 routes on the local device.
The tunnel policy specified in the apply clause is applied to the labeled BGP-IPv4 routes that matching the if-match clause. The labeled BGP-IPv4 routes that are filtered out by the if-match clause are iterated to LSP tunnels.

----End

Verifying the Configuration

After configuring and applying a tunnel selector, run the following commands to check information about the tunnel selector and tunnel policy specified in the tunnel selector.

- Run the **display tunnel-selector** *tunnel-selector-name* command to check detailed information about the tunnel selector.
- Run the **display tunnel-policy** *tunnel-policy-name* command to check information about the tunnel policy specified by the apply clause in the tunnel selector.
- Run the **display bgp vpnv4 all routing-table** *ipv4-address* [*mask* [**longer-prefixes**] | *mask-length* [**longer-prefixes**]] command to check tunnels selected for VPNv4 routes on the ASBR or SPE device.
- Run the **display ip routing-table** *ip-address* [*mask* | *mask-length*] [**longer-match**] **verbose** command to check the tunnels selected for labeled BGP-IPv4 routes on the PE device.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [*slots*] } command to check information about tunnels in the system.

8.7.16 Connecting a VPN to the Internet

Generally, users within a VPN cannot communicate with Internet users because VPN users cannot access the Internet. If each VPN site needs to access the Internet, configure the interconnection between the VPN and the Internet.

Pre-configuration Tasks

- [8.7.1 Configuring Basic BGP/MPLS IP VPN Functions](#)

Configuration Procedure

Step 1, step 2, and step 3 can be performed at any sequence.

Procedure

Step 1 Configure a static route on the CE device.

1. Run **system-view**

The system view is displayed.

2. Run **ip route-static** *ip-address* { *mask* | *mask-length* } { *interface-type interface-number* [*nexthop-address*] | *nexthop-address* } [**preference** *preference* | **tag** *tag*] * [**description** *text*]

The static route to a public network destination address is configured.

ip-address can be a public network address or 0.0.0.0. If the *dest-ip-address* is 0.0.0.0, the static route is also called the default route. The *mask* of a default route must be 0.0.0.0 or the *mask-length* of the default route must be 0. The out-interface must be the interface connected directly with the PE device, and the next-hop is the IP address of the peer PE interface connected directly with the CE device.

NOTE

If the CE and PE devices are connected through an Ethernet network, the next-hop must be specified.

Step 2 Configure a static VPN route to the Internet on the PE device.

1. Run **system-view**

The system view is displayed.

2. Run **ip route-static vpn-instance** *vpn-source-name destination-address { mask | mask-length }* *nexthop-address public [preference preference | tag tag] * [description text]*

A static route from the VPN to the Internet is configured and the next-hop address is a public network address.

Step 3 Configure a static route to the VPN on the PE device.

1. Run **system-view**

The system view is displayed.

2. Run **ip route-static** *ip-address { mask | mask-length }* *{ interface-type interface-number [nexthop-address] | vpn-instance vpn-instance-name nexthop-address | nexthop-address }* *[preference preference | tag tag] * [description text]*

The static route from the public network to the VPN is configured and the next-hop address is a private network address.

 **NOTE**

If the CE and PE devices are connected through an Ethernet network, the next-hop must be specified.

3. Advertise the static route to the Internet.

For detailed configuration, see the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*. For example, if OSPF is running between the PE device and the Internet, perform the following steps:

- a. Run **system-view**

The system view is displayed.

- b. Run **ospf** *[process-id]*

The OSPF view is displayed.

- c. Run **import-route static**

Static routes are imported into OSPF.

----End

Verifying the Configuration

- Run the **display ip routing-table vpn-instance** *vpn-instance-name* command to check the VPN routing table on the PE device. The command output shows that the route to the CE and the route to the destination device in the public network exist in the VPN routing table.
- Run the **display ip routing-table** command to check the routing table on the CE and the destination device in the public network. The command output shows that the CE has the route to the destination device in the public network and the destination device in the public network has the route to the CE.
- Run the **ping** command to check the connectivity between the CE and the destination device on the public network. The CE device and the destination device on the public network can ping each other.

8.8 Maintaining BGP/MPLS IP VPN

You can check route summary information in a VPN instance, monitor network connectivity, and reset BGP connections when maintaining a BGP/MPLS IP VPN network.

8.8.1 Collecting Statistics About L3VPN Traffic

Prerequisites

L3VPN traffic statistics collection is applicable to the interface traffic at the user side of a VPN. Before collecting L3VPN traffic statistics, you need to enable the L3VPN traffic statistics function.

NOTE

- Currently, L3VPN traffic statistics collection can count only unicast packets.
- In L3VPN over MPLS TE scenarios, if the device is enabled to collect L3VPN traffic statistics and traffic statistics on an MPLS TE tunnel interface simultaneously, packets received from the interface bound to a VPN instance are not counted as L3VPN traffic.
- Enabling L3VPN traffic statistics function may affect the forwarding performance. For example, when all interfaces provide line-speed forwarding, some interface may be unable to forward packets at line speed. Exercise caution when you enable traffic statistics on a VLANIF interface.
- L3VPN traffic statistics is unavailable for error packets.

Perform the following steps on the device:

Procedure

- Step 1** Run the **system-view** command to enter the system view.
- Step 2** Run the **ip vpn-instance *vpn-instance-name*** command to enter the VPN instance view.
- Step 3** Run the **traffic-statistics enable** command to enable the function of collecting statistics about L3VPN traffic.

----End

8.8.2 Checking L3VPN Traffic

Context

This function displays traffic statistics on the interface at the user side of the VPN. Note that traffic statistics are collected only after the L3VPN traffic statistics function is enabled.

Procedure

- Run the **display traffic-statistics vpn-instance *vpn-instance-name*** command to check the statistics about L3VPN traffic of a specified VPN instance.

----End

8.8.3 Clearing L3VPN Traffic

Context

Run the following command in the user view to clear L3VPN traffic statistics.



NOTICE

Statistics cannot be restored after being cleared. Therefore, use this command with caution.

Procedure

- Run the **reset traffic-statistics vpn-instance** { name *vpn-instance-name* | all } command in the user view to clear statistics about L3VPN traffic of a specified VPN instance or all VPN instances.

----End

8.8.4 Displaying BGP/MPLS IP VPN Information

Context

In routine maintenance, you can run the following commands in any view to check the status of BGP/MPLS IP VPN.

Procedure

- Run the **display ip vpn-instance** [verbose] [*vpn-instance-name*] command to check information about the VPN instance.
- Run the **display default-parameter l3vpn** command to check the default configuration of L3VPN during initialization.
- Run the **display ip routing-table vpn-instance** *vpn-instance-name* command to check the IP routing table of a VPN instance.
- Run the **display bgp vpnv4** { all | *vpn-instance* *vpn-instance-name* } **routing-table** [**statistics**] **label** command to check information about labeled routes in the BGP routing table.
- Run the **display bgp vpnv4** { all | **route-distinguisher** *route-distinguisher* | **vpn-instance** *vpn-instance-name* } **routing-table** *ipv4-address* [*mask* | *mask-length*] command to check information about the BGP VPNv4 routing table.
- Run the **display bgp vpnv4** { all | **route-distinguisher** *route-distinguisher* | **vpn-instance** *vpn-instance-name* } **routing-table** **statistics** command to check statistics about the BGP VPNv4 routing table.
- Run the **display bgp vpnv4** { all | **route-distinguisher** *route-distinguisher* | **vpn-instance** *vpn-instance-name* } **routing-table** command to check information about the BGP VPNv4 routing table.
- Run the **display bgp vpnv4** { all | *vpn-instance* *vpn-instance-name* } **group** [*group-name*] command to check information about the BGP VPNv4 peer group.

- Run the **display bgp vpnv4** { **all** | **vpn-instance** *vpn-instance-name* } **peer** [[*ipv4-address*] **verbose**] command to check BGP VPNv4 peer information.
- Run the **display bgp vpnv4** { **all** | **vpn-instance** *vpn-instance-name* } **network** command to check the routing information advertised by BGP VPNv4.
- Run the **display bgp vpnv4** { **all** | **vpn-instance** *vpn-instance-name* } **paths** [*as-regular-expression*] command to check the AS path information of BGP VPNv4.
- Run the **display bgp vpnv4 vpn-instance** *vpn-instance-name* **peer** { *group-name* | *ipv4-address* } **log-info** command to check the BGP peer's log information of a specified VPN instance.

----End

8.8.5 Checking Network Connectivity and Reachability

Context

After completing VPN configuration, you can:

- Run the **ping** command on the local CE to check whether the local CE and the remote CE in the same VPN can communicate with each other. If the ping fails, you can run the **tracert** command to locate the faulty node.
- Run the **ping** command with the **-vpn-instance** *vpn-instance-name* parameter on the PE to check whether the PE and the CE in the same VPN as the PE can communicate with each other. If the ping fails, you can run the **tracert** command with the **-vpn-instance** *vpn-instance-name* parameter to locate the faulty node.

If multiple interfaces on the PE are bound to the same VPN, you need to specify the source IP address, that is, the **-a** *source-ip-address* when you **ping** or **tracert** the remote CE that accesses the peer PE. If no source IP address is specified, the PE selects the smallest IP address from the IP addresses of the interfaces on the PE bound to this VPN as the source address of the Internet Control Message Protocol (ICMP) messages. If the CE has no route to the selected IPv4 route, the CE discards the returned ICMP message.

NOTE

By default, as for the MPLS time to live (MPLS TTL) timeout packet with a single label, the router returns the ICMP message according to the local IP route (that is, the public network route). However, no VPN route exists in the public network routing table of the ASBR and therefore, the ICMP message is discarded when being sent to or returned by the ASBR.

Procedure

- Run the **ping** [**ip**] [**-a** *source-ip-address* | **-c** *count* | **-d** | **-f** | **-h** *ttl-value* | [**-i** *interface-type interface-number* | **-si** *source-interface-type source-interface-number*] | **-m** *time* | **-n** | **-name** | **-p** *pattern* | **-q** | **-r** | **-s** *packet-size* | **-system-time** | **-t** *timeout* | **-tos** *tos-value* | **-v** | **-vpn-instance** *vpn-instance-name* | **ignore-mtu**] * *host* [**ip-forwarding**] command to check network connectivity from the local device to a specified destination IP address.
- Run the **tracert** [**-a** *source-ip-address* | **-f** *first-ttl* | **-m** *max-ttl* | **-name** | **-p** *port* | **-q** *nqueries* | **-vpn-instance** *vpn-instance-name* | **-w** *timeout* | **-v**] * *host* command to check the gateways that a data packet passes when it is sent from the local device to the destination.
- Run the **ping lsp** [**-a** *source-ip* | **-c** *count* | **-exp** *exp-value* | **-h** *ttl-value* | **-m** *interval* | **-r** *reply-mode* | **-s** *packet-size* | **-t** *time-out* | **-v**] * **ip** *destination-address mask-length* [*ip-*

address] [**nexthop** *nexthop-address* | **draft6**] command to check connectivity of an Label Switched Path (LSP).

- Run the **tracert lsp** [**-a** *source-ip* | **-exp** *exp-value* | **-h** *ttl-value* | **-r** *reply-mode* | **-t** *time-out* | **-v**] * **ip** *destination-address mask-length* [*ip-address*] [**nexthop** *nexthop-address* | **draft6**] command to check the gateways that a data packet passes when it is sent from the local device to the destination along the LSP.

----End

8.8.6 Viewing the Integrated Route Statistics of IPv4 VPN Instances

Procedure

- Run the **display ip routing-table vpn-instance** *vpn-instance-name* **statistics** command to check the integrated route statistics of an IPv4 VPN instance.
- Run the **display ip routing-table all-vpn-instance** **statistics** command to check the integrated route statistics of all IPv4 VPN instances.

----End

8.8.7 Resetting BGP Statistics of a VPN Instance IPv4 Address Family

Procedure

- Run the **reset bgp vpn-instance** *vpn-instance-name* **ipv4-family** [*ipv4-address*] **flap-info** command in the user view to clear statistics of the BGP peer flap for a specified VPN instance IPv4 address family.
- Run the **reset bgp vpn-instance** *vpn-instance-name* **ipv4-family dampening** [*ipv4-address* [*mask* | *mask-length*]] command in the user view to clear dampening information of the VPN instance IPv4 address family.

----End

8.8.8 Resetting BGP Connections

Context



NOTICE

VPN services are interrupted after the BGP connection is reset. Exercise caution when running the commands.

When the BGP configuration changes, you can use the soft reset or reset BGP connections to let the new configurations take effect. A soft reset requires that the BGP peers have route refreshment capability (supporting Route-Refresh messages).

Procedure

- Run the **refresh bgp vpn-instance** *vpn-instance-name* **ipv4-family** { **all** | *ipv4-address* | **group** *group-name* | **internal** | **external** } **import** command in the user view to trigger the inbound soft reset of the VPN instance IPv4 address family's BGP connection.
- Run the **refresh bgp vpn-instance** *vpn-instance-name* **ipv4-family** { **all** | *ipv4-address* | **group** *group-name* | **internal** | **external** } **export** command in the user view to trigger the outbound soft reset of the VPN instance IPv4 address family's BGP connection.
- Run the **refresh bgp vpnv4** { **all** | *ipv4-address* | **group** *group-name* | **internal** | **external** } **import** command in the user view to trigger the inbound soft reset of the BGP VPNv4 connection.
- Run the **refresh bgp vpnv4** { **all** | *ipv4-address* | **group** *group-name* | **internal** | **external** } **export** command in the user view to trigger the outbound soft reset of the BGP VPNv4 connection.
- Run the **reset bgp vpn-instance** *vpn-instance-name* **ipv4-family** { *as-number* | *ipv4-address* | **group** *group-name* | **all** | **internal** | **external** } command in the user view to reset BGP connections of the VPN instance IPv4 address family.
- Run the **reset bgp vpnv4** { *as-number* | *ipv4-address* | **group** *group-name* | **all** | **internal** | **external** } command in the user view to reset BGP VPNv4 connections.

----End

8.8.9 Monitoring the Running Status of VPN Tunnels

Context

In routine maintenance, run the following commands in any check to check the tunnel status.

Procedure

- Run the **display interface tunnel** *interface-number* command to check information about a specified tunnel interface.
- Run the **display tunnel-info tunnel-id** *tunnel-id* command to check detailed information about a specified tunnel.
- Run the **display tunnel-info all** command to check information about all tunnels.
- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check the configuration of a tunnel policy.
- Run the **display ip vpn-instance verbose** [*vpn-instance-name*] command to check information about the tunnel policy applied to a VPN instance.
- Run the **display ip routing-table vpn-instance** *vpn-instance-name* [*ip-address*] **verbose** command to check the tunnel to which VPN routes are iterated.

----End

8.9 Configuration Examples for BGP/MPLS IP VPN

This section provides several configuration examples of BGP/MPLS IP VPN networking. In each configuration example, the networking requirements, configuration roadmap, configuration procedures, and configuration files are provided.

8.9.1 Example for Configuring BGP/MPLS IP VPN

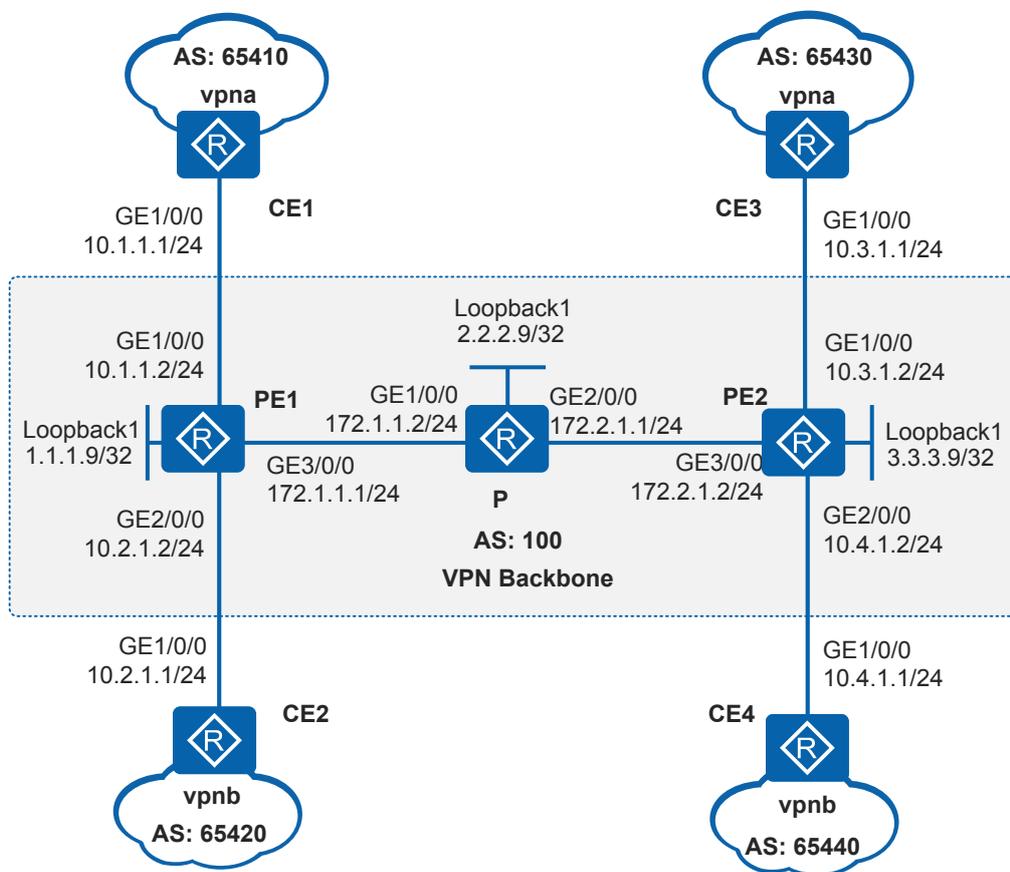
Networking Requirements

As shown in [Figure 8-42](#):

- CE1 connects to the headquarters R&D area of a company, and CE3 connects to the branch R&D area. CE1 and CE3 belong to vpna.
- CE2 connects to the headquarters non-R&D area, and CE4 connects to the branch non-R&D area. CE2 and CE4 belong to vpnb.

BGP/MPLS IP VPN needs to be deployed for the company to ensure secure communication between the headquarters and branches.

Figure 8-42 Networking diagram for configuring BGP/MPLS IP VPN



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the P and PEs to ensure IP connectivity on the backbone network.
2. Configure basic MPLS capabilities and MPLS LDP on the P and PEs to set up MPLS LSP tunnels for VPN data transmission on the backbone network.

3. Configure VPN instances `vpna` and `vpnb` on PE1 and PE2. Set the VPN target of `vpna` to 111:1 and the VPN target of `vpnb` to 222:2. This configuration allows users in the same VPN to communicate with each other and isolates users in different VPNs. Bind the VPN instance to the PE interfaces connected to CEs to provide access for VPN users.
4. Configure MP-IBGP on PE1 and PE2 to enable them to exchange VPN routing information.
5. Configure EBGP on the CEs and PEs to exchange VPN routing information.

Procedure

- Step 1** Configure OSPF on the MPLS backbone network so that the PEs and Ps can communicate with each other.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip address 172.1.1.1 24
[PE1-GigabitEthernet3/0/0] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Configure P.

```
<Huawei> system-view
[Huawei] sysname P
[P] interface loopback 1
[P-LoopBack1] ip address 2.2.2.9 32
[P-LoopBack1] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] ip address 172.2.1.1 24
[P-GigabitEthernet2/0/0] quit
[P] ospf
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[P-ospf-1-area-0.0.0.0] quit
[P-ospf-1] quit
```

Configure PE2.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface loopback 1
[PE2-LoopBack1] ip address 3.3.3.9 32
[PE2-LoopBack1] quit
[PE2] interface gigabitethernet 3/0/0
[PE2-GigabitEthernet3/0/0] ip address 172.2.1.2 24
[PE2-GigabitEthernet3/0/0] quit
[PE2] ospf
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] network 3.3.3.9 0.0.0.0
```

```
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

After the configuration is complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. The command output shows that the neighbor status is Full. Run the **display ip routing-table** command. The command output shows that PEs have learned the routes to Loopback1 of each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
Destinations : 11          Routes : 11

Destination/Mask    Proto Pre  Cost    Flags NextHop         Interface
-----
1.1.1.9/32          Direct 0    0        D  127.0.0.1         LoopBack1
2.2.2.9/32          OSPF   10   1        D  172.1.1.2
GigabitEthernet3/0/0
3.3.3.9/32          OSPF   10   2        D  172.1.1.2
GigabitEthernet3/0/0
127.0.0.0/8         Direct 0    0        D  127.0.0.1         InLoopBack0
127.0.0.1/32        Direct 0    0        D  127.0.0.1         InLoopBack0
127.255.255.255/32 Direct 0    0        D  127.0.0.1         InLoopBack0
172.1.1.0/24        Direct 0    0        D  172.1.1.1
GigabitEthernet3/0/0
172.1.1.1/32        Direct 0    0        D  127.0.0.1
GigabitEthernet3/0/0
172.1.1.255/32     Direct 0    0        D  127.0.0.1
GigabitEthernet3/0/0
172.2.1.0/24        OSPF   10   2        D  172.1.1.2
GigabitEthernet3/0/0
255.255.255.255/32 Direct 0    0        D  127.0.0.1         InLoopBack0
[PE1] display ospf peer

OSPF Process 1 with Router ID 1.1.1.9
Neighbors

Area 0.0.0.0 interface 172.1.1.1(GigabitEthernet3/0/0)'s neighbors
Router ID: 2.2.2.9      Address: 172.1.1.2
State: Full Mode:Nbr is Master Priority: 1
DR: 172.1.1.1 BDR: 172.1.1.2 MTU: 0
Dead timer due in 37 sec
Retrans timer interval: 5
Neighbor is up for 00:16:21
Authentication Sequence: [ 0 ]
```

Step 2 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mppls] quit
[PE1] mpls ldp
[PE1-mppls-ldp] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls
[PE1-GigabitEthernet3/0/0] mpls ldp
[PE1-GigabitEthernet3/0/0] quit
```

Configure P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
```

```
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 3/0/0
[PE2-GigabitEthernet3/0/0] mpls
[PE2-GigabitEthernet3/0/0] mpls ldp
[PE2-GigabitEthernet3/0/0] quit
```

After the configuration is complete, LDP sessions can be set up between PE1 and the P and between the P and PE2. Run the **display mpls ldp session** command. The command output shows that the **Status** field is **Operational**. Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status          LAM  SsnRole  SsnAge         KASent/Rcv
-----
2.2.2.9:0        Operational    DU   Active   0000:00:01     6/6
-----
TOTAL: 1 session(s) Found.
[PE1] display mpls ldp lsp

LDP LSP Information
-----
DestAddress/Mask  In/OutLabel    UpstreamPeer    NextHop         OutInterface
-----
1.1.1.9/32        3/NULL         2.2.2.9         127.0.0.1      InLoop0
*1.1.1.9/32       Liberal/1024
2.2.2.9/32        NULL/3         -               172.1.1.2      GE3/0/0
2.2.2.9/32        1024/3         2.2.2.9         172.1.1.2      GE3/0/0
3.3.3.9/32        NULL/1025      -               172.1.1.2      GE3/0/0
3.3.3.9/32        1025/1025     2.2.2.9         172.1.1.2      GE3/0/0
-----
TOTAL: 5 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
```

Step 3 Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

Configure PE1.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
```

```
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpb
[PE1-GigabitEthernet2/0/0] ip address 10.2.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 200:1
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv4-family
[PE2-vpn-instance-vpb-af-ipv4] route-distinguisher 200:2
[PE2-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE2-vpn-instance-vpb-af-ipv4] quit
[PE2-vpn-instance-vpb] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet1/0/0] ip address 10.3.1.2 24
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpb
[PE2-GigabitEthernet2/0/0] ip address 10.4.1.2 24
[PE2-GigabitEthernet2/0/0] quit
```

Assign IP addresses to interfaces on CEs according to [Figure 8-42](#).

Configure CE1. The configurations of CE2, CE3, and CE4 are similar to the configuration of CE1, and are not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances. Each PE can ping its connected CE.

NOTE

If a PE has multiple interfaces bound to the same VPN instance, specify a source IP addresses by setting **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping the remote CE. If the source IP address is not specified, the ping operation fails.

The information displayed on PE1 is used as an example.

```
[PE1] display ip vpn-instance verbose
Total VPN-Instances configured : 2
Total IPv4 VPN-Instances configured : 2
Total IPv6 VPN-Instances configured : 0
```

```

VPN-Instance Name and ID : vpna, 1
  Interfaces : GigabitEthernet1/0/0
Address family ipv4
Create date : 2012/07/25 00:58:17
Up time : 0 days, 22 hours, 24 minutes and 53 seconds
Route Distinguisher : 100:1
Export VPN Targets : 111:1
Import VPN Targets : 111:1
Label Policy : label per route
Log Interval : 5

VPN-Instance Name and ID : vpnb, 2
  Interfaces : GigabitEthernet2/0/0
Address family ipv4
Create date : 2012/07/25 00:58:17
Up time : 0 days, 22 hours, 24 minutes and 53 seconds
Route Distinguisher : 100:2
Export VPN Targets : 222:2
Import VPN Targets : 222:2
Label Policy : label per route
Log Interval : 5
[PE1] ping -vpn-instance vpna 10.1.1.1
PING 10.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=5 ms
  Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=3 ms
  Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=3 ms
  Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=3 ms
  Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=16 ms

--- 10.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/6/16 ms

```

Step 4 Set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```

[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit

```

Configure PE2.

```

[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit

```

After the configuration is complete, run the **display bgp peer** or **display bgp vpnv4 all peer** command on the PEs. The command output shows that BGP peer relationships have been established between the PEs.

```

[PE1] display bgp peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V   AS  MsgRcvd  MsgSent  OutQ  Up/Down
State         PrefRcv

```

```

3.3.3.9      4    100      12      6      0 00:02:21
Established  0
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down  State
PrefRcv
3.3.3.9      4    100    12      18      0    00:09:38  Established  0
  
```

Step 5 Set up EBGP peer relationships between the PEs and CEs and import VPN routes into BGP.

Configure CE1. The configurations of CE2, CE3, and CE4 are similar to the configuration of CE1, and are not mentioned here.

```

[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
  
```

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```

[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] quit
[PE1-bgp] ipv4-family vpn-instance vpnb
[PE1-bgp-vpnb] peer 10.2.1.1 as-number 65420
[PE1-bgp-vpnb] import-route direct
[PE1-bgp-vpnb] quit
[PE1-bgp] quit
  
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance peer** command on the PEs. The command output shows that BGP peer relationships have been established between the PEs and CEs.

The peer relationship between PE1 and CE1 is used as an example.

```

[PE1] display bgp vpnv4 vpn-instance vpna peer

BGP local router ID : 1.1.1.9
Local AS number : 100

VPN-Instance vpna, Router ID 1.1.1.9:
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down  State
PrefRcv
10.1.1.1      4    65410    6      3      0 00:00:02
Established  4
  
```

Step 6 Verify the configuration.

Run the **display ip routing-table vpn-instance** command on the PEs to view the routes to the remote CEs.

The information displayed on PE1 is used as an example.

```

[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
  
```

```

-----
Routing Tables: vpna
      Destinations : 5          Routes : 5

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
      10.1.1.0/24   Direct  0    0           D   10.1.1.2
GigabitEthernet1/0/0
      10.1.1.2/32   Direct  0    0           D   127.0.0.1
GigabitEthernet1/0/0
      10.1.1.255/32 Direct  0    0           D   127.0.0.1
GigabitEthernet1/0/0
      10.3.1.0/24   IBGP    255  0          RD   3.3.3.9
GigabitEthernet3/0/0
255.255.255.255/32 Direct  0    0           D   127.0.0.1      InLoopBack0
[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----

Routing Tables: vpnb
      Destinations : 5          Routes : 5

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
      10.2.1.0/24   Direct  0    0           D   10.2.1.2
GigabitEthernet2/0/0
      10.2.1.2/32   Direct  0    0           D   127.0.0.1
GigabitEthernet2/0/0
      10.2.1.255/32 Direct  0    0           D   127.0.0.1
GigabitEthernet2/0/0
      10.4.1.0/24   IBGP    255  0          RD   3.3.3.9
GigabitEthernet3/0/0
255.255.255.255/32 Direct  0    0           D   127.0.0.1      InLoopBack0
  
```

CEs in the same VPN can ping each other, whereas CEs in different VPNs cannot.

For example, CE1 can ping CE3 at 10.3.1.1 but cannot ping CE4 at 10.4.1.1.

```

[CE1] ping 10.3.1.1
PING 10.3.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.1: bytes=56 Sequence=1 ttl=253 time=72 ms
  Reply from 10.3.1.1: bytes=56 Sequence=2 ttl=253 time=34 ms
  Reply from 10.3.1.1: bytes=56 Sequence=3 ttl=253 time=50 ms
  Reply from 10.3.1.1: bytes=56 Sequence=4 ttl=253 time=50 ms
  Reply from 10.3.1.1: bytes=56 Sequence=5 ttl=253 time=34 ms
--- 10.3.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 34/48/72 ms
[CE1] ping 10.4.1.1
PING 10.4.1.1: 56 data bytes, press CTRL_C to break
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
--- 10.4.1.1 ping statistics ---
  5 packet(s) transmitted
  0 packet(s) received
  100.00% packet loss
  
```

---End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
```

```
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 100:2
    vpn-target 222:2 export-extcommunity
    vpn-target 222:2 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpna
  ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpnb
  ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
  ip address 172.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpna
    import-route direct
    peer 10.1.1.1 as-number 65410
#
  ipv4-family vpn-instance vpnb
    import-route direct
    peer 10.2.1.1 as-number 65420
#
ospf 1
  area 0.0.0.0
    network 1.1.1.9 0.0.0.0
    network 172.1.1.0 0.0.0.255
#
return
```

● P configuration file

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
```

```
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 172.2.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 172.1.1.0 0.0.0.255
  network 172.2.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
 sysname PE2
#
ip vpn-instance
vpna
 ipv4-family
  route-distinguisher 200:1
  vpn-target 111:1 export-extcommunity
  vpn-target 111:1 import-extcommunity
#
ip vpn-instance
vpnb
 ipv4-family
  route-distinguisher 200:2
  vpn-target 222:2 export-extcommunity
  vpn-target 222:2 import-extcommunity
#
 mpls lsr-id 3.3.3.9
 mpls
#
 mpls ldp
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpna
 ip address 10.3.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpnb
 ip address 10.4.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 172.2.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
 peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
```

```
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpna
import-route direct
peer 10.3.1.1 as-number 65430
#
ipv4-family vpn-instance vpnb
import-route direct
peer 10.4.1.1 as-number 65440
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 172.2.1.0 0.0.0.255
#
return
```

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
bgp 65420
peer 10.2.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.2.1.2 enable
#
return
```

- CE3 configuration file

```
#
sysname CE3
#
interface GigabitEthernet1/0/0
ip address 10.3.1.1 255.255.255.0
#
bgp 65430
peer 10.3.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.3.1.2 enable
#
return
```

- CE4 configuration file

```
#
 sysname CE4
#
interface GigabitEthernet1/0/0
 ip address 10.4.1.1 255.255.255.0
#
bgp 65440
 peer 10.4.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.4.1.2 enable
#
return
```

8.9.2 Example for Configuring BGP/MPLS IP VPNs with Overlapping Address Spaces

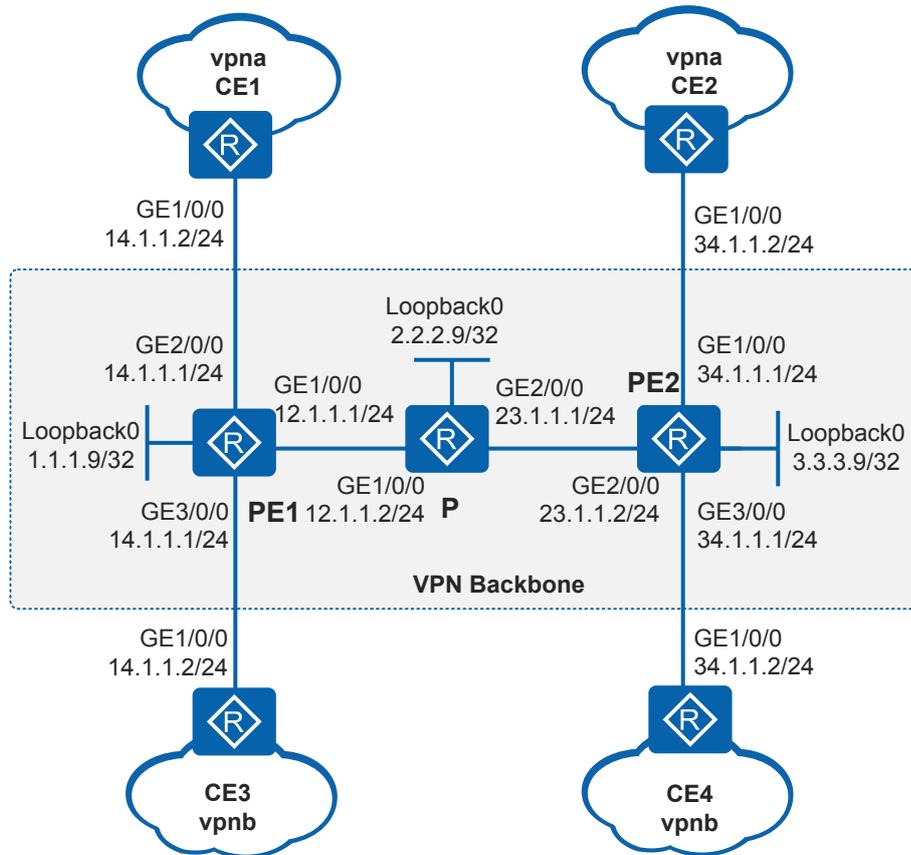
Networking Requirements

As shown in [Figure 8-43](#):

- CE1 connects to the headquarters R&D area of a company, and CE2 connects to the branch R&D area. CE1 and CE2 belong to vpna.
- CE3 connects to the headquarters non-R&D area, and CE4 connects to the branch non-R&D area. CE3 and CE4 belong to vpnb.
- The headquarters and branches use overlapping address spaces.

The company wants to ensure secure communication between the headquarters and branches and isolate the R&D areas from non-R&D areas, without changing the current network deployment.

Figure 8-43 Networking diagram for configuring BGP/MPLS IP VPNs with overlapping address spaces



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the P and PEs to ensure IP connectivity on the backbone network.
2. Configure basic MPLS capabilities and MPLS LDP on the P and PEs to set up MPLS LSP tunnels for VPN data transmission on the backbone network.
3. Configure MP-IBGP on PE1 and PE2 to enable them to exchange VPN routing information.
4. Configure VPN instances *vpna* and *vpnb* on PE1 and PE2. Set the VPN target of *vpna* to 100:100 and the VPN target of *vpnb* to 200:200. This configuration allows users in the same VPN to communicate with each other and isolates users in different VPNs. Bind the VPN instance to the PE interfaces connected to CEs to provide access for VPN users.
5. Configure static routes on the CEs and PEs to exchange VPN routing information.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-43](#).

Configure PE1. The configuration on PE2, P, and CE1 to CE4 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 1.1.1.9 32
[PE1-LoopBack0] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 12.1.1.1 24
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure OSPF on the MPLS backbone network so that the PEs and Ps can communicate with each other.

Configure PE1.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 12.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Configure P.

```
[P] ospf
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[P-ospf-1-area-0.0.0.0] network 12.1.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] network 23.1.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] quit
[P-ospf-1] quit
```

Configure PE2.

```
[PE2] ospf
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 3.3.3.9 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] network 23.1.1.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

After the configuration is complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. The command output shows that the neighbor status is Full. Run the **display ip routing-table** command. The command output shows that PEs have learned the routes to Loopback0 of each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
Destinations : 11          Routes : 11

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
1.1.1.9/32          Direct  0    0        D  127.0.0.1         LoopBack0
2.2.2.9/32          OSPF   10    1        D  12.1.1.2
GigabitEthernet1/0/0
3.3.3.9/32          OSPF   10    2        D  12.1.1.2
GigabitEthernet1/0/0
12.1.1.0/24         Direct  0    0        D  12.1.1.1
GigabitEthernet1/0/0
12.1.1.1/32         Direct  0    0        D  127.0.0.1
GigabitEthernet1/0/0
12.1.1.255/32       Direct  0    0        D  127.0.0.1
GigabitEthernet1/0/0
23.1.1.0/24         OSPF   10    2        D  12.1.1.2
GigabitEthernet1/0/0
```

127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0
255.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0

Step 3 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls
[PE2-GigabitEthernet2/0/0] mpls ldp
[PE2-GigabitEthernet2/0/0] quit
```

After the configuration is complete, LDP sessions can be set up between PE1 and the P and between the P and PE2. Run the **display mpls ldp session** command. The command output shows that the **Status** field is **Operational**. Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status          LAM  SsnRole  SsnAge         KASent/Rcv
-----
2.2.2.9:0        Operational    DU   Active   0000:00:01     6/6
-----
TOTAL: 1 session(s) Found.

[PE1] display mpls ldp lsp
```

```
LDP LSP Information
-----
DestAddress/Mask  In/OutLabel  UpstreamPeer  NextHop  OutInterface
-----
1.1.1.9/32        3/NULL       2.2.2.9       127.0.0.1  InLoop0
*1.1.1.9/32       Liberal/1024  -              DS/2.2.2.9
2.2.2.9/32        NULL/3       -              12.1.1.2   GE1/0/0
2.2.2.9/32        1024/3       2.2.2.9       12.1.1.2   GE1/0/0
3.3.3.9/32        NULL/1025    -              12.1.1.2   GE1/0/0
3.3.3.9/32        1025/1025    2.2.2.9       12.1.1.2   GE1/0/0
-----
TOTAL: 5 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
```

Step 4 Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

Configure PE1.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:100
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:100 export-extcommunity
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:100 import-extcommunity
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] route-distinguisher 300:300
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 200:200 export-extcommunity
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 200:200 import-extcommunity
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet2/0/0] ip address 14.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip binding vpn-instance vpb
[PE1-GigabitEthernet3/0/0] ip address 14.1.1.1 255.255.255.0
[PE1-GigabitEthernet3/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 200:200
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 100:100 export-extcommunity
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 100:100 import-extcommunity
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv4-family
[PE2-vpn-instance-vpb-af-ipv4] route-distinguisher 400:400
[PE2-vpn-instance-vpb-af-ipv4] vpn-target 200:200 export-extcommunity
[PE2-vpn-instance-vpb-af-ipv4] vpn-target 200:200 import-extcommunity
[PE2-vpn-instance-vpb-af-ipv4] quit
[PE2-vpn-instance-vpb] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet1/0/0] ip address 34.1.1.1 255.255.255.0
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface gigabitethernet 3/0/0
[PE2-GigabitEthernet3/0/0] ip binding vpn-instance vpb
```

```
[PE2-GigabitEthernet3/0/0] ip address 34.1.1.1 255.255.255.0
[PE2-GigabitEthernet3/0/0] quit
```

Assign IP addresses to interfaces on CEs according to [Figure 8-43](#).

Configure CE1. The configurations of CE2, CE3, and CE4 are similar to the configuration of CE1, and are not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 14.1.1.2 24
[CE1-GigabitEthernet1/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances. Each PE can ping its connected CE.

The information displayed on PE1 is used as an example.

```
[PE1] display ip vpn-instance verbose
Total VPN-Instances configured : 2
Total IPv4 VPN-Instances configured : 2
Total IPv6 VPN-Instances configured : 0

VPN-Instance Name and ID : vpna, 1
  Interfaces : GigabitEthernet2/0/0
  Address family ipv4
  Create date : 2012/07/25 00:58:17 UTC+08:00
  Up time : 0 days, 22 hours, 24 minutes and 53 seconds
  Route Distinguisher : 100:100
  Export VPN Targets : 100:100
  Import VPN Targets : 100:100
  Label Policy : label per route
  Log Interval : 5

VPN-Instance Name and ID : vpnb, 2
  Interfaces : GigabitEthernet3/0/0
  Address family ipv4
  Create date : 2012/07/25 00:58:17 UTC+08:00
  Up time : 0 days, 22 hours, 24 minutes and 53 seconds
  Route Distinguisher : 300:300
  Export VPN Targets : 200:200
  Import VPN Targets : 200:200
  Label Policy : label per route
  Log Interval : 5
[PE1] ping -vpn-instance vpna 14.1.1.2
PING 14.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 14.1.1.2: bytes=56 Sequence=1 ttl=255 time=5 ms
  Reply from 14.1.1.2: bytes=56 Sequence=2 ttl=255 time=3 ms
  Reply from 14.1.1.2: bytes=56 Sequence=3 ttl=255 time=3 ms
  Reply from 14.1.1.2: bytes=56 Sequence=4 ttl=255 time=3 ms
  Reply from 14.1.1.2: bytes=56 Sequence=5 ttl=255 time=16 ms

--- 14.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/6/16 ms
```

Step 5 Set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
```

```
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] quit
[PE1-bgp] ipv4-family vpn-instance vpb
[PE1-bgp-vpb] import-route direct
[PE1-bgp-vpb] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 0
[PE2-bgp] ipv4-family vpv4
[PE2-bgp-af-vpv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpv4] quit
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] import-route direct
[PE2-bgp-vpna] quit
[PE2-bgp] ipv4-family vpn-instance vpb
[PE2-bgp-vpb] import-route direct
[PE2-bgp-vpb] quit
[PE2-bgp] quit
```

After the configuration is complete, run the **display bgp peer** command on the PEs. The command output shows that a BGP peer relationship has been set up between the PEs.

```
[PE1] display bgp peer
BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V   AS  MsgRcvd  MsgSent  OutQ  Up/Down      State      PrefRcv
-----
3.3.3.9      4  100      3         3       0  00:01:08    Established
0
```

Step 6 On CE1, CE2, CE3, and CE4, configure static routes to their connected PEs.

Configure CE1. The configurations of CE2, CE3, and CE4 are similar to the configuration of CE1, and are not mentioned here.

```
[CE1] ip route-static 0.0.0.0 0.0.0.0 gigabitethernet 1/0/0 14.1.1.1
```

Step 7 Verify the configuration.

Run the **display ip routing-table vpn-instance** command on the PEs to view the routes to the remote CEs.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
Destinations : 5          Routes : 5

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
14.1.1.0/24         Direct  0    0        D  14.1.1.1
GigabitEthernet2/0/0
14.1.1.1/32         Direct  0    0        D  127.0.0.1
GigabitEthernet2/0/0
14.1.1.255/32       Direct  0    0        D  127.0.0.1
GigabitEthernet2/0/0
34.1.1.0/24         IBGP   255  0        RD  3.3.3.9
```

```
GigabitEthernet1/0/0
255.255.255.255/32   Direct 0    0          D 127.0.0.1      InLoopBack0
[PE1] display ip routing-table vpn-instance vpnb
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpnb
      Destinations : 5          Routes : 5

  Destination/Mask   Proto  Pre  Cost   Flags NextHop         Interface
-----
          14.1.1.0/24   Direct  0    0          D 14.1.1.1
GigabitEthernet3/0/0
          14.1.1.1/32   Direct  0    0          D 127.0.0.1
GigabitEthernet3/0/0
          14.1.1.255/32 Direct  0    0          D 127.0.0.1
GigabitEthernet3/0/0
          34.1.1.0/24   IBGP    255  0          RD 3.3.3.9
GigabitEthernet1/0/0
255.255.255.255/32   Direct  0    0          D 127.0.0.1      InLoopBack0
```

Run the **ping 34.1.1.2** command on CE1, and the ping is successful. Run the **display interface** command on PE2 to view traffic statistics on GE1/0/0 and GE3/0/0. The command output shows that there are packets passing through GE1/0/0 but no packet passing through GE3/0/0. This indicates that the two VPN instances have overlapping address spaces but they are isolated from each other.

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:100
    vpn-target 100:100 export-extcommunity
    vpn-target 100:100 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 300:300
    vpn-target 200:200 export-extcommunity
    vpn-target 200:200 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 12.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpna
  ip address 14.1.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance vpnb
  ip address 14.1.1.1 255.255.255.0
#
interface LoopBack0
  ip address 1.1.1.9 255.255.255.255
```

```
#
bgp 100
peer 3.3.3.9 as-number 100
peer 3.3.3.9 connect-interface LoopBack0
#
ipv4-family unicast
undo synchronization
peer 3.3.3.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 3.3.3.9 enable
#
ipv4-family vpn-instance vpna
import-route direct
#
ospf 1
area 0.0.0.0
network 12.1.1.0 0.0.0.255
network 1.1.1.9 0.0.0.0
#
return
```

● P configuration file

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 12.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 23.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 12.1.1.0 0.0.0.255
network 23.1.1.0 0.0.0.255
network 2.2.2.9 0.0.0.0
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 200:200
vpn-target 100:100 export-extcommunity
vpn-target 100:100 import-extcommunity
#
ip vpn-instance vpnb
ipv4-family
route-distinguisher 400:400
vpn-target 200:200 export-extcommunity
vpn-target 200:200 import-extcommunity
#
mpls lsr-id 3.3.3.9
```

```
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpna
 ip address 34.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 23.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance vpnb
 ip address 34.1.1.1 255.255.255.0
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack0
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.9 enable
#
 ipv4-family vpn-instance vpna
  import-route direct
#
 ipv4-family vpn-instance vpnb
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 23.1.1.0 0.0.0.255
  network 3.3.3.9 0.0.0.0
#
return
```

● CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 14.1.1.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet 1/0/0 14.1.1.1
#
return
```

● CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 34.1.1.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet 1/0/0 34.1.1.1
#
return
```

● CE3 configuration file

```
#
sysname CE3
```

```
#
interface GigabitEthernet1/0/0
ip address 14.1.1.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet 1/0/0 14.1.1.1
#
return
```

- CE4 configuration file

```
#
sysname CE4
#
interface GigabitEthernet1/0/0
ip address 34.1.1.2 255.255.255.0
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet 1/0/0 34.1.1.1
#
return
```

8.9.3 Example for Configuring Communication Between Local VPNs

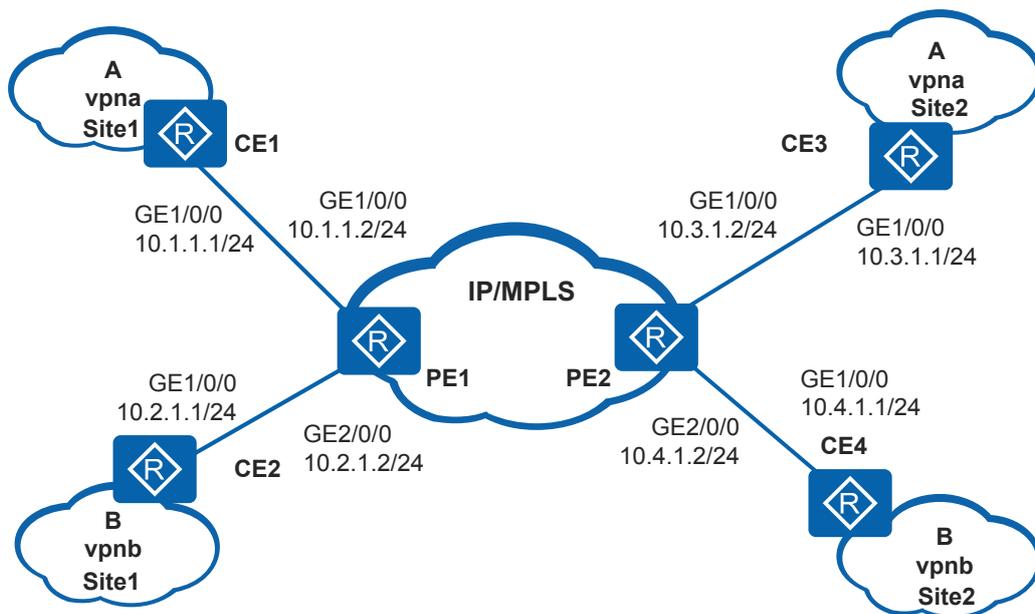
Networking Requirements

As shown in **Figure 8-44**, company A and company B realize communication between their respective headquarters and branches through BGP/MPLS IP VPN. The network deployment is as follows:

- CE1 connects to the headquarters of company A, and CE3 connects to the branches of company A. CE1 and CE3 belong to vpna.
- CE2 connects to the headquarters of company B, and CE4 connects to the branches of company B. CE2 and CE4 belong to vpnb.

Headquarters of company A and headquarters of company B need to communicate with each other for business.

Figure 8-44 Networking diagram for configuring communication between local VPNs



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure VPN instances on PE1 and configure different VPN targets for the instances to isolate VPNs.
2. On PE1, bind the VPN instances to the interfaces connected to CEs to provide access for VPN users.
3. Import direct routes to the local CEs into the VPN routing table on PE1. On each CE connected to PE1, configure a static route to the other local CE to enable the CEs to communicate with each other.

Procedure

Step 1 # Assign IP addresses to interfaces on CEs according to [Figure 8-44](#).

Configure CE1. The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 export-extcommunity
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 222:2 import-extcommunity
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 222:2 export-extcommunity
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 222:2 111:1 import-extcommunity
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpb
[PE1-GigabitEthernet2/0/0] ip address 10.2.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

Each PE can ping its connected CE. The information displayed on PE1 and CE1 is used as an example.

```
[PE1] ping -vpn-instance vpna 10.1.1.1
PING 10.1.1.1: 56 data bytes, press CTRL_C to break
Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=5 ms
```

```

Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=3 ms
Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=3 ms
Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=3 ms
Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=16 ms

--- 10.1.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 3/6/16 ms
  
```

Step 3 Configure BGP and import the direct routes to local CEs to the VPN routing table.

Configure PE1.

```

[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] quit
[PE1-bgp] ipv4-family vpn-instance vpb
[PE1-bgp-vpb] import-route direct
[PE1-bgp-vpb] quit
[PE1-bgp] quit
  
```

Step 4 Configure static routes on the CEs.

Configure CE1.

```

[CE1] ip route-static 10.2.1.0 24 10.1.1.2
  
```

Configure CE2.

```

[CE2] ip route-static 10.1.1.0 24 10.2.1.2
  
```

Step 5 Verify the configuration.

After the configuration is complete, run the **display ip routing-table vpn-instance vpna** command on PE1. The command output shows that the VPNs have imported routes of each other. The VPN instance vpna is used as an example.

```

[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
      Destinations : 6          Routes : 6

Destination/Mask    Proto  Pre  Cost           Flags NextHop         Interface
-----
      10.1.1.0/24    Direct  0    0                D    10.1.1.2
GigabitEthernet1/0/0
      10.1.1.2/32    Direct  0    0                D    127.0.0.1
GigabitEthernet1/0/0
      10.1.1.255/32  Direct  0    0                D    127.0.0.1
GigabitEthernet1/0/0
      10.2.1.0/24    BGP     255  0                D    10.2.1.2
GigabitEthernet2/0/0
      10.2.1.2/32    BGP     255  0                D    127.0.0.1          InLoopBack0
255.255.255.255/32  Direct  0    0                D    127.0.0.1          InLoopBack0
  
```

CE1 and CE2 can ping each other.

```

[CE1] ping 10.2.1.1
PING 10.2.1.1: 56 data bytes, press CTRL_C to break
Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=253 time=72 ms
Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=253 time=34 ms
Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=253 time=50 ms
Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=253 time=50 ms
Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=253 time=34 ms
  
```

```
--- 10.2.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 34/48/72 ms
```

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 222:2 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 100:2
    vpn-target 222:2 export-extcommunity
    vpn-target 222:2 111:1 import-extcommunity
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpna
  ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpnb
  ip address 10.2.1.2 255.255.255.0
#
bgp 100
#
  ipv4-family unicast
    undo synchronization
#
  ipv4-family vpn-instance vpna
    import-route direct
#
  ipv4-family vpn-instance vpnb
    import-route direct
#
return
```

- PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 222:2 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 100:2
    vpn-target 222:2 export-extcommunity
    vpn-target 222:2 111:1 import-extcommunity
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpna
  ip address 10.3.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
```

```
ip binding vpn-instance vpnb
ip address 10.4.1.2 255.255.255.0
#
bgp 100
#
ipv4-family unicast
undo synchronization
#
ipv4-family vpn-instance vpna
import-route direct
#
ipv4-family vpn-instance vpnb
import-route direct
#
return
```

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
ip route-static 10.2.1.0 255.255.255.0 10.1.1.2
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
ip route-static 10.1.1.0 255.255.255.0 10.2.1.2
#
return
```

- CE3 configuration file

```
#
sysname CE3
#
interface GigabitEthernet1/0/0
ip address 10.3.1.1 255.255.255.0
#
ip route-static 10.4.1.0 255.255.255.0 10.3.1.2
#
return
```

- CE4 configuration file

```
#
sysname CE4
#
interface GigabitEthernet1/0/0
ip address 10.4.1.1 255.255.255.0
#
ip route-static 10.3.1.0 255.255.255.0 10.4.1.2
#
return
```

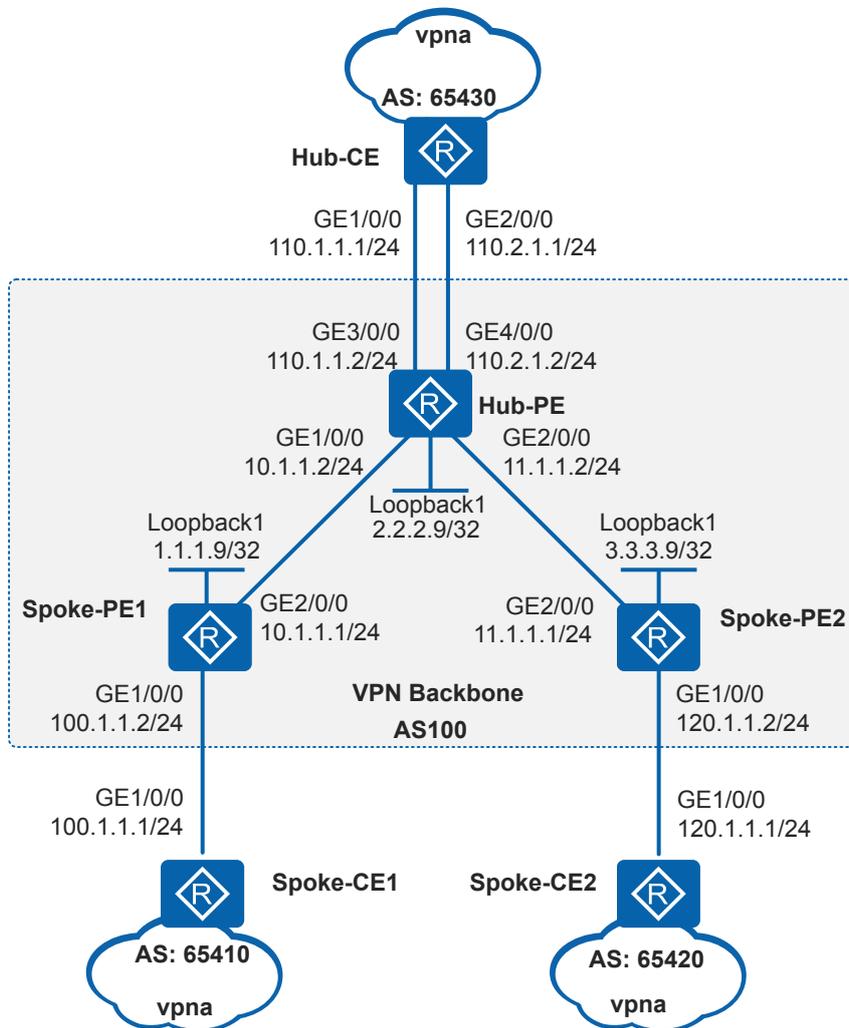
8.9.4 Example for Configuring Hub and Spoke

Networking Requirements

A bank wants to realize secure communication between its headquarters and branches through MPLS VPN. VPN traffic from branches passes the headquarters so that the headquarters can monitor the traffic. The Hub and Spoke networking can meet the bank's needs. As shown in

Figure 8-45, the Spoke-CEs connect to branches, and the Hub-CE connects to the headquarters. All traffic transmitted between Spoke-CEs is forwarded by the Hub-CE.

Figure 8-45 Networking diagram for configuring Hub and Spoke



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IGP protocol on the backbone network to enable the Hub-PE and Spoke-PEs to communicate with each other.
2. Configure basic MPLS capabilities and MPLS LDP on the backbone network to set up LDP LSPs.
3. Set up MP-IBGP peer relationships between the Hub-PE and the Spoke-PEs. The Spoke-PEs do not need to set up an MP-IBGP peer relationship or exchange VPN routing information.
4. Create two VPN instances on the Hub-PE. One is used to receive routes from Spoke-PEs, and the other is used to advertise routes to the Spoke-PEs. Set import target of the first VPN instance to 100:1 and the export target of the other VPN instance to 200:1.

5. Create a VPN instance on the Spoke-PEs. Set the export target of the VPN instance to 100:1 and the import target to 200:1.
6. Configure EBGP on the CEs and PEs to enable them to exchange VPN routing information. Configure Hub-PE to allow Hub-PE to receive the route with the AS repeated for one time.

Procedure

- Step 1** Configure OSPF on the backbone network to enable the Hub-PE and Spoke-PEs to communicate with each other.

Configure Spoke-PE1. The configuration on the Hub-PE and Spoke-PE2 is similar to the configuration on Spoke-PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname Spoke-PE1
[Spoke-PE1] interface loopback 1
[Spoke-PE1-LoopBack1] ip address 1.1.1.9 32
[Spoke-PE1-LoopBack1] quit
[Spoke-PE1] interface gigabitethernet 2/0/0
[Spoke-PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 24
[Spoke-PE1-GigabitEthernet2/0/0] quit
[Spoke-PE1] ospf 1
[Spoke-PE1-ospf-1] area 0
[Spoke-PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[Spoke-PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[Spoke-PE1-ospf-1-area-0.0.0.0] quit
[Spoke-PE1-ospf-1] quit
```

After the configuration is complete, Hub-PE can establish OSPF neighbor relationships with the Spoke-PEs. Run the **display ospf peer** command on the PEs. The command output shows that the status of OSPF neighbor relationships is Full. Run the **display ip routing-table** command. The command output shows that the Hub-PE and the Spoke-PEs have learned the route to the loopback interface of each other.

- Step 2** Configure basic MPLS capabilities and MPLS LDP on the backbone network to set up LDP LSPs.

Configure the Hub-PE. The configuration on the Spoke-PEs is similar to the configuration on the Hub-PE and is not mentioned here.

```
[Hub-PE] mpls lsr-id 2.2.2.9
[Hub-PE] mpls
[Hub-PE-mpls] label advertise non-null
[Hub-PE-mpls] quit
[Hub-PE] mpls ldp
[Hub-PE-mpls-ldp] quit
[Hub-PE] interface gigabitethernet 1/0/0
[Hub-PE-GigabitEthernet1/0/0] mpls
[Hub-PE-GigabitEthernet1/0/0] mpls ldp
[Hub-PE-GigabitEthernet1/0/0] quit
[Hub-PE] interface gigabitethernet 2/0/0
[Hub-PE-GigabitEthernet2/0/0] mpls
[Hub-PE-GigabitEthernet2/0/0] mpls ldp
[Hub-PE-GigabitEthernet2/0/0] quit
```

After the configuration is complete, the Hub-PE can set up LDP peer relationships with the Spoke-PEs. Run the **display mpls ldp session** command on the PEs. In the command output, the state is Operational. Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

- Step 3** Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

 **NOTE**

The import target of the VPN instances on the Hub-PE is the export target of the VPN instance on the Spoke-PEs. The import target and export target on the Hub-PE are different. The import VPN target on the Spoke-PEs is the export VPN target on the Hub-PE.

Configure Spoke-PE1.

```
[Spoke-PE1] ip vpn-instance vpna
[Spoke-PE1-vpn-instance-vpna] ipv4-family
[Spoke-PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[Spoke-PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:1 export-extcommunity
[Spoke-PE1-vpn-instance-vpna-af-ipv4] vpn-target 200:1 import-extcommunity
[Spoke-PE1-vpn-instance-vpna-af-ipv4] quit
[Spoke-PE1-vpn-instance-vpna] quit
[Spoke-PE1] interface gigabitethernet 1/0/0
[Spoke-PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[Spoke-PE1-GigabitEthernet1/0/0] ip address 100.1.1.2 24
[Spoke-PE1-GigabitEthernet1/0/0] quit
```

#Configure Spoke-PE2.

```
[Spoke-PE2] ip vpn-instance vpna
[Spoke-PE2-vpn-instance-vpna] ipv4-family
[Spoke-PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 100:3
[Spoke-PE2-vpn-instance-vpna-af-ipv4] vpn-target 100:1 export-extcommunity
[Spoke-PE2-vpn-instance-vpna-af-ipv4] vpn-target 200:1 import-extcommunity
[Spoke-PE2-vpn-instance-vpna-af-ipv4] quit
[Spoke-PE2-vpn-instance-vpna] quit
[Spoke-PE2] interface gigabitethernet 1/0/0
[Spoke-PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[Spoke-PE2-GigabitEthernet1/0/0] ip address 120.1.1.2 24
[Spoke-PE2-GigabitEthernet1/0/0] quit
```

Configure the Hub-PE.

```
[Hub-PE] ip vpn-instance vpn_in
[Hub-PE-vpn-instance-vpn_in] ipv4-family
[Hub-PE-vpn-instance-vpn_in-af-ipv4] route-distinguisher 100:21
[Hub-PE-vpn-instance-vpn_in-af-ipv4] vpn-target 100:1 import-extcommunity
[Hub-PE-vpn-instance-vpn_in-af-ipv4] quit
[Hub-PE-vpn-instance-vpn_in] quit
[Hub-PE] ip vpn-instance vpn_out
[Hub-PE-vpn-instance-vpn_out] ipv4-family
[Hub-PE-vpn-instance-vpn_out-af-ipv4] route-distinguisher 100:22
[Hub-PE-vpn-instance-vpn_out-af-ipv4] vpn-target 200:1 export-extcommunity
[Hub-PE-vpn-instance-vpn_out-af-ipv4] quit
[Hub-PE-vpn-instance-vpn_out] quit
[Hub-PE] interface gigabitethernet 3/0/0
[Hub-PE-GigabitEthernet3/0/0] ip binding vpn-instance vpn_in
[Hub-PE-GigabitEthernet3/0/0] ip address 110.1.1.2 24
[Hub-PE-GigabitEthernet3/0/0] quit
[Hub-PE] interface gigabitethernet 4/0/0
[Hub-PE-GigabitEthernet4/0/0] ip binding vpn-instance vpn_out
[Hub-PE-GigabitEthernet4/0/0] ip address 110.2.1.2 24
[Hub-PE-GigabitEthernet4/0/0] quit
```

Assign IP addresses to interfaces on CEs according to [Figure 8-45](#).

Configure Spoke-CE1. The configuration on other CEs is similar to the configuration on Spoke-CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname Spoke-CE1
[Spoke-CE1] interface gigabitethernet 1/0/0
[Spoke-CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 24
[Spoke-CE1-GigabitEthernet1/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances. Each PE can ping its connected CE by using the **ping -vpn-instance vpn-name ip-address** command.

 **NOTE**

If a PE has multiple interfaces bound to the same VPN instance, you need to specify the source IP addresses by setting **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping the remote CE. If the source IP address is not specified, the ping operation fails.

Step 4 Set up EBGP peer relationships between the PEs and CEs and import VPN routes into BGP.

 **NOTE**

To accept the routes advertised by Hub-CE, configure the Hub-PE to allow AS number to be repeated once.

Configure Spoke-CE1.

```
[Spoke-CE1] bgp 65410
[Spoke-CE1-bgp] peer 100.1.1.2 as-number 100
[Spoke-CE1-bgp] import-route direct
[Spoke-CE1-bgp] quit
```

Configure Spoke-PE1.

```
[Spoke-PE1] bgp 100
[Spoke-PE1-bgp] ipv4-family vpn-instance vpna
[Spoke-PE1-bgp-vpna] peer 100.1.1.1 as-number 65410
[Spoke-PE1-bgp-vpna] import-route direct
[Spoke-PE1-bgp-vpna] quit
[Spoke-PE1-bgp] quit
```

Configure Spoke-CE2.

```
[Spoke-CE2] bgp 65420
[Spoke-CE2-bgp] peer 120.1.1.2 as-number 100
[Spoke-CE2-bgp] import-route direct
[Spoke-CE2-bgp] quit
```

#Configure Spoke-PE2.

```
[Spoke-PE2] bgp 100
[Spoke-PE2-bgp] ipv4-family vpn-instance vpna
[Spoke-PE2-bgp-vpna] peer 120.1.1.1 as-number 65420
[Spoke-PE2-bgp-vpna] import-route direct
[Spoke-PE2-bgp-vpna] quit
[Spoke-PE2-bgp] quit
```

Configure the Hub-CE.

```
[Hub-CE] bgp 65430
[Hub-CE-bgp] peer 110.1.1.2 as-number 100
[Hub-CE-bgp] peer 110.2.1.2 as-number 100
[Hub-CE-bgp] import-route direct
[Hub-CE-bgp] quit
```

Configure the Hub-PE.

```
[Hub-PE] bgp 100
[Hub-PE-bgp] ipv4-family vpn-instance vpn_in
[Hub-PE-bgp-vpn_in] peer 110.1.1.1 as-number 65430
[Hub-PE-bgp-vpn_in] import-route direct
[Hub-PE-bgp-vpn_in] quit
[Hub-PE-bgp] ipv4-family vpn-instance vpn_out
[Hub-PE-bgp-vpn_out] peer 110.2.1.1 as-number 65430
[Hub-PE-bgp-vpn_out] peer 110.2.1.1 allow-as-loop 1
[Hub-PE-bgp-vpn_out] import-route direct
```

```
[Hub-PE-bgp-vpn_out] quit
[Hub-PE-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and CEs and are in Established state.

Step 5 Set up MP-IBGP peer relationships between the Spoke-PEs and Hub-PE.

NOTE

The Spoke-PEs do not need to allow the repeated AS number, because the router does not check the AS_Path attribute in the routing information advertised by the IBGP peers.

Configure Spoke-PE1.

```
[Spoke-PE1] bgp 100
[Spoke-PE1-bgp] peer 2.2.2.9 as-number 100
[Spoke-PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[Spoke-PE1-bgp] ipv4-family vpnv4
[Spoke-PE1-bgp-af-vpnv4] peer 2.2.2.9 enable
[Spoke-PE1-bgp-af-vpnv4] quit
```

#Configure Spoke-PE2.

```
[Spoke-PE2] bgp 100
[Spoke-PE2-bgp] peer 2.2.2.9 as-number 100
[Spoke-PE2-bgp] peer 2.2.2.9 connect-interface loopback 1
[Spoke-PE2-bgp] ipv4-family vpnv4
[Spoke-PE2-bgp-af-vpnv4] peer 2.2.2.9 enable
[Spoke-PE2-bgp-af-vpnv4] quit
```

Configure the Hub-PE.

```
[Hub-PE] bgp 100
[Hub-PE-bgp] peer 1.1.1.9 as-number 100
[Hub-PE-bgp] peer 1.1.1.9 connect-interface loopback 1
[Hub-PE-bgp] peer 3.3.3.9 as-number 100
[Hub-PE-bgp] peer 3.3.3.9 connect-interface loopback 1
[Hub-PE-bgp] ipv4-family vpnv4
[Hub-PE-bgp-af-vpnv4] peer 1.1.1.9 enable
[Hub-PE-bgp-af-vpnv4] peer 3.3.3.9 enable
[Hub-PE-bgp-af-vpnv4] quit
```

After the configuration is complete, run the **display bgp peer** or **display bgp vpnv4 all peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the Spoke-PEs and the Hub-PE and are in Established state.

Step 6 Verify the configuration.

After the configuration is complete, the Spoke-CEs can ping each other. Run the **tracert** command on the CEs. The command output shows that the traffic between the Spoke-CEs is forwarded through the Hub-CE. You can also deduce the number of forwarding devices between the Spoke-CEs based on the TTL in the ping result.

The information displayed on Spoke-CE1 is used as an example.

```
[Spoke-CE1] ping 120.1.1.1
PING 120.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 120.1.1.1: bytes=56 Sequence=1 ttl=250 time=80 ms
  Reply from 120.1.1.1: bytes=56 Sequence=2 ttl=250 time=129 ms
  Reply from 120.1.1.1: bytes=56 Sequence=3 ttl=250 time=132 ms
  Reply from 120.1.1.1: bytes=56 Sequence=4 ttl=250 time=92 ms
  Reply from 120.1.1.1: bytes=56 Sequence=5 ttl=250 time=126 ms
--- 120.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
```

```

0.00% packet loss
  round-trip min/avg/max = 80/111/132 ms
[Spoke-CE1] tracert 120.1.1.1
traceroute to 120.1.1.1(120.1.1.1), max hops: 30 ,packet length: 40,press CTRL
C to break
 1 100.1.1.2 10 ms  2 ms  1 ms
 2 110.2.1.2 < AS=100 > 10 ms  2 ms  2 ms
 3 110.2.1.1 < AS=100 > 10 ms  2 ms  2 ms
 4 110.1.1.2 < AS=65430 > 10 ms  2 ms  2 ms
 5 120.1.1.2 < AS=100 > 10 ms  2 ms  2 ms
 6 120.1.1.1 < AS=100 > 10 ms  2 ms  5 ms
  
```

Run the **display bgp routing-table** command on the Spoke-CEs. The command output shows the repeated AS number in AS paths of the BGP routes to the remote Spoke-CE.

The information displayed on Spoke-CE1 is used as an example.

```

[Spoke-CE1] display bgp routing-table

BGP Local router ID is 100.1.1.1
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 8
   Network          NextHop      MED        LocPrf    PrefVal  Path/Ogn
*>  100.1.1.0/24     0.0.0.0      0           0          0        ?
   100.1.1.1/32     100.1.1.2    0           0          0        100?
*>  100.1.1.1/32     0.0.0.0      0           0          0        ?
*>  110.1.1.0/24     100.1.1.2    0           0          0        100 65430?
*>  110.2.1.0/24     100.1.1.2    0           0          0        100?
*>  120.1.1.0/24     100.1.1.2    0           0          0        100 65430
100?
*>  127.0.0.0        0.0.0.0      0           0          0        ?
*>  127.0.0.1/32     0.0.0.0      0           0          0        ?
  
```

----End

Configuration Files

- Spoke-CE1 configuration file

```

#
sysname Spoke-CE1
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.1 255.255.255.0
#
bgp 65410
 peer 100.1.1.2 as-number 100
#
ipv4-family unicast
 undo synchronization
 import-route direct
 peer 100.1.1.2 enable
#
return
  
```

- Spoke-PE1 configuration file

```

#
sysname Spoke-PE1
#
ip vpn-instance vpna
 ipv4-family
  route-distinguisher 100:1
  vpn-target 100:1 export-extcommunity
  vpn-target 200:1 import-extcommunity
  
```

```
#
mpls lsr-id 1.1.1.9
mpls
  label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpna
  ip address 100.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
bgp 100
  peer 2.2.2.9 as-number 100
  peer 2.2.2.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 2.2.2.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 2.2.2.9 enable
#
  ipv4-family vpn-instance vpna
    peer 100.1.1.1 as-number 65410
    import-route direct
#
ospf 1
  area 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 1.1.1.9 0.0.0.0
#
return
```

● Spoke-PE2 configuration file

```
#
sysname Spoke-PE2
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:3
    vpn-target 100:1 export-extcommunity
    vpn-target 200:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
  label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpna
  ip address 120.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip address 11.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 3.3.3.9 255.255.255.255
#
```

```
bgp 100
 peer 2.2.2.9 as-number 100
 peer 2.2.2.9 connect-interface LoopBack1
 #
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.9 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 2.2.2.9 enable
 #
 ipv4-family vpn-instance vpna
  peer 120.1.1.1 as-number 65420
  import-route direct
 #
 ospf 1
  area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 11.1.1.0 0.0.0.255
 #
 return
```

- Spoke-CE2 configuration file

```
#
 sysname Spoke-CE2
 #
 interface GigabitEthernet1/0/0
  ip address 120.1.1.1 255.255.255.0
 #
 bgp 65420
  peer 120.1.1.2 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 120.1.1.2 enable
 #
 return
```

- Hub-CE configuration file

```
#
 sysname Hub-CE
 #
 interface GigabitEthernet1/0/0
  ip address 110.1.1.1 255.255.255.0
 #
 interface GigabitEthernet2/0/0
  ip address 110.2.1.1 255.255.255.0
 #
 bgp 65430
  peer 110.1.1.2 as-number 100
  peer 110.2.1.2 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 110.2.1.2 enable
  peer 110.1.1.2 enable
 #
 return
```

- Hub-PE configuration file

```
#
 sysname Hub-PE
 #
 ip vpn-instance vpn_in
  ipv4-family
  route-distinguisher 100:21
  vpn-target 100:1 import-extcommunity
```

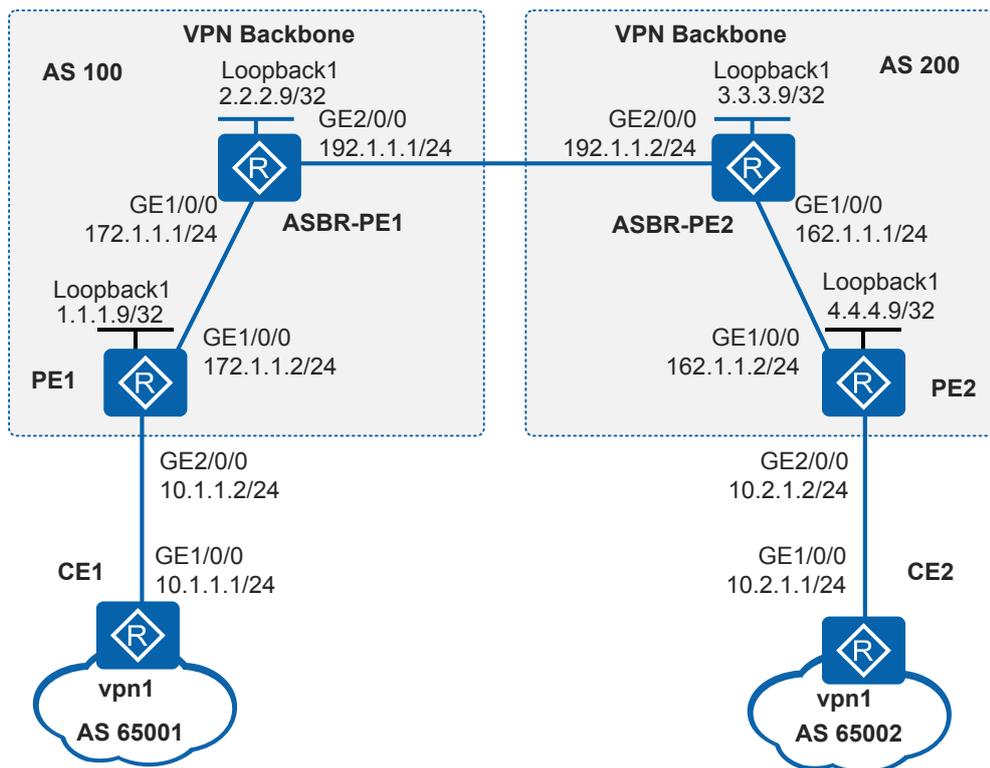
```
#
ip vpn-instance vpn_out
  ipv4-family
    route-distinguisher 100:22
    vpn-target 200:1 export-extcommunity
#
mpls lsr-id 2.2.2.9
mpls
  label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 10.1.1.2 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip address 11.1.1.2 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance vpn_in
  ip address 110.1.1.2 255.255.255.0
#
interface GigabitEthernet4/0/0
  ip binding vpn-instance vpn_out
  ip address 110.2.1.2 255.255.255.0
#
interface LoopBack1
  ip address 2.2.2.9 255.255.255.255
#
bgp 100
  peer 1.1.1.9 as-number 100
  peer 1.1.1.9 connect-interface LoopBack1
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 1.1.1.9 enable
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 1.1.1.9 enable
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpn_in
    peer 110.1.1.1 as-number 65430
    import-route direct
#
  ipv4-family vpn-instance vpn_out
    peer 110.2.1.1 as-number 65430
    peer 110.2.1.1 allow-as-loop
    import-route direct
#
ospf 1
  area 0.0.0.0
    network 2.2.2.9 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 11.1.1.0 0.0.0.255
#
return
```

8.9.5 Example for Configuring Inter-AS VPN Option A

Networking Requirements

The headquarters and branches of a company connect to networks of different carriers. To enable the headquarters and branches to communicate, Inter-AS BGP/MPLS IP VPN needs to be implemented. As shown in **Figure 8-46**, CE1 is located in the headquarters and connects to PE1 in AS 100. CE2 is located at the branch and connects to PE2 in AS 200. Both CE1 and CE2 belong to vpn1.

Figure 8-46 Networking diagram for configuring inter-AS VPN Option A



Configuration Roadmap

Inter-AS Option A can be deployed to meet the company's requirement. The configuration roadmap is as follows:

1. On the MPLS backbone network in AS 100 and AS 200, configure an IGP protocol to enable the PE and ASBR-PEs to communicate with each other.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.
3. Set up an MP-IBGP peer relationship between the PE and ASBR-PEs in each AS to exchange VPN routing information.
4. Create a VPN instance on the PE in each AS and bind the VPN instance to the interface connected to the CE.
5. Set up an EBGP peer relationship between the PEs and CEs in each AS to exchange VPN routing information.

6. Create a VPN instance on each ASBR-PE and bind the instance to the interface connected to the other ASBR-PE (regarding the ASBR-PE as its CE). Set up an EBGP peer relationship between the ASBR-PEs to exchange VPN routing information.

Procedure

- Step 1** Assign IP addresses to interfaces according to [Figure 8-46](#).

Configure PE1. The configuration on PE2, CE1, CE2, ASBR-PE1, and ASBR-PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

- Step 2** On the MPLS backbone network in AS 100 and AS 200, configure OSPF to enable the PEs and the ASBR-PEs to communicate with each other.

Configure PE1. The configuration on PE2 and ASBR-PEs is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

NOTE

The 32-bit loopback interface address used as LSR ID should be advertised by the PEs and ASBR-PEs using OSPF.

After the configuration is complete, the ASBR-PEs and PEs in the same AS can set up an OSPF neighbor relationship. Run the **display ospf peer** command to verify that the status of the neighbor relationship is Full. Run the **display ip routing-table** command. The command output shows that the ASBR and PEs in the same AS have learned the routes to Loopback1 of each other.

- Step 3** Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network of AS 100 and AS 200 to set up LDP LSPs.

Configure basic MPLS capabilities on PE1 and enable LDP on the interface connected to ASBR-PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] label advertise non-null
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE1 and enable LDP on the interface connected to PE1.

```
[ASBR-PE1] mpls lsr-id 2.2.2.9
[ASBR-PE1] mpls
[ASBR-PE1-mpls] label advertise non-null
[ASBR-PE1-mpls] quit
[ASBR-PE1] mpls ldp
[ASBR-PE1-mpls-ldp] quit
[ASBR-PE1] interface gigabitethernet 1/0/0
[ASBR-PE1-GigabitEthernet1/0/0] mpls
[ASBR-PE1-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE2 and enable LDP on the interface connected to PE2.

```
[ASBR-PE2] mpls lsr-id 3.3.3.9
[ASBR-PE2] mpls
[ASBR-PE2-mpls] label advertise non-null
[ASBR-PE2-mpls] quit
[ASBR-PE2] mpls ldp
[ASBR-PE2-mpls-ldp] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] mpls
[ASBR-PE2-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on PE2 and enable LDP on the interface connected to ASBR-PE2.

```
[PE2] mpls lsr-id 4.4.4.9
[PE2] mpls
[PE2-mpls] label advertise non-null
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, the PE and ASBR-PEs in the same AS can set up an LDP peer relationship. Run the **display mpls ldp session** command on the PE and ASBR-PEs to verify that the state is Operational.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session
LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0        Operational DU   Active    0002:23:46 17225/17224
-----
TOTAL: 1 session(s) Found.
```

Step 4 Set up an MP-IBGP peer relationship between the PE and ASBR-PEs in each AS to exchange VPN routing information.

On PE1: set up an MP-IBGP peer relationship with ASBR-PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
```

```
[PE1-bgp] ipv4-family vpnv4  
[PE1-bgp-af-vpnv4] peer 2.2.2.9 enable  
[PE1-bgp-af-vpnv4] quit  
[PE1-bgp] quit
```

On ASBR-PE1: set up an MP-IBGP peer relationship with PE1. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100  
[ASBR-PE1-bgp] peer 1.1.1.9 as-number 100  
[ASBR-PE1-bgp] peer 1.1.1.9 connect-interface loopback 1  
[ASBR-PE1-bgp] ipv4-family vpnv4  
[ASBR-PE1-bgp-af-vpnv4] peer 1.1.1.9 enable  
[ASBR-PE1-bgp-af-vpnv4] quit  
[ASBR-PE1-bgp] quit
```

Step 5 On the PEs, create a VPN instance, enable the IPv4 address family in the instance, and bind the instance to the interfaces connected to CEs.

 **NOTE**

The VPN targets of the VPN instances on the ASBR-PE and PEs in an AS must match. In different ASs, the VPN targets of the PEs do not need to match.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ip vpn-instance vpn1  
[PE1-vpn-instance-vpn1] ipv4-family  
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1  
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both  
[PE1-vpn-instance-vpn1-af-ipv4] quit  
[PE1-vpn-instance-vpn1] quit  
[PE1] interface gigabitethernet 2/0/0  
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1  
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 24  
[PE1-GigabitEthernet2/0/0] quit
```

Step 6 Set up EBGP peer relationships between the PEs and CEs to exchange VPN routing information.

Configure CE1. The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

```
[CE1] interface gigabitethernet 1/0/0  
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24  
[CE1-GigabitEthernet1/0/0] quit  
[CE1] bgp 65001  
[CE1-bgp] peer 10.1.1.2 as-number 100  
[CE1-bgp] import-route direct  
[CE1-bgp] quit
```

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] bgp 100  
[PE1-bgp] ipv4-family vpn-instance vpn1  
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65001  
[PE1-bgp-vpn1] import-route direct  
[PE1-bgp-vpn1] quit  
[PE1-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance vpn-instance-name peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and CEs and are in Established state. Run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that each PE

has set up a BGP peer relationship with the CE and ASBR-PEs in the same AS, and the BGP peer relationships are in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.9
Local AS number : 100

VPN-Instance vpn1, Router ID 1.1.1.9:
Total number of peers : 1                Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
-----
  10.1.1.1      4          65001    5         4       0 00:00:01  Established    3
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
-----
  2.2.2.9       4          100     11        11       0 00:07:09  Established    0

Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 1.1.1.9:
  10.1.1.1      4          65001    5         4       0 00:00:12  Established    3
```

Step 7 Configure Inter-AS VPN Option A.

On ASBR-PE1, create a VPN instance and bind the VPN instance to the interface connected to ASBR-PE2 (ASBR-PE1 considers ASBR-PE2 as its CE).

```
[ASBR-PE1] ip vpn-instance vpn1
[ASBR-PE1-vpn-instance-vpn1] ipv4-family
[ASBR-PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[ASBR-PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[ASBR-PE1-vpn-instance-vpn1-af-ipv4] quit
[ASBR-PE1-vpn-instance-vpn1] quit
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[ASBR-PE1-GigabitEthernet2/0/0] ip address 192.1.1.1 24
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

On ASBR-PE2, create a VPN instance and bind the VPN instance to the interface connected to ASBR-PE1 (ASBR-PE2 considers ASBR-PE1 as its CE).

```
[ASBR-PE2] ip vpn-instance vpn1
[ASBR-PE2-vpn-instance-vpn1] ipv4-family
[ASBR-PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 200:2
[ASBR-PE2-vpn-instance-vpn1-af-ipv4] vpn-target 2:2 both
[ASBR-PE2-vpn-instance-vpn1-af-ipv4] quit
[ASBR-PE2-vpn-instance-vpn1] quit
[ASBR-PE2] interface gigabitethernet 2/0/0
[ASBR-PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[ASBR-PE2-GigabitEthernet2/0/0] ip address 192.1.1.2 24
[ASBR-PE2-GigabitEthernet2/0/0] quit
```

On ASBR-PE1, set up an EBGP peer relationship with ASBR-PE2.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] ipv4-family vpn-instance vpn1
[ASBR-PE1-bgp-vpn1] peer 192.1.1.2 as-number 200
[ASBR-PE1-bgp-vpn1] import-route direct
```

```
[ASBR-PE1-bgp-vpn1] quit
[ASBR-PE1-bgp] quit
```

On ASBR-PE2, set up an EBGP peer relationship with ASBR-PE1.

```
[ASBR-PE2] bgp 200
[ASBR-PE2-bgp] ipv4-family vpn-instance vpn1
[ASBR-PE2-bgp-vpn1] peer 192.1.1.1 as-number 100
[ASBR-PE2-bgp-vpn1] import-route direct
[ASBR-PE2-bgp-vpn1] quit
[ASBR-PE2-bgp] quit
```

Run the **display bgp vpnv4 vpn-instance vpn1 peer** command on the ASBR-PEs. The command output shows that a BGP peer relationship has been established between the ASBR-PEs and is in Established state.

Step 8 Verify the configuration.

After the configuration is complete, CE1 and CE2 learn routes to interfaces on each other and can ping each other successfully.

The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 9          Routes : 9
Destination/Mask    Proto Pre  Cost    Flags NextHop         Interface
 10.1.1.0/24        Direct 0     0        D  10.1.1.1
GigabitEthernet1/0/0
 10.1.1.1/32        Direct 0     0        D  127.0.0.1
GigabitEthernet1/0/0
 10.1.1.255/32      Direct 0     0        D  127.0.0.1
GigabitEthernet1/0/0
 10.2.1.0/24        EBGP   255   0        D  10.1.1.2
GigabitEthernet1/0/0
 127.0.0.0/8        Direct 0     0        D  127.0.0.1      InLoopBack0
 127.0.0.1/32       Direct 0     0        D  127.0.0.1      InLoopBack0
127.255.255.255/32  Direct 0     0        D  127.0.0.1      InLoopBack0
 192.1.1.0/24       EBGP   255   0        D  10.1.1.2
GigabitEthernet1/0/0
255.255.255.255/32  Direct 0     0        D  127.0.0.1      InLoopBack0
[CE1] ping 10.2.1.1
  PING 10.2.1.1: 56 data bytes, press CTRL_C to break
    Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=251 time=119 ms
    Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=251 time=141 ms
    Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=251 time=136 ms
    Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=251 time=113 ms
    Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=251 time=78 ms
  --- 10.2.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 78/117/141 ms
```

Run the **display ip routing-table vpn-instance** command on an ASBR-PE to check the VPN routing table.

```
[ASBR-PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
      Destinations : 6          Routes : 6
Destination/Mask    Proto Pre  Cost    Flags NextHop         Interface
 10.1.1.0/24        IBGP   255   0        RD  1.1.1.9
```

```
GigabitEthernet1/0/0
  10.2.1.0/24 EBGP 255 0 D 192.1.1.2
GigabitEthernet2/0/0
  192.1.1.0/24 Direct 0 0 D 192.1.1.1
GigabitEthernet2/0/0
  192.1.1.1/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
  192.1.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
  255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Run the **display bgp vpnv4 all routing-table** command on an ASBR-PE to check the VPNv4 routes.

```
[ASBR-PE1] display bgp vpnv4 all routing-table

BGP Local router ID is 2.2.2.9
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total number of routes from all PE: 5
Route Distinguisher: 100:1

      Network          NextHop      MED      LocPrf    PrefVal  Path/Ogn
*>i  10.1.1.0/24      1.1.1.9      0        100       0        ?

Route Distinguisher: 100:2

      Network          NextHop      MED      LocPrf    PrefVal  Path/Ogn
*>   10.2.1.0/24      192.1.1.2          0        0        0        200?
*>   192.1.1.0        0.0.0.0          0        0        0        ?
*    192.1.1.1/32    192.1.1.2          0        0        0        200?
*>   192.1.1.1/32    0.0.0.0          0        0        0        ?

VPN-Instance vpn1, Router ID 2.2.2.9:

Total Number of Routes: 5
      Network          NextHop      MED      LocPrf    PrefVal  Path/Ogn
*>i  10.1.1.0/24      1.1.1.9      0        100       0        ?
*>   10.2.1.0/24      192.1.1.2          0        0        0        200?
*>   192.1.1.0        0.0.0.0          0        0        0        ?
*    192.1.1.1/32    192.1.1.2          0        0        0        200?
*>   192.1.1.1/32    0.0.0.0          0        0        0        ?
```

----End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65001
peer 10.1.1.2 as-number 100
#
```

```
ipv4-family unicast
 undo synchronization
 import-route direct
 peer 10.1.1.2 enable
#
return
```

● PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
 label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 10.1.1.2 255.255.255.0
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
bgp 100
 peer 2.2.2.9 as-number 100
 peer 2.2.2.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 2.2.2.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.1.1.1 as-number 65001
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 1.1.1.9 0.0.0.0
  network 172.1.1.0 0.0.0.255
#
return
```

● ASBR-PE1 configuration file

```
#
sysname ASBR-PE1
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:2
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 2.2.2.9
mpls
```

```
    label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 192.1.1.1 255.255.255.0
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 192.1.1.2 as-number 200
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 172.1.1.0 0.0.0.255
#
return
```

● ASBR-PE2 configuration file

```
#
sysname ASBR-PE2
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 200:2
  vpn-target 2:2 export-extcommunity
  vpn-target 2:2 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
 label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 162.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 192.1.1.2 255.255.255.0
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 200
 peer 4.4.4.9 as-number 200
```

```
peer 4.4.4.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 4.4.4.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 4.4.4.9 enable
#
ipv4-family vpn-instance vpn1
peer 192.1.1.1 as-number 100
import-route direct
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 162.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 200:1
vpn-target 2:2 export-extcommunity
vpn-target 2:2 import-extcommunity
#
mpls lsr-id 4.4.4.9
mpls
label advertise non-null
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 162.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
ip address 4.4.4.9 255.255.255.255
#
bgp 200
peer 3.3.3.9 as-number 200
peer 3.3.3.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 3.3.3.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 3.3.3.9 enable
#
ipv4-family vpn-instance vpn1
peer 10.2.1.1 as-number 65002
import-route direct
#
ospf 1
area 0.0.0.0
network 4.4.4.9 0.0.0.0
network 162.1.1.0 0.0.0.255
```

```
#
return

● CE2 configuration file

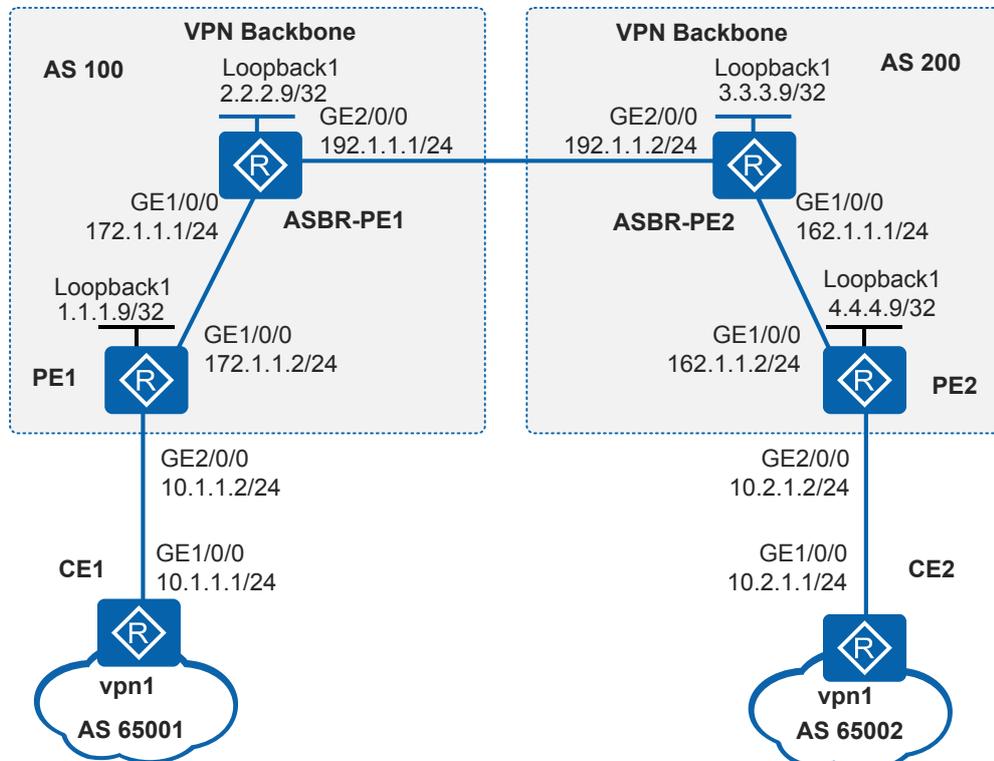
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
bgp 65002
peer 10.2.1.2 as-number 200
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.2.1.2 enable
#
return
```

8.9.6 Example for Configuring Inter-AS VPN Option B

Networking Requirements

The headquarters and branches of a company connect to networks of different carriers. To enable the headquarters and branches to communicate, Inter-AS BGP/MPLS IP VPN needs to be implemented. As shown in **Figure 8-47**, CE1 is located in the headquarters and connects to PE1 in AS 100. CE2 is located at the branch and connects to PE2 in AS 200. Both CE1 and CE2 belong to vpn1.

Figure 8-47 Networking diagram for configuring inter-AS VPN Option B



Configuration Roadmap

Inter-AS Option B can be deployed to meet the company's requirement. The configuration roadmap is as follows:

1. On the MPLS backbone network in AS 100 and AS 200, configure an IGP protocol to enable the PE and ASBR-PEs to communicate with each other.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.
3. Set up an MP-IBGP peer relationship between the PE and ASBR-PEs in each AS to exchange VPN routing information.
4. Create a VPN instance on the PE in each AS and bind the VPN instance to the interface connected to the CE.
5. Set up an EBGP peer relationship between the PEs and CEs between the ASs to exchange VPN routing information.
6. Enable MPLS on the interfaces connecting the ASBRs and set up an MP-EBGP peer relationship between the ASBRs. Configure the ASBRs not to filter received VPNv4 routes based on VPN targets.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-47](#).

Configure PE1. The configuration on PE2, CE1, CE2, ASBR-PE1, and ASBR-PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 On the MPLS backbone network in AS 100 and AS 200, configure OSPF to enable the PEs and the ASBR-PEs to communicate with each other.

Configure PE1. The configuration on PE2 and ASBR-PEs is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

NOTE

The 32-bit loopback interface address used as LSR ID should be advertised by the PEs and ASBR-PEs using OSPF.

After the configuration is complete, the ASBR and PEs in the same AS can set up an OSPF neighbor relationship. Run the **display ospf peer** command to verify that the status of the neighbor relationship is Full. Run the **display ip routing-table** command. The command output shows that the ASBR and PEs in the same AS have learned the routes to Loopback1 of each other.

Step 3 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network of AS 100 and AS 200 to set up LDP LSPs.

Configure basic MPLS capabilities on PE1 and enable LDP on the interface connected to ASBR-PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE1 and enable LDP on the interface connected to PE1.

```
[ASBR-PE1] mpls lsr-id 2.2.2.9
[ASBR-PE1] mpls
[ASBR-PE1-mpls] quit
[ASBR-PE1] mpls ldp
[ASBR-PE1-mpls-ldp] quit
[ASBR-PE1] interface gigabitethernet 1/0/0
[ASBR-PE1-GigabitEthernet1/0/0] mpls
[ASBR-PE1-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE2 and enable LDP on the interface connected to PE2.

```
[ASBR-PE2] mpls lsr-id 3.3.3.9
[ASBR-PE2] mpls
[ASBR-PE2-mpls] quit
[ASBR-PE2] mpls ldp
[ASBR-PE2-mpls-ldp] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] mpls
[ASBR-PE2-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on PE2 and enable LDP on the interface connected to ASBR-PE2.

```
[PE2] mpls lsr-id 4.4.4.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, the PE and ASBR-PEs in the same AS can set up an LDP peer relationship. Run the **display mpls ldp session** command on the PE and ASBR-PEs to verify that the state is Operational.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session
LDP Session(s) in Public Network
```

```
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0        Operational DU   Active  0002:23:46  17225/17224
-----
TOTAL: 1 session(s) Found.
```

Step 4 Set up an MP-IBGP peer relationship between the PE and ASBR-PEs in each AS to exchange VPN routing information.

On PE1: set up an MP-IBGP peer relationship with ASBR-PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

On ASBR-PE1: set up an MP-IBGP peer relationship with PE1. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] peer 1.1.1.9 as-number 100
[ASBR-PE1-bgp] peer 1.1.1.9 connect-interface loopback 1
[ASBR-PE1-bgp] ipv4-family vpnv4
[ASBR-PE1-bgp-af-vpnv4] peer 1.1.1.9 enable
[ASBR-PE1-bgp-af-vpnv4] quit
[ASBR-PE1-bgp] quit
```

Step 5 On the PEs, create a VPN instance, enable the IPv4 address family in the instance, and bind the instance to the interfaces connected to CEs.

 **NOTE**

The VPN targets of the VPN instances on the ASBR-PE and PEs in an AS must match. In different ASs, the VPN targets of the PEs do not need to match.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

Step 6 Set up EBGP peer relationships between the PEs and CEs to exchange VPN routing information.

Configure CE1. The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

```
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] bgp 65001
[CE1-bgp] peer 10.1.1.2 as-number 100
```

```
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1. The configuration on PE2 is similar to the configuration PE1 and is not mentioned here.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65001
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance vpn-instancename peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and CEs and are in Established state. Run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that each PE has set up a BGP peer relationship with the CE and ASBR-PEs in the same AS, and the BGP peer relationships are in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.9
Local AS number : 100

VPN-Instance vpn1, Router ID 1.1.1.9:
Total number of peers : 1                      Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.1.1.1      4          65001    965      967      0 16:00:58
Established  3

[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                      Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  2.2.2.9       4          100     979      974      0 16:08:16
Established  0

  Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 1.1.1.9:
  10.1.1.1      4          65001    966      968      0 16:01:19
Established  3
```

Step 7 Configure Inter-AS VPN Option B.

On ASBR-PE1: Enable MPLS on the interface connected to ASBR-PE2.

```
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] ip address 192.1.1.1 24
[ASBR-PE1-GigabitEthernet2/0/0] mpls
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

On ASBR-PE1: set up the MP-EBGP peer relationship with ASBR-PE2, disable ASBR-PE1 from filtering VPNv4 routes based on VPN targets, and enable next-hop-based label allocation. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] peer 192.1.1.2 as-number 200
[ASBR-PE1-bgp] ipv4-family vpnv4
[ASBR-PE1-bgp-af-vpnv4] peer 192.1.1.2 enable
[ASBR-PE1-bgp-af-vpnv4] undo policy vpn-target
[ASBR-PE1-bgp-af-vpnv4] apply-label per-nexthop
[ASBR-PE1-bgp-af-vpnv4] quit
[ASBR-PE1-bgp] quit
```

Step 8 Verify the configuration.

After the configuration is complete, CE1 and CE2 learn routes to interfaces on each other and can ping each other successfully.

The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 8          Routes : 8
Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
10.1.1.0/24        Direct 0    0        D 10.1.1.1
GigabitEthernet1/0/0
10.1.1.1/32        Direct 0    0        D 127.0.0.1
GigabitEthernet1/0/0
10.1.1.255/32      Direct 0    0        D 127.0.0.1
GigabitEthernet1/0/0
10.2.1.0/24       EBGP   255  0        D 10.1.1.2
GigabitEthernet1/0/0
127.0.0.0/8        Direct 0    0        D 127.0.0.1         InLoopBack0
127.0.0.1/32       Direct 0    0        D 127.0.0.1         InLoopBack0
127.255.255.255/32 Direct 0    0        D 127.0.0.1         InLoopBack0
255.255.255.255/32 Direct 0    0        D 127.0.0.1         InLoopBack0
[CE1] ping 10.2.1.1
  PING 10.2.1.1: 56 data bytes, press CTRL_C to break
    Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=251 time=119 ms
    Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=251 time=141 ms
    Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=251 time=136 ms
    Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=251 time=113 ms
    Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=251 time=78 ms
  --- 10.2.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 78/117/141 ms
```

Run the **display bgp vpnv4 all routing-table** command on an ASBR-PE to check the VPNv4 routes.

```
[ASBR-PE1] display bgp vpnv4 all routing-table

BGP Local router ID is 110.1.1.2
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total number of routes from all PE: 2
Route Distinguisher: 100:1

      Network          NextHop      MED      LocPrf    PrefVal Path/Ogn
-----
*>i 10.1.1.0/24        1.1.1.9      0         100       0        ?
```

```
Route Distinguisher: 200:1
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.2.1.0/24	192.1.1.2			0	200?

----End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65001
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 10.1.1.2 255.255.255.0
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 2.2.2.9 enable
#
ipv4-family vpn-instance vpn1
```

```
peer 10.1.1.1 as-number 65001
import-route direct
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● ASBR-PE1 configuration file

```
#
sysname ASBR-PE1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 172.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 192.1.1.1 255.255.255.0
mpls
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
bgp 100
peer 192.1.1.2 as-number 200
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 192.1.1.2 enable
peer 1.1.1.9 enable
#
ipv4-family vpnv4
undo policy vpn-target
apply-label per-nexthop
peer 1.1.1.9 enable
peer 192.1.1.2 enable
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● ASBR-PE2 configuration file

```
#
sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 162.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 192.1.1.2 255.255.255.0
```

```
mpls
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 200
 peer 192.1.1.1 as-number 100
 peer 4.4.4.9 as-number 200
 peer 4.4.4.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 192.1.1.1 enable
  peer 4.4.4.9 enable
#
 ipv4-family vpnv4
  undo policy vpn-target
  apply-label per-nexthop
  peer 4.4.4.9 enable
  peer 192.1.1.1 enable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 162.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 200:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 162.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
 ip address 4.4.4.9 255.255.255.255
#
bgp 200
 peer 3.3.3.9 as-number 200
 peer 3.3.3.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 3.3.3.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 3.3.3.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.2.1.1 as-number 65002
  import-route direct
```

```
#
ospf 1
 area 0.0.0.0
  network 4.4.4.9 0.0.0.0
  network 162.1.1.0 0.0.0.255
#
return
```

- CE2 configuration file

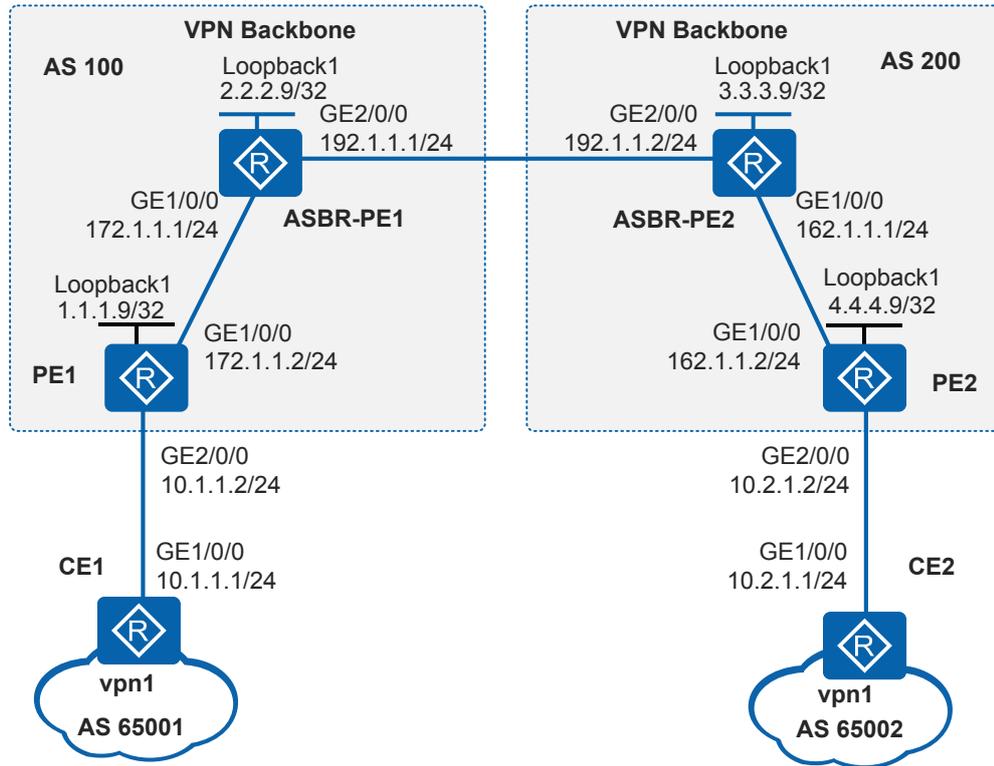
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
bgp 65002
 peer 10.2.1.2 as-number 200
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.2.1.2 enable
#
return
```

8.9.7 Example for Configuring Inter-AS VPN Option C (Solution 1)

Networking Requirements

The headquarters and branches of a company connect to networks of different carriers. To enable the headquarters and branches to communicate, Inter-AS BGP/MPLS IP VPN needs to be implemented. As shown in [Figure 8-48](#), CE1 is located in the headquarters and connects to PE1 in AS 100. CE2 is located at the branch and connects to PE2 in AS 200. Both CE1 and CE2 belong to vpn1.

Figure 8-48 Networking diagram for configuring inter-AS VPN Option C



Configuration Roadmap

Inter-AS Option C can be deployed to meet the company's requirement. The configuration roadmap is as follows:

1. On the MPLS backbone network in AS 100 and AS 200, configure an IGP protocol to enable the PE and ASBR-PEs to communicate with each other.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.
3. Set up an MP-IBGP peer relationship between the PE and ASBR-PEs in each AS to exchange the labeled IPv4 routes.
4. Create a VPN instance on the PE in each AS and bind the VPN instance to the interface connected to the CE.
5. Set up an EBGP peer relationship between the PEs and CEs in each AS to exchange VPN routing information.
6. Enable the capability of exchanging labeled IPv4 routes between the local ASBR-PE and the remote ASBR-PE.
7. Set up an MP-EBGP relationship between PEs in different ASs and set the maximum hops between the PEs.
8. Configure a routing policy on the ASBR-PE: Assign MPLS labels to the routes advertised to the remote ASBR-PE; assign new MPLS labels to the labeled IPv4 routes advertised to the PE in the local AS.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-48](#).

Configure PE1. The configuration on PE2, CE1, CE2, ASBR-PE1, and ASBR-PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 On the MPLS backbone network in AS 100 and AS 200, configure OSPF to enable the PEs and the ASBR-PEs to communicate with each other.

Configure PE1. The configuration on PE2 and ASBR-PEs is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

NOTE

The 32-bit loopback interface address used as LSR ID should be advertised by the PEs and ASBR-PEs using OSPF.

After the configuration is complete, the ASBR and PEs in the same AS can set up an OSPF neighbor relationship. Run the **display ospf peer** command to verify that the status of the neighbor relationship is Full. Run the **display ip routing-table** command. The command output shows that the ASBR and PEs in the same AS have learned the routes to Loopback1 of each other.

Step 3 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network of AS 100 and AS 200 to set up LDP LSPs.

Configure basic MPLS capabilities on PE1 and enable LDP on the interface connected to ASBR-PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE1 and enable LDP on the interface connected to PE1.

```
[ASBR-PE1] mpls lsr-id 2.2.2.9
[ASBR-PE1] mpls
[ASBR-PE1-mpls] quit
[ASBR-PE1] mpls ldp
[ASBR-PE1-mpls-ldp] quit
```

```
[ASBR-PE1] interface gigabitethernet 1/0/0
[ASBR-PE1-GigabitEthernet1/0/0] mpls
[ASBR-PE1-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE2 and enable LDP on the interface connected to PE2.

```
[ASBR-PE2] mpls lsr-id 3.3.3.9
[ASBR-PE2] mpls
[ASBR-PE2-mpls] quit
[ASBR-PE2] mpls ldp
[ASBR-PE2-mpls-ldp] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] mpls
[ASBR-PE2-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on PE2 and enable LDP on the interface connected to ASBR-PE2.

```
[PE2] mpls lsr-id 4.4.4.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, the PE and ASBR-PEs in the same AS can set up an LDP peer relationship. Run the **display mpls ldp session** command on the PE and ASBR-PEs to verify that the state is Operational.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session
LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status          LAM   SsnRole   SsnAge         KASent/Rcv
-----
2.2.2.9:0        Operational    DU    Active    0002:23:46    17225/17224
-----
TOTAL: 1 session(s) Found.
```

Step 4 Set up an MP-IBGP peer relationship between the PE and ASBR-PEs.

On PE1: set up an MP-IBGP peer relationship with ASBR-PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

On ASBR-PE1: set up an MP-IBGP peer relationship with PE1. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] peer 1.1.1.9 as-number 100
```

```
[ASBR-PE1-bgp] peer 1.1.1.9 connect-interface loopback 1
[ASBR-PE1-bgp] ipv4-family vpnv4
[ASBR-PE1-bgp-af-vpnv4] peer 1.1.1.9 enable
[ASBR-PE1-bgp-af-vpnv4] quit
[ASBR-PE1-bgp] quit
```

Step 5 On the PEs, create a VPN instance, enable the IPv4 address family in the instance, and bind the instance to the interfaces connected to CEs.

NOTE

The VPN targets of the VPN instances on the ASBR-PE and PEs in an AS must match. In different ASs, the VPN targets of the PEs do not need to match.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

Step 6 Set up EBGP peer relationships between the PEs and CEs to exchange VPN routing information.

Configure CE1. The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

```
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] bgp 65001
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65001
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance vpn-instancename peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and CEs and are in Established state. Run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that each PE has set up a BGP peer relationship with the CE and ASBR-PEs in the same AS, and the BGP peer relationships are in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.9
Local AS number : 100
```

```

VPN-Instance vpn1, Router ID 1.1.1.9:
Total number of peers : 1                Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.1.1.1      4          65001   1043     1048     0  17:17:21
Established      2
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  2.2.2.9       4          100     59       52       0  00:45:16
Established      0

  Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 1.1.1.9:
  10.1.1.1      4          65001   1045     1050     0  17:19:21
Established      2
  
```

Step 7 Enable the capability of exchanging labeled IPv4 routes.

On PE1: Enable the capability of exchanging labeled IPv4 routes with ASBR-PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```

[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 label-route-capability
[PE1-bgp] quit
  
```

On ASBR-PE1: Enable MPLS on the interface connected to ASBR-PE2. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```

[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] ip address 192.1.1.1 24
[ASBR-PE1-GigabitEthernet2/0/0] mpls
[ASBR-PE1-GigabitEthernet2/0/0] quit
  
```

On ASBR-PE1: Create a routing policy. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```

[ASBR-PE1] route-policy policy1 permit node 1
[ASBR-PE1-route-policy] apply mpls-label
[ASBR-PE1-route-policy] quit
[ASBR-PE1] route-policy policy2 permit node 1
[ASBR-PE1-route-policy] if-match mpls-label
[ASBR-PE1-route-policy] apply mpls-label
[ASBR-PE1-route-policy] quit
  
```

On ASBR-PE1: Apply a routing policy to the routes advertised to PE1, and enable the capability of exchanging labeled IPv4 routes with PE1. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```

[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] peer 1.1.1.9 route-policy policy2 export
[ASBR-PE1-bgp] peer 1.1.1.9 label-route-capability
  
```

On ASBR-PE1: Apply a routing policy to the routes advertised to ASBR-PE2, and enable the capability of exchanging labeled IPv4 routes with ASBR-PE2.

```

[ASBR-PE1-bgp] peer 192.1.1.2 as-number 200
[ASBR-PE1-bgp] peer 192.1.1.2 route-policy policy1 export
[ASBR-PE1-bgp] peer 192.1.1.2 label-route-capability
[ASBR-PE1-bgp] quit
  
```

On ASBR-PE1: Advertise routes to loopback interfaces to ASBR-PE2, and then to PE2. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] network 1.1.1.9 32
[ASBR-PE1-bgp] quit
```

Step 8 Set up an MP-EBGP peer relationship between PE1 and PE2.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 4.4.4.9 as-number 200
[PE1-bgp] peer 4.4.4.9 connect-interface LoopBack 1
[PE1-bgp] peer 4.4.4.9 ebgp-max-hop 10
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 4.4.4.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 200
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface LoopBack 1
[PE2-bgp] peer 1.1.1.9 ebgp-max-hop 10
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

Step 9 Verify the configuration.

After the configuration is complete, CE1 and CE2 learn routes to interfaces on each other and can ping each other successfully.

The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 8          Routes : 8
Destination/Mask    Proto Pre  Cost    Flags NextHop         Interface
-----
10.1.1.0/24        Direct 0     0           D 10.1.1.1
GigabitEthernet1/0/0
10.1.1.1/32        Direct 0     0           D 127.0.0.1
GigabitEthernet1/0/0
10.1.1.255/32      Direct 0     0           D 127.0.0.1
GigabitEthernet1/0/0
10.2.1.0/24        EBGP   255   0           D 10.1.1.2
GigabitEthernet1/0/0
127.0.0.0/8        Direct 0     0           D 127.0.0.1      InLoopBack0
127.0.0.1/32      Direct 0     0           D 127.0.0.1      InLoopBack0
127.255.255.255/32 Direct 0     0           D 127.0.0.1      InLoopBack0
255.255.255.255/32 Direct 0     0           D 127.0.0.1      InLoopBack0

[CE1] ping 10.2.1.1
PING 10.2.1.1: 56 data bytes, press CTRL_C to break
Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=251 time=119 ms
Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=251 time=141 ms
Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=251 time=136 ms
Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=251 time=113 ms
Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=251 time=78 ms
--- 10.2.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
```

```
0.00% packet loss
round-trip min/avg/max = 78/117/141 ms
```

No VPNv4 route exists on ASBR-PEs. Run the **display bgp routing-table label** command on an ASBR-PE to check information about labels of routes.

ASBR-PE1 is used as an example.

```
[ASBR-PE1] display bgp routing-table label

BGP Local router ID is 2.2.2.9
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 2

      Network          NextHop          In/Out Label
* >    1.1.1.9          172.1.1.2        1098/NULL
* >    4.4.4.9          192.1.1.2        1099/1067
```

---End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65001
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 10.1.1.2 255.255.255.0
```

```
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
bgp 100
 peer 2.2.2.9 as-number 100
 peer 2.2.2.9 connect-interface LoopBack1
 peer 4.4.4.9 as-number 200
 peer 4.4.4.9 ebgp-max-hop 10
 peer 4.4.4.9 connect-interface LoopBack1
#
ipv4-family unicast
 undo synchronization
 peer 2.2.2.9 enable
 peer 2.2.2.9 label-route-capability
 peer 4.4.4.9 enable
#
ipv4-family vpnv4
 policy vpn-target
 peer 2.2.2.9 enable
 peer 4.4.4.9 enable
#
ipv4-family vpn-instance vpn1
 peer 10.1.1.1 as-number 65001
 import-route direct
#
ospf 1
 area 0.0.0.0
 network 1.1.1.9 0.0.0.0
 network 172.1.1.0 0.0.0.255
#
return
```

● ASBR-PE1 configuration file

```
#
sysname ASBR-PE1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 192.1.1.1 255.255.255.0
 mpls
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
bgp 100
 peer 192.1.1.2 as-number 200
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
 undo synchronization
 network 1.1.1.9 255.255.255.255
 peer 192.1.1.2 enable
 peer 192.1.1.2 route-policy policy1 export
 peer 192.1.1.2 label-route-capability
 peer 1.1.1.9 enable
 peer 1.1.1.9 route-policy policy2 export
 peer 1.1.1.9 label-route-capability
#
ipv4-family vpnv4
```

```
policy vpn-target
peer 1.1.1.9 enable
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
route-policy policy1 permit node 1
apply mpls-label
route-policy policy2 permit node 1
if-match mpls-label
apply mpls-label
#
return
```

● ASBR-PE2 configuration file

```
#
sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 162.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 192.1.1.2 255.255.255.0
mpls
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
bgp 200
peer 192.1.1.1 as-number 100
peer 4.4.4.9 as-number 200
peer 4.4.4.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
network 4.4.4.9 255.255.255.255
peer 192.1.1.1 enable
peer 192.1.1.1 route-policy policy1 export
peer 192.1.1.1 label-route-capability
peer 4.4.4.9 enable
peer 4.4.4.9 route-policy policy2 export
peer 4.4.4.9 label-route-capability
#
ipv4-family vpnv4
policy vpn-target
peer 4.4.4.9 enable
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 162.1.1.0 0.0.0.255
#
route-policy policy1 permit node 1
apply mpls-label
route-policy policy2 permit node 1
if-match mpls-label
apply mpls-label
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 200:1
    vpn-target 1:1 export-extcommunity
    vpn-target 1:1 import-extcommunity
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 162.1.1.2 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpn1
  ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
  ip address 4.4.4.9 255.255.255.255
#
bgp 200
  peer 1.1.1.9 as-number 100
  peer 1.1.1.9 ebgp-max-hop 10
  peer 1.1.1.9 connect-interface LoopBack1
  peer 3.3.3.9 as-number 200
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 1.1.1.9 enable
    peer 3.3.3.9 enable
    peer 3.3.3.9 label-route-capability
#
  ipv4-family vpnv4
    policy vpn-target
    peer 1.1.1.9 enable
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpn1
    peer 10.2.1.1 as-number 65002
    import-route direct
#
ospf 1
  area 0.0.0.0
    network 4.4.4.9 0.0.0.0
    network 162.1.1.0 0.0.0.255
#
return
```

● CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
  ip address 10.2.1.1 255.255.255.0
#
bgp 65002
  peer 10.2.1.2 as-number 200
#
  ipv4-family unicast
    undo synchronization
    import-route direct
    peer 10.2.1.2 enable
```

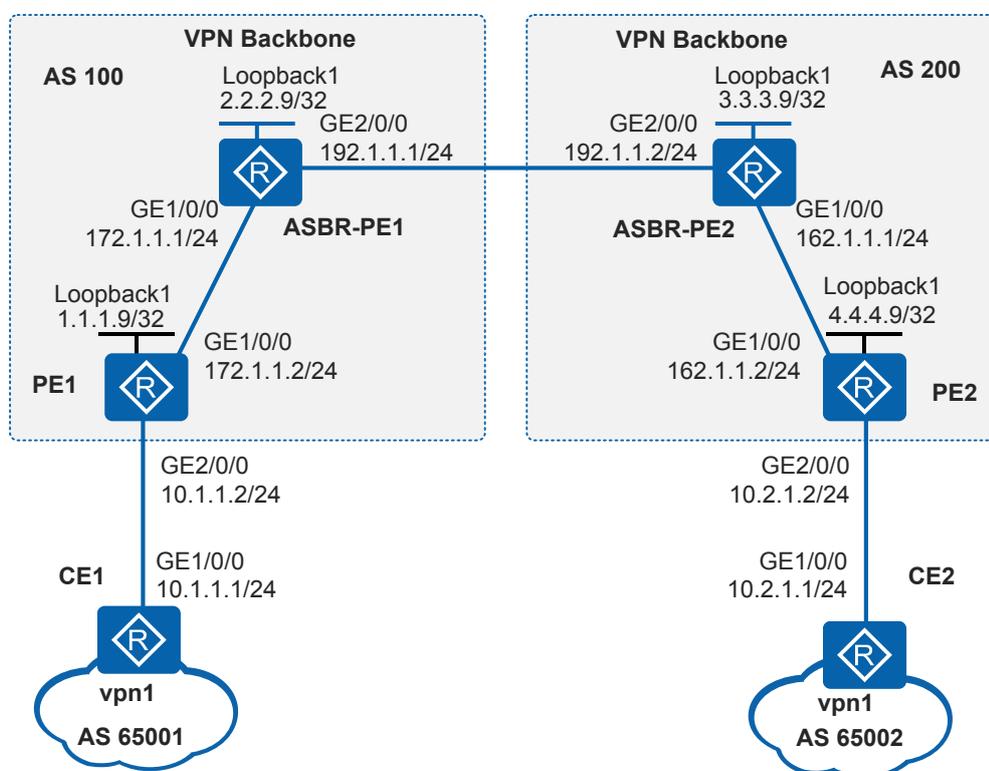
```
#
return
```

8.9.8 Example for Configuring Inter-AS VPN Option C (Solution 2)

Networking Requirements

The headquarters and branches of a company connect to networks of different carriers. To enable the headquarters and branches to communicate, Inter-AS BGP/MPLS IP VPN needs to be implemented. As shown in **Figure 8-49**, CE1 is located in the headquarters and connects to PE1 in AS 100. CE2 is located at the branch and connects to PE2 in AS 200. Both CE1 and CE2 belong to vpn1.

Figure 8-49 Networking diagram for configuring Inter-AS VPN Option C



No IBGP peer relationship is required between the PE and ASBR-PEs. The ASBR-PE learns the labeled BGP routes of the public network at the remote AS from the remote ASBR-PE. Then these BGP routes are imported to IGP. In this manner, LDP can distribute labels for these routes and establish an inter-AS LDP LSP. The inter-AS BGP/MPLS IP VPN Option C can then be implemented.

Configuration Roadmap

Inter-AS Option C can be deployed to meet the company's requirement. The configuration roadmap is as follows:

1. On the MPLS backbone network in AS 100 and AS 200, configure an IGP protocol to enable the PE and ASBR-PEs to communicate with each other.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.
3. Create a VPN instance on the PE in each AS and bind the VPN instance to the interface connected to the CE.
4. Set up an EBGP peer relationship between the PEs and CEs in each AS to exchange VPN routing information.
5. Advertise routes of the PE in an AS to the remote PE: First on the local ASBR-PE, advertise the routes of the PE in an AS to the remote ASBR-PE through BGP; then on the remote ASBR-PE, import these BGP routes to IGP. Then the remote PE learns routes of the PE in the local AS through IGP.
6. Configure a routing policy on the ASBR-PE: Assign MPLS labels to the routes advertised to the remote ASBR-PE.
7. Enable the capability of exchanging labeled IPv4 routes between the local ASBR-PE and the remote ASBR-PE.
8. Configure LDP LSPs for the labeled BGP routes of the public network on ASBR-PEs.
9. Set up MP-EBGP peer relationships between PEs of different ASs. In most cases, these PEs are not directly connected, and the maximum hops between them must be specified.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-49](#).

Configure PE1. The configuration on PE2, CE1, CE2, ASBR-PE1, and ASBR-PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 On the MPLS backbone network in AS 100 and AS 200, configure OSPF to enable the PEs and the ASBR-PEs to communicate with each other.

Configure PE1. The configuration on PE2 and ASBR-PEs is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

NOTE

The 32-bit loopback interface address used as LSR ID should be advertised by the PEs and ASBR-PEs using OSPF.

After the configuration is complete, the ASBR and PEs in the same AS can set up an OSPF neighbor relationship. Run the **display ospf peer** command to verify that the status of the neighbor relationship is Full. Run the **display ip routing-table** command. The command

output shows that the ASBR and PEs in the same AS have learned the routes to Loopback1 of each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ospf peer

      OSPF Process 1 with Router ID 1.1.1.9
      Neighbors

Area 0.0.0.0 interface 172.1.1.2(GigabitEthernet1/0/0)'s neighbors
Router ID: 2.2.2.9           Address: 172.1.1.1
  State: Full  Mode:Nbr is Master  Priority: 1
  DR: 172.1.1.2  BDR: 172.1.1.1  MTU: 0
  Dead timer due in 34 sec
  Retrans timer interval: 5
  Neighbor is up for 18:50:53
  Authentication Sequence: [ 0 ]
```

The ASBR-PE and PEs in the same AS have obtained the IP address of Loopback1 interface of each other and can ping Loopback1 interface address of each other.

Step 3 Set up the EBGP peer relationship between ASBR-PEs.

Configure ASBR-PE1.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] peer 192.1.1.2 as-number 200
[ASBR-PE1-bgp] quit
```

Configure ASBR-PE2.

```
[ASBR-PE2] bgp 200
[ASBR-PE2-bgp] peer 192.1.1.1 as-number 100
[ASBR-PE2-bgp] quit
```

After the configuration is complete, run the **display bgp peer** command on ASBR-PEs. The command output shows that the statue of neighbors is **Established**.

ASBR-PE1 is used as an example.

```
[ASBR-PE1] display bgp peer

BGP local router ID : 2.2.2.9
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V  AS      MsgRcvd  MsgSent  OutQ  Up/Down      State          PrefRcv
-----
192.1.1.2    4  200      129      134      0  01:39:21  Established          1
```

Step 4 Advertise the routes of a PE in an AS to the remote PE.

On ASBR-PE1: Advertise routes to loopback interfaces to ASBR-PE2.

```
[ASBR-PE1] bgp 100
[ASBR-PE1-bgp] network 1.1.1.9 32
[ASBR-PE1-bgp] quit
```

On ASBR-PE2: Advertise routes to loopback interfaces to ASBR-PE1.

```
[ASBR-PE2] bgp 200
[ASBR-PE2-bgp] network 4.4.4.9 32
[ASBR-PE2-bgp] quit
```

On ASBR-PE1: Import BGP routes to OSPF, and advertise the routes of PE2 to PE1 according to OSPF.

```
[ASBR-PE1] ospf 1
[ASBR-PE1-ospf-1] import-route bgp
```

On ASBR-PE2: Import BGP routes to OSPF, and advertise the routes of PE1 to PE2 according to OSPF.

```
[ASBR-PE2] ospf 1
[ASBR-PE2-ospf-1] import-route bgp
```

After the configuration is complete, run the **display ip routing-table** command on PEs to check the routing table. PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 10          Routes : 10

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
-----
      1.1.1.9/32     Direct  0    0           D  127.0.0.1         LoopBack1
      2.2.2.9/32     OSPF   10    1           D  172.1.1.1
GigabitEthernet1/0/0
      4.4.4.9/32     O_ASE  150   1           D  172.1.1.1         GigabitEthernet1/0/0
      127.0.0.0/8     Direct  0    0           D  127.0.0.1         InLoopBack0
      127.0.0.1/32   Direct  0    0           D  127.0.0.1         InLoopBack0
127.255.255.255/32  Direct  0    0           D  127.0.0.1         InLoopBack0
      172.1.1.0/24   Direct  0    0           D  172.1.1.2
GigabitEthernet1/0/0
      172.1.1.2/32   Direct  0    0           D  127.0.0.1
GigabitEthernet1/0/0
      172.1.1.255/32 Direct  0    0           D  127.0.0.1
GigabitEthernet1/0/0
255.255.255.255/32 Direct  0    0           D  127.0.0.1         InLoopBack0
```

Step 5 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network of AS 100 and AS 200 to set up LDP LSPs.

Configure basic MPLS capabilities on PE1 and enable LDP on the interface connected to ASBR-PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE1 and enable LDP on the interface connected to PE1.

```
[ASBR-PE1] mpls lsr-id 2.2.2.9
[ASBR-PE1] mpls
[ASBR-PE1-mpls] quit
[ASBR-PE1] mpls ldp
[ASBR-PE1-mpls-ldp] quit
[ASBR-PE1] interface gigabitethernet 1/0/0
[ASBR-PE1-GigabitEthernet1/0/0] mpls
[ASBR-PE1-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE1-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on ASBR-PE2 and enable LDP on the interface connected to PE2.

```
[ASBR-PE2] mpls lsr-id 3.3.3.9
[ASBR-PE2] mpls
[ASBR-PE2-mpls] quit
[ASBR-PE2] mpls ldp
[ASBR-PE2-mpls-ldp] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] mpls
[ASBR-PE2-GigabitEthernet1/0/0] mpls ldp
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure basic MPLS capabilities on PE2 and enable LDP on the interface connected to ASBR-PE2.

```
[PE2] mpls lsr-id 4.4.4.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, the LDP sessions between PE1 and the ASBR-PE1, and between PE2 and ASBR-PE2 are set up. Run the **display mpls ldp session** command. The command output shows that the status is "Operational". Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0        Operational DU   Passive  0000:00:01  5/5
-----
TOTAL: 1 session(s) Found.

[PE1] display mpls ldp lsp
LDP LSP Information
-----
DestAddress/Mask  In/OutLabel  UpstreamPeer  NextHop      OutInterface
-----
1.1.1.9/32        3/NULL       2.2.2.9        127.0.0.1    InLoop0
*1.1.1.9/32       Liberal/1024
2.2.2.9/32        NULL/3        -              172.1.1.1    GE1/0/0
2.2.2.9/32        1024/3       2.2.2.9        172.1.1.1    GE1/0/0
-----
TOTAL: 3 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
```

Step 6 Enable the capability of exchanging labeled IPv4 routes on ASBR-PEs.

On ASBR-PE1: Enable MPLS on the interface connected to ASBR-PE2. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] ip address 192.1.1.1 24
```

```
[ASBR-PE1-GigabitEthernet2/0/0] mpls  
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

On ASBR-PE1: Create a routing policy. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] route-policy policy1 permit node 1  
[ASBR-PE1-route-policy] apply mpls-label  
[ASBR-PE1-route-policy] quit
```

On ASBR-PE1: Apply a routing policy to the routes advertised to ASBR-PE2, and enable the capability of exchanging labeled IPv4 routes with ASBR-PE2. The configuration on ASBR-PE2 is similar to the configuration on ASBR-PE1 and is not mentioned here.

```
[ASBR-PE1] bgp 100  
[ASBR-PE1-bgp] peer 192.1.1.2 route-policy policy1 export  
[ASBR-PE1-bgp] peer 192.1.1.2 label-route-capability  
[ASBR-PE1-bgp] quit
```

Step 7 Configure LDP LSPs for the labeled BGP routes of the public network on ASBR devices.

Configure ASBR-PE1.

```
[ASBR-PE1] mpls  
[ASBR-PE1-mpls] lsp-trigger bgp-label-route  
[ASBR-PE1-mpls] quit
```

Configure ASBR-PE2.

```
[ASBR-PE2] mpls  
[ASBR-PE2-mpls] lsp-trigger bgp-label-route  
[ASBR-PE2-mpls] quit
```

Step 8 Configure VPN instances to access CEs on PEs.

Configure PE1.

```
[PE1] ip vpn-instance vpn1  
[PE1-vpn-instance-vpn1] ipv4-family  
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1  
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 export-extcommunity  
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 import-extcommunity  
[PE1-vpn-instance-vpn1-af-ipv4] quit  
[PE1-vpn-instance-vpn1] quit  
[PE1] interface gigabitethernet 2/0/0  
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1  
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 24  
[PE1-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpn1  
[PE2-vpn-instance-vpn1] ipv4-family  
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 200:1  
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 export-extcommunity  
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 import-extcommunity  
[PE2-vpn-instance-vpn1-af-ipv4] quit  
[PE2-vpn-instance-vpn1] quit  
[PE2] interface gigabitethernet 2/0/0  
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1  
[PE2-GigabitEthernet2/0/0] ip address 10.2.1.2 24  
[PE2-GigabitEthernet2/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances. Each PE can ping its connected CE.

The information displayed on PE1 and CE1 is used as an example.

```
[PE1] display ip vpn-instance verbose
Total VPN-Instances configured : 1
Total IPv4 VPN-Instances configured : 1
Total IPv6 VPN-Instances configured : 0

VPN-Instance Name and ID : vpn1, 1
  Interfaces : GigabitEthernet2/0/0
Address family ipv4
Create date : 2008/02/27 09:53:47
Up time : 0 days, 00 hours, 35 minutes and 43 seconds
Route Distinguisher : 100:1
Export VPN Targets : 1:1
Import VPN Targets : 1:1
Label Policy : label per route
Log Interval : 5
[PE1] ping -vpn-instance vpn1 10.1.1.1
PING 10.1.1.1: 56 data bytes, press CTRL_C to break
Request time out
Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=50 ms
Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=40 ms
Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=30 ms
Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=10 ms

--- 10.1.1.1 ping statistics ---
 5 packet(s) transmitted
 4 packet(s) received
20.00% packet loss
round-trip min/avg/max = 10/32/50 ms
```

Step 9 Set up an MP-EBGP peer relationship between PE1 and PE2.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 4.4.4.9 as-number 200
[PE1-bgp] peer 4.4.4.9 connect-interface LoopBack 1
[PE1-bgp] peer 4.4.4.9 ebgp-max-hop 10
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 4.4.4.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 200
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface LoopBack 1
[PE2-bgp] peer 1.1.1.9 ebgp-max-hop 10
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

Step 10 Set up EBGP peer relationships between the PEs and CEs and import VPN routes into BGP.

Configure CE1.

```
[CE1] bgp 65001
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65002
[CE2-bgp] peer 10.2.1.2 as-number 200
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65001
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 200
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 as-number 65002
[PE2-bgp-vpn1] import-route direct
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance peer** command on the PEs. The command output shows that BGP peer relationships have been established between the PEs and CEs.

The peer relationship between PE1 and CE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.9
Local AS number : 100

VPN-Instance vpn1, router ID 1.1.1.9:
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
10.1.1.1      4 65001    3         3     0 00:00:52  Established    1
```

Step 11 Verify the configuration.

After the configuration is complete, CE1 and CE2 learn routes to interfaces on each other and can ping each other successfully.

The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
        Destinations : 8          Routes : 8
Destination/Mask    Proto    Pre  Cost    Flags NextHop          Interface
10.1.1.0/24         Direct   0    0        D   10.1.1.1
GigabitEthernet1/0/0
10.1.1.1/32         Direct   0    0        D   127.0.0.1
GigabitEthernet1/0/0
10.1.1.255/32       Direct   0    0        D   127.0.0.1
GigabitEthernet1/0/0
10.2.1.0/24         EBGP     255  0        D   10.1.1.2
GigabitEthernet1/0/0
127.0.0.0/8         Direct   0    0        D   127.0.0.1      InLoopBack0
127.0.0.1/32       Direct   0    0        D   127.0.0.1      InLoopBack0
127.255.255.255/32 Direct   0    0        D   127.0.0.1      InLoopBack0
255.255.255.255/32 Direct   0    0        D   127.0.0.1      InLoopBack0
[CE1] ping 10.2.1.1
PING 10.2.1.1: 56 data bytes, press CTRL_C to break
Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=251 time=102 ms
Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=251 time=89 ms
Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=251 time=106 ms
Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=251 time=104 ms
Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=251 time=56 ms

--- 10.2.1.1 ping statistics ---
 5 packet(s) transmitted
```

```
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 56/91/106 ms
```

After the configuration is complete, run the **display ip routing-table dest-ip-address verbose** command on ASBR-PE1. The command output shows that the routes from ASBR-PE1 to PE2 are labeled BGP routes of the public network: routing table is "Public", the protocol type is "BGP", and the label has a non-zero value.

The information displayed on ASBR-PE1 is used as an example.

```
[ASBR-PE1] display ip routing-table 4.4.4.9 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1

Destination      : 4.4.4.9/32
Protocol         : BGP                Process ID      : 0
Preference      : 255                Cost           : 1
NextHop         : 192.1.1.2           Neighbour      : 192.1.1.2
State           : Active Adv         Age            : 00h12m53s
Tag             : 0                  Priority        : 0
Label           : 15360              QoSInfo        : 0x0
IndirectID      : 0x0
RelayNextHop    : 192.1.1.2           Interface      : GigabitEthernet2/0/0
TunnelID        : 0x6002006          Flags           : D
```

Run the **display mpls lsp protocol ldp include dest-ip-address verbose** on ASBR-PE1 and PE2 respectively. The command output shows that an LDP LSP is established between ASBR-PE1 and PE2. Besides, you can find an LDP Ingress LSP on a PE to the remote PE.

```
[ASBR-PE1] display mpls lsp protocol ldp include 4.4.4.9 32 verbose
-----
LSP Information: LDP LSP
-----
No                : 1
VrfIndex          :
Fec               : 4.4.4.9/32
NextHop           : 192.1.1.2
In-Label          : 1024
Out-Label         : NULL
In-Interface      : -----
Out-Interface     : -----
LspIndex          : 13313
Token             : 0x0
FrrToken          : 0x0
LsrType           : Egress
Outgoing token    : 0x6002006
Label Operation   : SWAPPUSH
Mpls-Mtu          : -----
TimeStamp         : 15829sec
Bfd-State         : ---
BGPPKey           : 0x24
```

---End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
```

```
bgp 65001
 peer 10.1.1.2 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.1.1.2 enable
 #
return
```

● PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
 #
 mpls lsr-id 1.1.1.9
 mpls
 #
 mpls ldp
 #
 interface GigabitEthernet1/0/0
  ip address 172.1.1.2 255.255.255.0
  mpls
  mpls ldp
 #
 interface GigabitEthernet2/0/0
  ip binding vpn-instance vpn1
  ip address 10.1.1.2 255.255.255.0
 #
 interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
 #
 bgp 100
  peer 4.4.4.9 as-number 200
  peer 4.4.4.9 ebgp-max-hop 10
  peer 4.4.4.9 connect-interface LoopBack1
 #
  ipv4-family unicast
  undo synchronization
  peer 4.4.4.9 enable
 #
  ipv4-family vpnv4
  policy vpn-target
  peer 4.4.4.9 enable
 #
  ipv4-family vpn-instance vpn1
  import-route direct
  peer 10.1.1.1 as-number 65001
 #
 ospf 1
  area 0.0.0.0
  network 1.1.1.9 0.0.0.0
  network 172.1.1.0 0.0.0.255
 #
return
```

● ASBR-PE1 configuration file

```
#
sysname ASBR-PE1
#
 mpls lsr-id 2.2.2.9
 mpls
  lsp-trigger bgp-label-route
 #
 mpls ldp
```

```
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 192.1.1.1 255.255.255.0
 mpls
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
bgp 100
 peer 192.1.1.2 as-number 200
#
 ipv4-family unicast
  undo synchronization
  network 1.1.1.9 255.255.255.255
  peer 192.1.1.2 enable
  peer 192.1.1.2 route-policy policy1 export
  peer 192.1.1.2 label-route-capability
#
ospf 1
 import-route bgp
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 172.1.1.0 0.0.0.255
#
route-policy policy1 permit node 1
 apply mpls-label
#
return
```

● ASBR-PE2 configuration file

```
#
sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
 lsp-trigger bgp-label-route
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 162.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 192.1.1.2 255.255.255.0
 mpls
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 200
 peer 192.1.1.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  network 4.4.4.9 255.255.255.255
  peer 192.1.1.1 enable
  peer 192.1.1.1 route-policy policy1 export
  peer 192.1.1.1 label-route-capability
#
ospf 1
 import-route bgp
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
```

```
network 162.1.1.0 0.0.0.255
#
route-policy policy1 permit node 1
apply mpls-label
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 200:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 162.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
ip address 4.4.4.9 255.255.255.255
#
bgp 200
peer 1.1.1.9 as-number 100
peer 1.1.1.9 ebgp-max-hop 10
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpn1
import-route direct
peer 10.2.1.1 as-number 65002
#
ospf 1
area 0.0.0.0
network 4.4.4.9 0.0.0.0
network 162.1.1.0 0.0.0.255
#
return
```

● CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
bgp 65002
peer 10.2.1.2 as-number 200
#
ipv4-family unicast
undo synchronization
```

```
import-route direct
peer 10.2.1.2 enable
#
return
```

8.9.9 Example for Configuring MCE

Networking Requirements

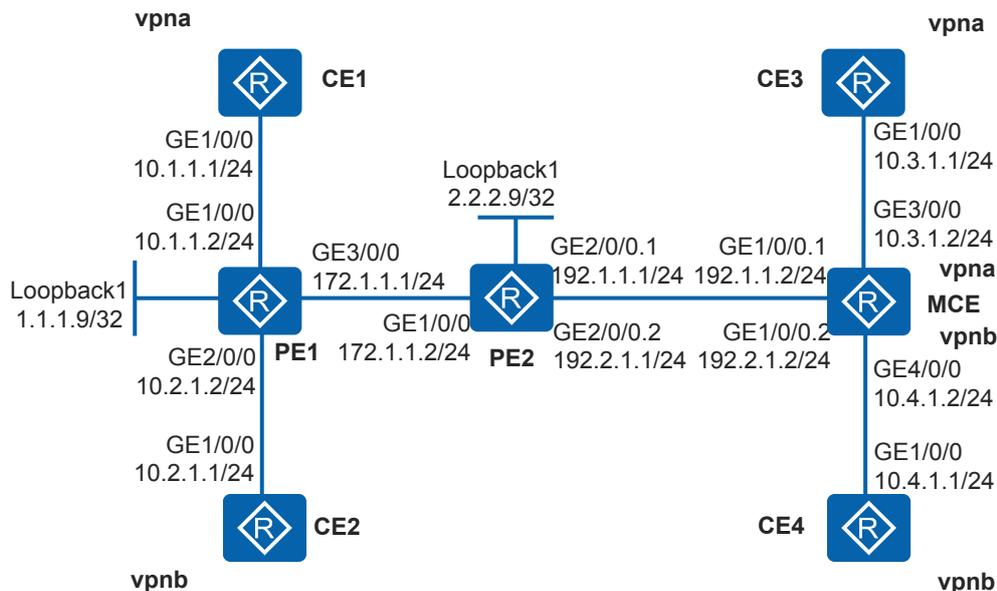
The headquarters and branches of a company need to communicate through MPLS VPN, and two services of the company must be isolated. To reduce hardware costs, the company wants the branches to connect to the PE through one CE.

As shown in [Figure 8-50](#), the networking requirements are as follows:

- CE1 and CE2 connect to the headquarters. CE1 belongs to vpna, and CE2 belongs to vpnb.
- The multi-VPN-instance CE (MCE) device connects to vpna and vpnb of the branches through CE3 and CE4.

Users in the same VPN need to communicate with each other, but users in different VPNs must be isolated.

Figure 8-50 Networking diagram for configuring MCE



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between PEs to implement interworking between them and configure MP-IBGP to exchange VPN routing information.
2. Configure basic MPLS capabilities and MPLS LDP on the PEs to set up LDP LSPs.
3. Create VPN instances vpna and vpnb on the MCEs and PEs to isolate services.

4. Set up EBGP peer relationships between PE1 and its connected CEs, and import BGP routes to the VPN routing table on PE1.
5. Configure routing between the MCE and VPN sites and between the MCE and PE2.

Procedure

Step 1 Configure OSPF on PEs of the backbone network.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip address 172.1.1.1 24
[PE1-GigabitEthernet3/0/0] quit
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configuration is complete, PEs can learn Loopback1 address of each other.

The information displayed on PE2 is used as an example.

```
[PE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
  Destinations : 9          Routes : 9
  Destination/Mask  Proto  Pre  Cost    Flags  NextHop         Interface
  1.1.1.9/32       OSPF   10   1        D  172.1.1.1
GigabitEthernet1/0/0
  2.2.2.9/32       Direct 0    0        D  127.0.0.1       LoopBack1
  127.0.0.0/8      Direct 0    0        D  127.0.0.1       InLoopBack0
  127.0.0.1/32     Direct 0    0        D  127.0.0.1       InLoopBack0
127.255.255.255/32 Direct 0    0        D  127.0.0.1       InLoopBack0
  172.1.1.0/24     Direct 0    0        D  172.1.1.2
GigabitEthernet1/0/0
  172.1.1.2/32     Direct 0    0        D  127.0.0.1
GigabitEthernet1/0/0
  172.1.1.255/32   Direct 0    0        D  127.0.0.1
GigabitEthernet1/0/0
255.255.255.255/32 Direct 0    0        D  127.0.0.1       InLoopBack0
```

Step 2 Configure basic MPLS capabilities and MPLS LDP on the PEs to set up LDP LSPs.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mps] quit
[PE1] mpls ldp
[PE1-mps-ldp] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls
[PE1-GigabitEthernet3/0/0] mpls ldp
[PE1-GigabitEthernet3/0/0] quit
```

After the configuration is complete, run the **display mpls ldp session** command on the PEs. The command output shows that the MPLS LDP session between the PEs is in Operational state.

The information displayed on PE2 is used as an example.

```
[PE2] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
1.1.1.9:0        Operational DU   Active  0000:00:04  17/17
-----
TOTAL: 1 session(s) Found.
```

Step 3 Configure VPN instances on the PEs. On PE1, bind the VPN instances to the interfaces connected to CE1 and CE2 respectively. On PE2, bind the VPN instances to the interfaces connected to the MCE.

Configure PE1.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpb
[PE1-GigabitEthernet2/0/0] ip address 10.2.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 200:1
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv4-family
[PE2-vpn-instance-vpb-af-ipv4] route-distinguisher 200:2
[PE2-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE2-vpn-instance-vpb-af-ipv4] quit
[PE2-vpn-instance-vpb] quit
[PE2] interface gigabitethernet 2/0/0.1
[PE2-GigabitEthernet2/0/0.1] dot1q termination vid 10
[PE2-GigabitEthernet2/0/0.1] ip binding vpn-instance vpna
[PE2-GigabitEthernet2/0/0.1] ip address 192.1.1.1 24
[PE2-GigabitEthernet2/0/0.1] quit
[PE2] interface gigabitethernet 2/0/0.2
[PE2-GigabitEthernet2/0/0.2] dot1q termination vid 20
[PE2-GigabitEthernet2/0/0.2] ip binding vpn-instance vpb
[PE2-GigabitEthernet2/0/0.2] ip address 192.2.1.1 24
[PE2-GigabitEthernet2/0/0.2] quit
```

Step 4 Configure VPN instances on the MCE, and bind the VPN instances to the interfaces connected to CE3, CE4, and PE2.

```
<Huawei> system-view
[Huawei] sysname MCE
[MCE] ip vpn-instance vpna
[MCE-vpn-instance-vpna] ipv4-family
[MCE-vpn-instance-vpna-af-ipv4] route-distinguisher 300:1
[MCE-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[MCE-vpn-instance-vpna-af-ipv4] quit
[MCE-vpn-instance-vpna] quit
[MCE] ip vpn-instance vpb
[MCE-vpn-instance-vpb] ipv4-family
[MCE-vpn-instance-vpb-af-ipv4] route-distinguisher 300:2
[MCE-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[MCE-vpn-instance-vpb-af-ipv4] quit
[MCE-vpn-instance-vpb] quit
[MCE] interface gigabitEthernet 3/0/0
[MCE-GigabitEthernet3/0/0] ip binding vpn-instance vpna
[MCE-GigabitEthernet3/0/0] ip address 10.3.1.2 24
[MCE-GigabitEthernet3/0/0] quit
[MCE] interface gigabitEthernet 4/0/0
[MCE-GigabitEthernet4/0/0] ip binding vpn-instance vpb
[MCE-GigabitEthernet4/0/0] ip address 10.4.1.2 24
[MCE-GigabitEthernet4/0/0] quit
[MCE] interface gigabitEthernet 1/0/0.1
[MCE-GigabitEthernet1/0/0.1] dot1q termination vid 10
[MCE-GigabitEthernet1/0/0.1] ip binding vpn-instance vpna
[MCE-GigabitEthernet1/0/0.1] ip address 192.1.1.2 24
[MCE-GigabitEthernet1/0/0.1] quit
[MCE] interface gigabitEthernet 1/0/0.2
[MCE-GigabitEthernet1/0/0.2] dot1q termination vid 20
[MCE-GigabitEthernet1/0/0.2] ip binding vpn-instance vpb
[MCE-GigabitEthernet1/0/0.2] ip address 192.2.1.2 24
[MCE-GigabitEthernet1/0/0.2] quit
```

Step 5 Set up an MP-IBGP peer relationship between PEs. Set up EBGP peer relationships between PE1 and CE1, and between PE1 and CE2.

Configure CE1. The configuration on other PE1 and PE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] ipv4-family unicast
[CE1-bgp-af-ipv4] import-route direct
[CE1-bgp-af-ipv4] quit
[CE1-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on PE1. The command output shows that the PE1 has set up an IBGP peer relationship with PE2 and EBGP peer relationships with CE1 and CE2. The peer relationships are in Established state.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
-----
2.2.2.9      4   100     288     287     0 01:19:16  Established    4

Peer of IPv4-family for vpn instance :

VPN-Instance vpna, router ID 1.1.1.9:
10.1.1.1     4 65410      9      11     0 00:04:14  Established    4
```

```
VPN-Instance vpnb, router ID 1.1.1.9:
10.2.1.1      4 65420      9      12      0 00:04:09 Established      3
```

Step 6 Configure OSPF multi-instance between the MCE and PE2.

Configure PE2.

```
[PE2] ospf 100 vpn-instance vpna
[PE2-ospf-100] area 0
[PE2-ospf-100-area-0.0.0.0] network 192.1.1.0 0.0.0.255
[PE2-ospf-100-area-0.0.0.0] quit
[PE2-ospf-100] import-route bgp
[PE2-ospf-100] quit
[PE2] ospf 200 vpn-instance vpnb
[PE2-ospf-200] area 0
[PE2-ospf-200-area-0.0.0.0] network 192.2.1.0 0.0.0.255
[PE2-ospf-200-area-0.0.0.0] quit
[PE2-ospf-200] import-route bgp
[PE2-ospf-200] quit
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] import-route ospf 100
[PE2-bgp-vpna] quit
[PE2-bgp] ipv4-family vpn-instance vpnb
[PE2-bgp-vpnb] import-route ospf 200
[PE2-bgp-vpnb] quit
[PE2-bgp] quit
```

Configure the MCE.

```
[MCE] ospf 100 vpn-instance vpna
[MCE-ospf-100] area 0
[MCE-ospf-100-area-0.0.0.0] network 192.1.1.0 0.0.0.255
[MCE-ospf-100-area-0.0.0.0] quit
[MCE-ospf-100] quit
[MCE] ospf 200 vpn-instance vpnb
[MCE-ospf-200] area 0
[MCE-ospf-200-area-0.0.0.0] network 192.2.1.0 0.0.0.255
[MCE-ospf-200-area-0.0.0.0] quit
[MCE-ospf-200] quit
```

Step 7 Configure RIPv2 between the MCE and CE3, and between the MCE and CE4.

Configure the MCE.

```
[MCE] rip 100 vpn-instance vpna
[MCE-rip-100] version 2
[MCE-rip-100] network 10.0.0.0
[MCE-rip-100] import-route ospf 100
[MCE-rip-100] quit
[MCE] rip 200 vpn-instance vpnb
[MCE-rip-200] version 2
[MCE-rip-200] network 10.0.0.0
[MCE-rip-200] import-route ospf 200
[MCE-rip-200] quit
```

Configure CE3.

```
<Huawei> system-view
[Huawei] sysname CE3
[CE3] rip 100
[CE3-rip-100] version 2
[CE3-rip-100] network 10.0.0.0
[CE3-rip-100] import-route direct
```

Configure CE4.

```
<Huawei> system-view
[Huawei] sysname CE4
```

```
[CE4] rip 200
[CE4-rip-200] version 2
[CE4-rip-200] network 10.0.0.0
[CE4-rip-200] import-route direct
```

Step 8 Disable loop detection on the MCE device and import RIP routes.

```
[MCE] ospf 100 vpn-instance vpna
[MCE-ospf-100] vpn-instance-capability simple
[MCE-ospf-100] import-route rip 100
[MCE-ospf-100] quit
[MCE] ospf 200 vpn-instance vpnb
[MCE-ospf-200] vpn-instance-capability simple
[MCE-ospf-200] import-route rip 200
[MCE-ospf-200] quit
```

Step 9 Verify the configuration.

After the configuration is complete, run the **display ip routing-table vpn-instance** command on the MCE. The command output shows the route to the remote CE.

The VPN instance vpna is used as an example.

```
[MCE] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
Destinations : 8          Routes : 8

Destination/Mask Proto Pre Cost      Flags NextHop      Interface
10.1.1.0/24 O_ASE 150 1          D 192.1.1.1
GigabitEthernet1/0/0.1
10.3.1.0/24 Direct 0 0          D 10.3.1.2
GigabitEthernet3/0/0
10.3.1.2/32 Direct 0 0          D 127.0.0.1
GigabitEthernet3/0/0
10.3.1.255/32 Direct 0 0          D 127.0.0.1
GigabitEthernet3/0/0
192.1.1.0/24 Direct 0 0          D 192.1.1.2
GigabitEthernet1/0/0.1
192.1.1.2/32 Direct 0 0          D 127.0.0.1
GigabitEthernet1/0/0.1
192.1.1.255/32 Direct 0 0          D 127.0.0.1
GigabitEthernet1/0/0.1
255.255.255.255/32 Direct 0 0          D 127.0.0.1 InLoopBack0
```

Run the **display ip routing-table vpn-instance** command on the PE. The command output shows the route to the remote CE.

The VPN instance vpna on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
Destinations : 6          Routes : 6

Destination/Mask Proto Pre Cost      Flags NextHop      Interface
10.1.1.0/24 Direct 0 0          D 10.1.1.2
GigabitEthernet1/0/0
10.1.1.2/32 Direct 0 0          D 127.0.0.1
GigabitEthernet1/0/0
10.1.1.255/32 Direct 0 0          D 127.0.0.1
GigabitEthernet1/0/0
10.3.1.0/24 IBGP 255 2          RD 2.2.2.9
GigabitEthernet3/0/0
192.1.1.0/24 IBGP 255 0          RD 2.2.2.9
```

```
GigabitEthernet3/0/0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

CE1 and CE3 can ping each other, and CE2 and CE4 can ping each other.

The ping from CE1 to CE3 is used as an example.

```
[CE1] ping 10.3.1.1
PING 10.3.1.1: 56 data bytes, press CTRL_C to break
Reply from 10.3.1.1: bytes=56 Sequence=1 ttl=252 time=125 ms
Reply from 10.3.1.1: bytes=56 Sequence=2 ttl=252 time=125 ms
Reply from 10.3.1.1: bytes=56 Sequence=3 ttl=252 time=125 ms
Reply from 10.3.1.1: bytes=56 Sequence=4 ttl=252 time=125 ms
Reply from 10.3.1.1: bytes=56 Sequence=5 ttl=252 time=125 ms
--- 10.3.1.1 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 125/125/125 ms
```

CE1 cannot ping CE2 or CE4. CE3 cannot ping CE2 or CE4.

For example, if you ping CE4 from CE1, the following information is displayed:

```
[CE1] ping 10.4.1.1
PING 10.4.1.1: 56 data bytes, press CTRL_C to break
Request time out
--- 10.4.1.1 ping statistics ---
5 packet(s) transmitted
0 packet(s) received
100.00% packet loss
```

----End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
bgp 65420
```

```
peer 10.2.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.2.1.2 enable
#
return
```

● PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 100:1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
ipv4-family
route-distinguisher 100:2
vpn-target 222:2 export-extcommunity
vpn-target 222:2 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpna
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpnb
ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
ip address 172.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 2.2.2.9 enable
#
ipv4-family vpn-instance vpna
peer 10.1.1.1 as-number 65410
import-route direct
#
ipv4-family vpn-instance vpnb
peer 10.2.1.1 as-number 65420
import-route direct
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 172.1.1.0 0.0.0.255
```

```
#  
return
```

● PE2 configuration file

```
#  
sysname PE2  
#  
ip vpn-instance vpna  
  ipv4-family  
    route-distinguisher 200:1  
    vpn-target 111:1 export-extcommunity  
    vpn-target 111:1 import-extcommunity  
#  
ip vpn-instance vpb  
  ipv4-family  
    route-distinguisher 200:2  
    vpn-target 222:2 export-extcommunity  
    vpn-target 222:2 import-extcommunity  
#  
mpls lsr-id 2.2.2.9  
mpls  
#  
mpls ldp  
#  
interface GigabitEthernet1/0/0  
  ip address 172.1.1.2 255.255.255.0  
  mpls  
  mpls ldp  
#  
interface GigabitEthernet2/0/0.1  
  dot1q termination vid 10  
  ip binding vpn-instance vpna  
  ip address 192.1.1.1 255.255.255.0  
#  
interface GigabitEthernet2/0/0.2  
  dot1q termination vid 20  
  ip binding vpn-instance vpb  
  ip address 192.2.1.1 255.255.255.0  
#  
interface LoopBack1  
  ip address 2.2.2.9 255.255.255.255  
#  
bgp 100  
  peer 1.1.1.9 as-number 100  
  peer 1.1.1.9 connect-interface LoopBack1  
#  
  ipv4-family unicast  
    undo synchronization  
    peer 1.1.1.9 enable  
#  
  ipv4-family vpnv4  
    policy vpn-target  
    peer 1.1.1.9 enable  
#  
  ipv4-family vpn-instance vpna  
    import-route ospf 100  
#  
  ipv4-family vpn-instance vpb  
    import-route ospf 200  
#  
ospf 1  
  area 0.0.0.0  
    network 2.2.2.9 0.0.0.0  
    network 172.1.1.0 0.0.0.255  
#  
ospf 100 vpn-instance vpna  
  import-route bgp  
  area 0.0.0.0  
    network 192.1.1.0 0.0.0.255  
#
```

```
ospf 200 vpn-instance vpb  
import-route bgp  
area 0.0.0.0  
network 192.2.1.0 0.0.0.255  
#  
return
```

- MCE configuration file

```
#  
sysname MCE  
#  
ip vpn-instance vpna  
ipv4-family  
route-distinguisher 300:1  
vpn-target 111:1 export-extcommunity  
vpn-target 111:1 import-extcommunity  
#  
ip vpn-instance vpb  
ipv4-family  
route-distinguisher 300:2  
vpn-target 222:2 export-extcommunity  
vpn-target 222:2 import-extcommunity  
#  
interface GigabitEthernet1/0/0.1  
dot1q termination vid 10  
ip binding vpn-instance vpna  
ip address 192.1.1.2 255.255.255.0  
#  
interface GigabitEthernet1/0/0.2  
dot1q termination vid 20  
ip binding vpn-instance vpb  
ip address 192.2.1.2 255.255.255.0  
#  
interface GigabitEthernet3/0/0  
ip binding vpn-instance vpna  
ip address 10.3.1.2 255.255.255.0  
#  
interface GigabitEthernet4/0/0  
ip binding vpn-instance vpb  
ip address 10.4.1.2 255.255.255.0  
#  
ospf 100 vpn-instance vpna  
import-route rip 100  
vpn-instance-capability simple  
area 0.0.0.0  
network 192.1.1.0 0.0.0.255  
#  
ospf 200 vpn-instance vpb  
import-route rip 200  
vpn-instance-capability simple  
area 0.0.0.0  
network 192.2.1.0 0.0.0.255  
#  
rip 100 vpn-instance vpna  
version 2  
network 10.0.0.0  
import-route ospf 100  
#  
rip 200 vpn-instance vpb  
version 2  
network 10.0.0.0  
import-route ospf 200  
#  
return
```

- CE3 configuration file

```
#  
sysname CE3  
#  
interface GigabitEthernet1/0/0
```

```
ip address 10.3.1.1 255.255.255.0
#
rip 100
version 2
network 10.0.0.0
import-route direct
#
return
```

- CE4 configuration file

```
#
sysname CE4
#
interface GigabitEthernet1/0/0
ip address 10.4.1.1 255.255.255.0
#
rip 200
version 2
network 10.0.0.0
import-route direct
#
return
```

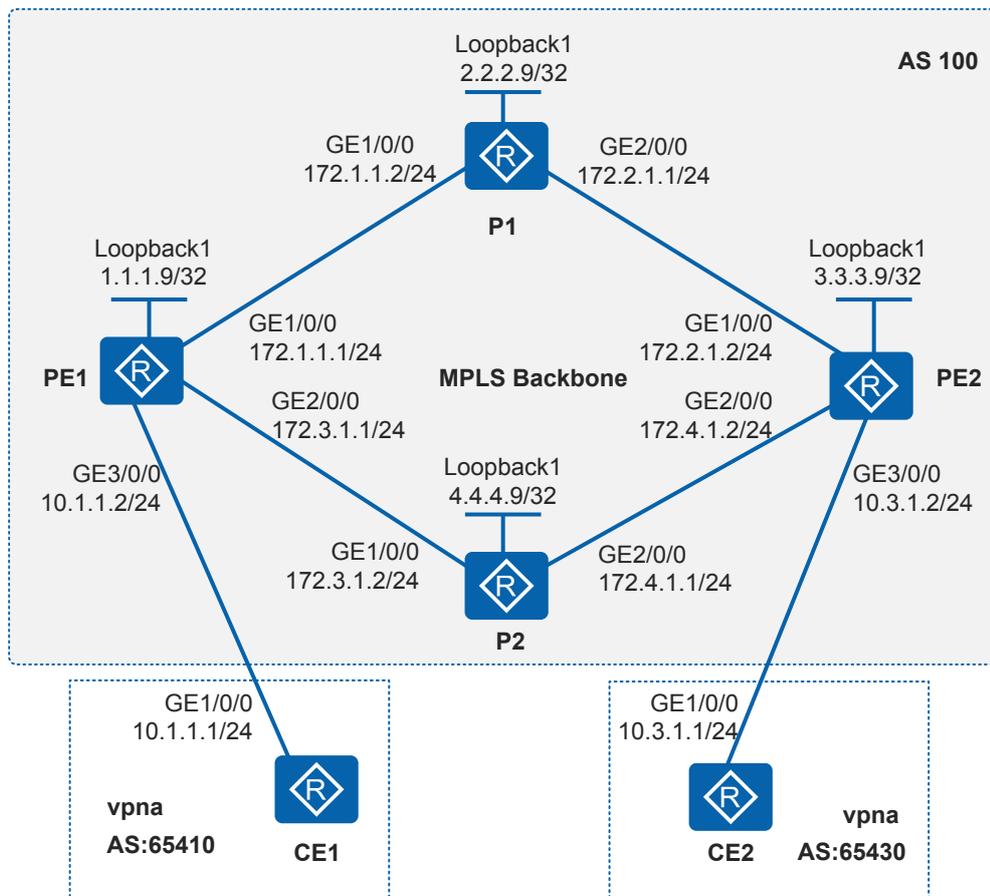
8.9.10 Example for Configuring PBR to an LSP for VPN Packets

Networking Requirements

As shown in [Figure 8-51](#), the BGP/MPLS IP VPN backbone network consists of PE1, PE2, P1, and P2. CE1 and CE2 connect to the backbone network through PE1 and PE2 respectively. The path PE1->P2->PE2 is the primary LSP, and the path PE1->P1->PE2 is the backup LSP.

If the PBR is configured on PE1, packets of 10 to 1000 bytes long sent from CE1 to CE2 are forwarded through P2.

Figure 8-51 Networking diagram for configuring the PBR to an LSP for VPN packets



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure BGP/MPLS VPN according to [8.9.1 Example for Configuring BGP/MPLS IP VPN](#).
2. Configure the PBR and policy node on the PE that requires the configuration of the PBR to an LSP. Set a matching rule of IP packet length and specify an LSP for forwarding VPN instance packets that meet the matching rule in the policy-based route view.
3. Apply the PBR to the outbound interface bound to the VPN instance on the PE.

Procedure

Step 1 Configure BGP/MPLS VPN.

For the configuration procedure, refer to [8.9.1 Example for Configuring BGP/MPLS IP VPN](#).

After the configuration is complete, run the **display mpls lsp** command to check LSPs on PE1.

```
[PE1] display mpls lsp
```

LSP Information: BGP LSP			
FEC	In/Out Label	In/Out IF	Vrf Name
10.1.1.0/24	15360/NULL	-/-	vpna

LSP Information: LDP LSP			
FEC	In/Out Label	In/Out IF	Vrf Name
2.2.2.9/32	NULL/3	-/GE1/0/0	
2.2.2.9/32	1024/3	-/GE1/0/0	
3.3.3.9/32	NULL/1024	-/GE1/0/0	
3.3.3.9/32	NULL/1024	-/GE2/0/0	
4.4.4.9/32	NULL/3	-/GE2/0/0	
4.4.4.9/32	1025/3	-/GE2/0/0	
1.1.1.9/32	3/NULL	-/-	

The LSPs to PE2 have two outbound interfaces: GE1/0/0 and GE2/0/0.

Step 2 Configure the PBR to an LSP on PE1.

```
[PE1] policy-based-route policy1 permit node 10
[PE1-policy-based-route-policy1-10] if-match packet-length 10 1000
[PE1-policy-based-route-policy1-10] apply lsp vpn vpna 10.3.1.1 3.3.3.9 172.3.1.2
[PE1-policy-based-route-policy1-10] quit
```

Step 3 Enable the PBR on PE1.

```
[PE1] ip local policy-based-route policy1
```

Step 4 Clear statistics on GE2/0/0 of PE1.

```
[PE1] quit
<PE1> reset counters interface GigabitEthernet 2/0/0
```

Step 5 Verify the configuration.

Ping CE2 from CE1 to check the forwarding path of the packets.

```
[CE1] ping -c 1500 -s 950 10.3.1.1
```

Check packet statistics on the interface of PE1.

```
<PE1> display interface gigabitethernet 2/0/0
GigabitEthernet2/0/0 current state : UP
Line protocol current state : UP
Last line protocol up time : 2012-09-14 18:13:40
Description:HUAWEI, AR Series, GigabitEthernet2/0/0 Interface
Route Port,The Maximum Transmit Unit is 1500
Internet Address is 172.3.1.1/24
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 80fb-0635-45b6
Last physical up time : 2012-09-14 18:13:40
Last physical down time : 2012-09-14 18:13:23
Current system time: 2012-09-14 18:23:37
Port Mode: COMMON COPPER
Speed : 1000, Loopback: NONE
Duplex: FULL, Negotiation: ENABLE
Mdi : AUTO
Last 300 seconds input rate 456 bits/sec, 0 packets/sec
Last 300 seconds output rate 472 bits/sec, 0 packets/sec
Input peak rate 18088 bits/sec,Record time: 2012-09-14 18:22:50
Output peak rate 18016 bits/sec,Record time: 2012-09-14 18:22:50

Input: 30 packets, 25402 bytes
  Unicast:          26, Multicast:          4
  Broadcast:        0, Jumbo:            0
  Discard:          0, Total Error:        0

  CRC:              0, Giants:            0
  Jabbers:          0, Throttles:         0
  Runts:            0, Symbols:           0
  Ignoreds:         0, Frames:            0
```

```
Output: 31 packets, 25970 bytes
  Unicast:          27, Multicast:          4
  Broadcast:        0, Jumbo:              0
  Discard:           0, Total Error:        0

  Collisions:       0, ExcessiveCollisions: 0
  Late Collisions: 0,  Deferreds:          0

  Input bandwidth utilization threshold : 100.00%
  Output bandwidth utilization threshold: 100.00%
  Input bandwidth utilization   : 0.01%
  Output bandwidth utilization   : 0.01%
```

Run the **display interface gigabitethernet 1/0/0** and **display interface gigabitethernet 2/0/0** commands repeatedly on PE1 to check the change of packet statistics on interfaces of PE1. The command output shows that packets are forwarded along the specified LSP.

----End

Configuration Files

- PE1 configuration file

```
#
 sysname PE1
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
 mpls lsr-id 1.1.1.9
 mpls
#
 mpls ldp
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance vpna
 ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 172.3.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
 bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpna
    peer 10.1.1.1 as-number 65410
```

```
import-route direct
#
ospf 1
 area 0.0.0.0
  network 1.1.1.9 0.0.0.0
  network 172.3.1.0 0.0.0.255
  network 172.1.1.0 0.0.0.255
#
policy-based-route policy1 permit node 10
 if-match packet-length 10 1000
 apply lsp vpn vpna 10.3.1.1 3.3.3.9 172.3.1.2
#
ip local policy-based-route policy1
#
return
```

● P1 configuration file

```
#
sysname P1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 172.2.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 172.2.1.0 0.0.0.255
  network 172.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
 ipv4-family
  route-distinguisher 100:2
  vpn-target 111:1 export-extcommunity
  vpn-target 111:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance vpna
 ip address 10.3.1.2 255.255.255.0
#
interface GigabitEthernet1/0/0
 ip address 172.2.1.2 255.255.255.0
 mpls
 mpls ldp
#
```

```
interface GigabitEthernet2/0/0
 ip address 172.4.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.9 enable
#
 ipv4-family vpn-instance vpna
  peer 10.3.1.1 as-number 65430
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 172.2.1.0 0.0.0.255
  network 172.4.1.0 0.0.0.255
#
return
```

- P2 configuration file

```
#
 sysname P2
#
 mpls lsr-id 4.4.4.9
 mpls
#
 mpls ldp
#
 interface GigabitEthernet1/0/0
  ip address 172.3.1.2 255.255.255.0
  mpls
  mpls ldp
#
 interface GigabitEthernet2/0/0
  ip address 172.4.1.1 255.255.255.0
  mpls
  mpls ldp
#
 interface LoopBack1
  ip address 4.4.4.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 4.4.4.9 0.0.0.0
  network 172.3.1.0 0.0.0.255
  network 172.4.1.0 0.0.0.255
#
return
```

- CE1 configuration file

```
#
 sysname CE1
#
 interface GigabitEthernet1/0/0
  ip address 10.1.1.1 255.255.255.0
#
bgp 65410
```

```
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
import-route direct
undo synchronization
peer 10.1.1.2 enable
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.3.1.1 255.255.255.0
#
bgp 65430
peer 10.3.1.2 as-number 100
#
ipv4-family unicast
import-route direct
undo synchronization
peer 10.3.1.2 enable
#
return
```

8.9.11 Example for Configuring HoVPN

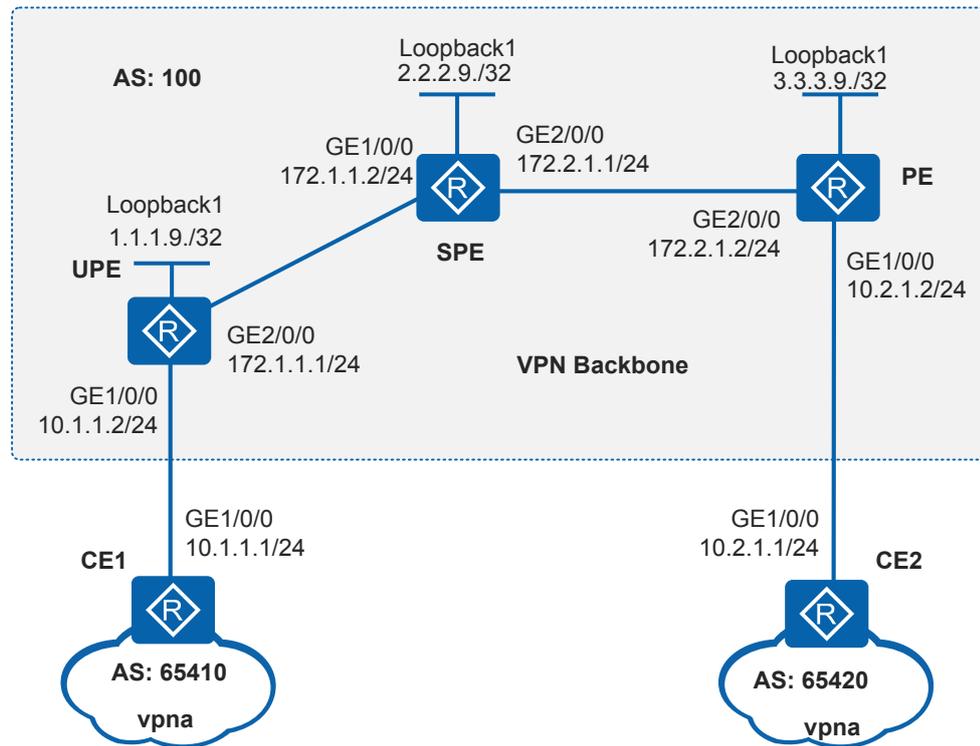
Networking Requirements

Figure 8-52 shows a hierarchical VPN network consisting of a provincial backbone network and a city MPLS VPN network.

- The SPE is located on the provincial backbone network and connects to the city MPLS VPN network.
- The UPE is located on the city network and connects to VPN users.

The routing and forwarding capabilities of the UPE are lower than those of the SPE and PEs. The HoVPN networking can enable users in vpna to communicate with each other while reducing the loads on the UPE.

Figure 8-52 Networking diagram for configuring HoVPN



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IGP on the backbone network to implement IP interworking.
2. Configure basic MPLS capabilities and MPLS LDP on the backbone network to set up MPLS LSPs.
3. Set up MP-IBGP peer relationships between the UPE and SPE and between the PE and SPE to exchange VPN routing information.
4. On the UPE and PEs, create VPN instances and set up EBGP peer relationships with CEs to exchange VPN routing information.
5. On the SPE, create a VPN instance and specify the UPE as its underlayer PE (or user-end PE). Advertise the default route of the VPN instance to the UPE to reduce the loads on the UPE.

Procedure

Step 1 Configure OSPF on the backbone network to implement IP interworking.

Configure the UPE.

```
<Huawei> system-view
[Huawei] sysname UPE
[UPE] interface loopback 1
[UPE-LoopBack1] ip address 1.1.1.9 32
[UPE-LoopBack1] quit
[UPE] interface gigabitethernet 2/0/0
```

```
[UPE-GigabitEthernet2/0/0] ip address 172.1.1.1 24
[UPE-GigabitEthernet2/0/0] quit
[UPE] ospf
[UPE-ospf-1] area 0
[UPE-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[UPE-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[UPE-ospf-1-area-0.0.0.0] quit
[UPE-ospf-1] quit
```

The configuration on the SPE and PEs is similar to the configuration on the UPE and is not mentioned here.

After the configuration is complete, OSPF neighbor relationships are set up between the UPE, SPE, and PE. Run the **display ospf peer** command on these devices. The command output shows that the neighbor relationships are in Full state. Run the **display ip routing-table** command on these devices. The command output shows that they have learned the route to the loopback interface of each other.

- Step 2** Configure basic MPLS capabilities and MPLS LDP on the backbone network to set up LDP LSPs.

Configure the UPE.

```
[UPE] mpls lsr-id 1.1.1.9
[UPE] mpls
[UPE-mpls] quit
[UPE] mpls ldp
[UPE-mpls-ldp] quit
[UPE] interface gigabitethernet 2/0/0
[UPE-GigabitEthernet2/0/0] mpls
[UPE-GigabitEthernet2/0/0] mpls ldp
[UPE-GigabitEthernet2/0/0] quit
```

The configuration on the SPE and PEs is similar to the configuration on the UPE and is not mentioned here.

After the configuration is complete, LDP sessions are established between UPE and SPE, and between SPE and PE. Run the **display mpls ldp session** command on these devices. The command output shows that the status is Operational. Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

- Step 3** Set up MP-IBGP peer relationships between the UPE and SPE and between the PE and SPE.

Configure the UPE.

```
[UPE] bgp 100
[UPE-bgp] peer 2.2.2.9 as-number 100
[UPE-bgp] peer 2.2.2.9 connect-interface loopback 1
[UPE-bgp] ipv4-family vpnv4
[UPE-bgp-af-vpnv4] peer 2.2.2.9 enable
[UPE-bgp-af-vpnv4] quit
[UPE-bgp] quit
```

Configure the SPE.

```
[SPE] bgp 100
[SPE-bgp] peer 1.1.1.9 as-number 100
[SPE-bgp] peer 1.1.1.9 connect-interface loopback 1
[SPE-bgp] peer 3.3.3.9 as-number 100
[SPE-bgp] peer 3.3.3.9 connect-interface loopback 1
[SPE-bgp] ipv4-family vpnv4
[SPE-bgp-af-vpnv4] peer 1.1.1.9 enable
[SPE-bgp-af-vpnv4] peer 3.3.3.9 enable
```

```
[SPE-bgp-af-ipv4] quit  
[SPE-bgp] quit
```

Configure the PE.

```
[PE] bgp 100  
[PE-bgp] peer 2.2.2.9 as-number 100  
[PE-bgp] peer 2.2.2.9 connect-interface loopback 1  
[PE-bgp] ipv4-family vpnv4  
[PE-bgp-af-ipv4] peer 2.2.2.9 enable  
[PE-bgp-af-ipv4] quit  
[PE-bgp] quit
```

Step 4 On the UPE and PEs, create a VPN instance and set up EBGP peer relationships with the CEs.

Configure the UPE.

```
[UPE] ip vpn-instance vpna  
[UPE-vpn-instance-vpna] ipv4-family  
[UPE-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1  
[UPE-vpn-instance-vpna-af-ipv4] vpn-target 1:1  
[UPE-vpn-instance-vpna-af-ipv4] quit  
[UPE-vpn-instance-vpna] quit  
[UPE] interface gigabitethernet 1/0/0  
[UPE-GigabitEthernet1/0/0] ip binding vpn-instance vpna  
[UPE-GigabitEthernet1/0/0] ip address 10.1.1.2 24  
[UPE-GigabitEthernet1/0/0] quit  
[UPE] bgp 100  
[UPE-bgp] ipv4-family vpn-instance vpna  
[UPE-bgp-vpna] peer 10.1.1.1 as-number 65410  
[UPE-bgp-vpna] import-route direct  
[UPE-bgp-vpna] quit  
[UPE-bgp] quit
```

Configure CE1.

```
<Huawei> system-view  
[Huawei] sysname CE1  
[CE1] interface gigabitethernet 1/0/0  
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24  
[CE1-GigabitEthernet1/0/0] quit  
[CE1] bgp 65410  
[CE1-bgp] peer 10.1.1.2 as-number 100  
[CE1-bgp] import-route direct  
[CE1-bgp] quit
```

Configure the PE.

```
[PE] ip vpn-instance vpna  
[PE-vpn-instance-vpna] ipv4-family  
[PE-vpn-instance-vpna-af-ipv4] route-distinguisher 100:2  
[PE-vpn-instance-vpna-af-ipv4] vpn-target 1:1  
[PE-vpn-instance-vpna-af-ipv4] quit  
[PE-vpn-instance-vpna] quit  
[PE] interface gigabitethernet 1/0/0  
[PE-GigabitEthernet1/0/0] ip binding vpn-instance vpna  
[PE-GigabitEthernet1/0/0] ip address 10.2.1.2 24  
[PE-GigabitEthernet1/0/0] quit  
[PE] bgp 100  
[PE-bgp] ipv4-family vpn-instance vpna  
[PE-bgp-vpna] peer 10.2.1.1 as-number 65420  
[PE-bgp-vpna] import-route direct  
[PE-bgp-vpna] quit  
[PE-bgp] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 10.2.1.1 24
[CE2-GigabitEthernet1/0/0] quit
[CE2] bgp 65420
[CE2-bgp] peer 10.2.1.2 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the UPE and PEs to check the configuration of VPN instances. Run the **ping -vpn-instance** command on the UPE and PEs to ping the connected CEs. The ping operations succeed.

NOTE

If a PE has multiple interfaces bound to the same VPN instance, you need to specify the source IP addresses by setting **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping the remote CE. If the source IP address is not specified, the ping operation fails.

UPE is used as an example.

```
[UPE] display ip vpn-instance verbose
Total VPN-Instances configured : 1
Total IPv4 VPN-Instances configured : 1
Total IPv6 VPN-Instances configured : 0

VPN-Instance Name and ID : vpna, 1
  Interfaces : GigabitEthernet1/0/0
Address family ipv4
  Create date : 2012/09/14 14:34:10
  Up time : 0 days, 00 hours, 16 minutes and 01 seconds
  Route Distinguisher : 100:1
  Export VPN Targets : 1:1
  Import VPN Targets : 1:1
  Label Policy : label per route
  Log Interval : 5
```

Step 5 On the SPE, create a VPN instance, specify the UPE as its underlayer PE, and advertise the default route of the VPN instance to the UPE.

Configure the VPN instance.

```
[SPE] ip vpn-instance vpna
[SPE-vpn-instance-vpna] route-distinguisher 200:1
[SPE-vpn-instance-vpna] vpn-target 1:1
[SPE-vpn-instance-vpna] quit
```

Specify the UPE for the SPE.

```
[SPE] bgp 100
[SPE-bgp] ipv4-family vpnv4
[SPE-bgp-af-vpnv4] peer 1.1.1.9 upe
```

Advertise the default route of the VPN instance to the UPE.

```
[SPE-bgp-af-vpnv4] peer 1.1.1.9 default-originate vpn-instance vpna
[SPE-bgp-af-vpnv4] quit
[SPE-bgp] quit
```

Step 6 Verify the configuration.

After the configuration is complete, CE1 has no route to the network segment of the interface on CE2, but CE1 has a default route with the next hop as UPE. CE2 has a BGP route to the network segment of the interface on CE1. CE1 and CE2 can ping each other.

```
[CE1] display ip routing-table
Route Flags: R - relay,
```

```
D - download to fib
-----
Routing Tables: Public
  Destinations : 8          Routes : 8

  Destination/Mask  Proto  Pre  Cost    Flags  NextHop          Interface
-----
      0.0.0.0/0     EBGP   255  0        D    10.1.1.2
GigabitEthernet1/0/0
      10.1.1.0/24   Direct  0     0        D    10.1.1.1
GigabitEthernet1/0/0
      10.1.1.1/32   Direct  0     0        D    127.0.0.1
GigabitEthernet1/0/0
      10.1.1.255/32 Direct  0     0        D    127.0.0.1
GigabitEthernet1/0/0
      127.0.0.0/8   Direct  0     0        D    127.0.0.1          InLoopBack0
      127.0.0.1/32 Direct  0     0        D    127.0.0.1          InLoopBack0
127.255.255.255/32 Direct  0     0        D    127.0.0.1          InLoopBack0
255.255.255.255/32 Direct  0     0        D    127.0.0.1          InLoopBack0
```

```
[CE1] ping 10.2.1.1
PING 10.2.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=252 time=2 ms
  Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=252 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=252 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=252 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=252 time=1 ms

--- 10.2.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/2 ms
```

```
[CE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
  Destinations : 8          Routes : 8

  Destination/Mask  Proto  Pre  Cost    Flags  NextHop          Interface
-----
      10.1.1.0/24   EBGP   255  0        D    10.2.1.2
GigabitEthernet1/0/0
      10.2.1.0/24   Direct  0     0        D    10.2.1.1
GigabitEthernet1/0/0
      10.2.1.1/32   Direct  0     0        D    127.0.0.1
GigabitEthernet1/0/0
      10.2.1.255/32 Direct  0     0        D    127.0.0.1
GigabitEthernet1/0/0
      127.0.0.0/8   Direct  0     0        D    127.0.0.1          InLoopBack0
      127.0.0.1/32 Direct  0     0        D    127.0.0.1          InLoopBack0
127.255.255.255/32 Direct  0     0        D    127.0.0.1          InLoopBack0
255.255.255.255/32 Direct  0     0        D    127.0.0.1          InLoopBack0
```

Run the **display bgp vpnv4 all routing-table** command on the UPE. The command output shows a default route of vpnv4 with the next hop as SPE.

```
[UPE] display bgp vpnv4 all routing-table

BGP Local router ID is 1.1.1.9
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total number of routes from all PE: 4
```

```

Route Distinguisher: 100:1

      Network          NextHop      MED      LocPrf    PrefVal Path/Ogn
* >  10.1.1.0/24      0.0.0.0      0         0         0      ?
*    10.1.1.1        10.1.1.1      0         0         0      65410?
* >  10.1.1.2/32      0.0.0.0      0         0         0      ?

Route Distinguisher: 200:1

      Network          NextHop      MED      LocPrf    PrefVal Path/Ogn
* > i 0.0.0.0         2.2.2.9      0         100        0      i

VPN-Instance vpna, Router ID 1.1.1.9:

Total Number of Routes: 4
      Network          NextHop      MED      LocPrf    PrefVal Path/Ogn
* > i 0.0.0.0         2.2.2.9      0         100        0      i
* >  10.1.1.0/24      0.0.0.0      0         0         0      ?
*    10.1.1.1        10.1.1.1      0         0         0      65410?
* >  10.1.1.2/32      0.0.0.0      0         0         0      ?
  
```

----End

Configuration Files

- CE1 configuration file

```

#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
  
```

- UPE configuration file

```

#
sysname UPE
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 100:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpna
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.1.1.1 255.255.255.0
  
```

```
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 2.2.2.9 enable
#
ipv4-family vpn-instance vpna
peer 10.1.1.1 as-number 65410
import-route direct
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● SPE configuration file

```
#
sysname SPE
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 200:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 172.2.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 3.3.3.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
peer 3.3.3.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
peer 3.3.3.9 enable
#
ipv4-family vpnv4
policy vpn-target
```

```
peer 1.1.1.9 enable
peer 1.1.1.9 upe
peer 1.1.1.9 default-originate vpn-instance vpna
peer 3.3.3.9 enable
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
network 172.2.1.0 0.0.0.255
#
return
```

● PE configuration file

```
#
sysname PE
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 100:2
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpna
ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.2.1.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 2.2.2.9 enable
#
ipv4-family vpn-instance vpna
peer 10.2.1.1 as-number 65420
import-route direct
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 172.2.1.0 0.0.0.255
#
return
```

● CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
```

```

    bgp 65420
    peer 10.2.1.2 as-number 100
    #
    ipv4-family unicast
    undo synchronization
    import-route direct
    peer 10.2.1.2 enable
    #
    return
    
```

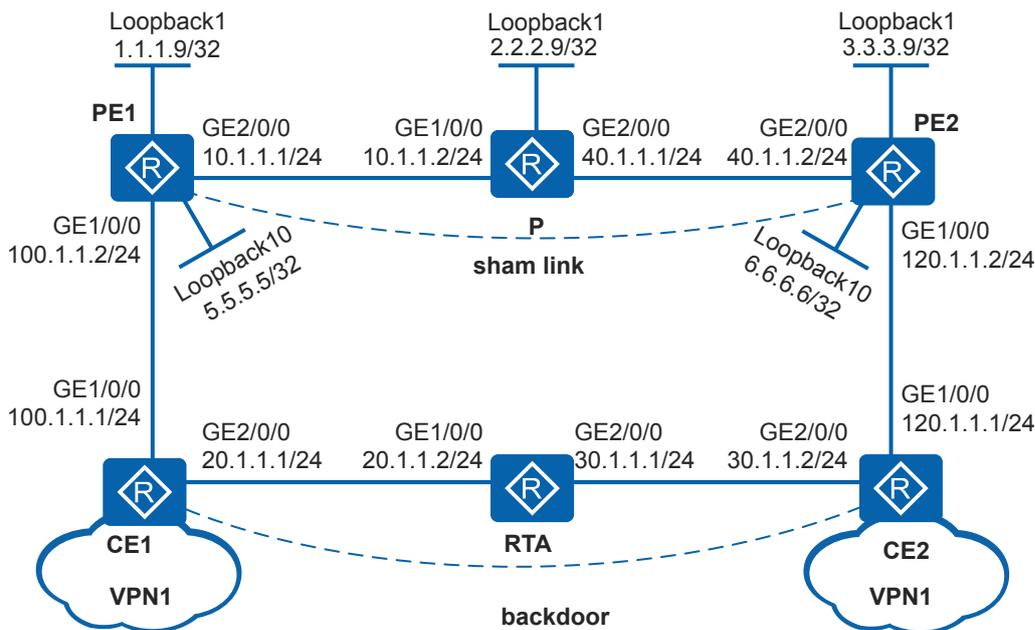
8.9.12 Example for Configuring an OSPF Sham Link

Networking Requirements

As shown in **Figure 8-53**, CE1 and CE2 belong to the same OSPF area of VPN1 and they connect to PE1 and PE2 respectively. A backdoor link exists between CE1 and CE2 and is used as a backup link.

The CEs and PEs need to run OSPF. When the backbone network is running properly, VPN traffic of CE1 and CE2 should be forwarded over the MPLS backbone network without passing through the backdoor link.

Figure 8-53 Networking diagram for configuring OSPF sham link



Configuration Roadmap

The configuration roadmap is as follows:

1. Set up an ME-IBGP peer relationship between the PEs and configure OSPF between the PEs and CEs.
2. Create a VPN instance on the PEs and bind it to the interfaces connected to CEs.
3. Create an OSPF sham link on the PEs.

4. Set the cost of the backdoor link to be larger than the cost of the sham link so that VPN traffic is transmitted over the MPLS backbone network.

Procedure

Step 1 Configure OSPF on the customer network.

Configure OSPF on CE1, RTA, and CE2 and advertise the network segment of each interface.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet2/0/0
[CE1-GigabitEthernet2/0/0] ip address 20.1.1.1 24
[CE1-GigabitEthernet2/0/0] quit
[CE1] interface gigabitethernet1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] ospf
[CE1-ospf-1] area 0
[CE1-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[CE1-ospf-1-area-0.0.0.0] network 100.1.1.0 0.0.0.255
[CE1-ospf-1-area-0.0.0.0] quit
[CE1-ospf-1] quit
```

Configure RTA.

```
<Huawei> system-view
[Huawei] sysname RTA
[RTA] interface gigabitethernet 1/0/0
[RTA-GigabitEthernet1/0/0] ip address 20.1.1.2 24
[RTA-GigabitEthernet1/0/0] quit
[RTA] interface gigabitethernet 2/0/0
[RTA-GigabitEthernet2/0/0] ip address 30.1.1.1 24
[RTA-GigabitEthernet2/0/0] quit
[RTA] ospf
[RTA-ospf-1] area 0
[RTA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RTA-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[RTA-ospf-1-area-0.0.0.0] quit
[RTA-ospf-1] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] ip address 30.1.1.2 24
[CE2-GigabitEthernet2/0/0] quit
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 120.1.1.2 24
[CE2-GigabitEthernet1/0/0] quit
[CE2] ospf
[CE2-ospf-1] area 0
[CE2-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[CE2-ospf-1-area-0.0.0.0] network 120.1.1.0 0.0.0.255
[CE2-ospf-1-area-0.0.0.0] quit
[CE2-ospf-1] quit
```

Step 2 Complete basic BGP/MPLS IP VPN configuration on the backbone network: configure an IGP, enable MPLS and LDP, and set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
```

```
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
[PE1] ospf 1 router-id 1.1.1.9
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure P.

```
<Huawei> system-view
[Huawei] sysname P
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] ip address 40.1.1.1 24
[P-GigabitEthernet2/0/0] quit
[P] interface loopback 1
[P-LoopBack1] ip address 2.2.2.9 32
[P-LoopBack1] quit
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
[P] ospf 1 router-id 2.2.2.9
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[P-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] network 40.1.1.0 0.0.0.255
[P-ospf-1-area-0.0.0.0] quit
[P-ospf-1] quit
```

Configure PE2.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip address 40.1.1.2 24
[PE2-GigabitEthernet2/0/0] quit
```

```
[PE2] interface loopback 1
[PE2-LoopBack1] ip address 3.3.3.9 32
[PE2-LoopBack1] quit
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls
[PE2-GigabitEthernet2/0/0] mpls ldp
[PE2-GigabitEthernet2/0/0] quit
[PE2] ospf 1 router-id 3.3.3.9
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 3.3.3.9 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] network 40.1.1.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

After the configuration is complete, PE1 and PE2 can learn the route to the loopback interface of each other and set up an MP-IBGP peer relationship.

Step 3 Configure OSPF between the PEs and CEs.

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 100.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] ospf 100 router-id 5.5.5.5 vpn-instance vpn1
[PE1-ospf-100] domain-id 10
[PE1-ospf-100] import-route bgp
[PE1-ospf-100] area 0
[PE1-ospf-100-area-0.0.0.0] network 100.1.1.0 0.0.0.255
[PE1-ospf-100-area-0.0.0.0] quit
[PE1-ospf-100] quit
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] import-route ospf 100
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 1:1
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet1/0/0] ip address 120.1.1.1 24
[PE2-GigabitEthernet1/0/0] quit
```

```
[PE2] ospf 100 router-id 6.6.6.6 vpn-instance vpn1
[PE2-ospf-100] import-route bgp
[PE2-ospf-100] domain-id 10
[PE2-ospf-100] area 0
[PE2-ospf-100-area-0.0.0.0] network 120.1.1.0 0.0.0.255
[PE2-ospf-100-area-0.0.0.0] quit
[PE2-ospf-100] quit
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] import-route direct
[PE2-bgp-vpn1] import-route ospf 100
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

After the configuration is complete, run the **display ip routing-table vpn-instance** command on the PEs. The command output shows that the routes to the remote CEs are OSPF routes through the customer network, not the BGP routes through the backbone network.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
  Destinations : 7          Routes : 7
-----
  Destination/Mask  Proto  Pre  Cost           Flags  NextHop         Interface
  20.1.1.0/24       OSPF   10   2              D      100.1.1.1
GigabitEthernet1/0/0
  30.1.1.0/24       OSPF   10   3              D      100.1.1.1
GigabitEthernet1/0/0
  100.1.1.0/24      Direct 0     0              D      100.1.1.2
GigabitEthernet1/0/0
  100.1.1.2/32      Direct 0     0              D      127.0.0.1
GigabitEthernet1/0/0
  100.1.1.255/32    Direct 0     0              D      127.0.0.1
GigabitEthernet1/0/0
  120.1.1.0/24      OSPF   10   4              D      100.1.1.1
GigabitEthernet1/0/0
  255.255.255.255/32 Direct 0     0              D      127.0.0.1      InLoopBack0
```

Step 4 Configure an OSPF sham link.

NOTE

To forward VPN traffic through the MPLS backbone network, ensure that the cost of the sham link is smaller than the cost of the OSPF route used for forwarding VPN traffic over the customer network. A commonly used method is to set the cost of the forwarding interface on the customer network to be larger than the cost of the sham link.

Configure CE1.

```
[CE1] interface gigabitethernet 2/0/0
[CE1-GigabitEthernet2/0/0] ospf cost 10
[CE1-GigabitEthernet2/0/0] quit
```

Configure CE2.

```
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] ospf cost 10
[CE2-GigabitEthernet2/0/0] quit
```

Configure PE1.

```
[PE1] interface loopback 10
[PE1-LoopBack10] ip binding vpn-instance vpn1
[PE1-LoopBack10] ip address 5.5.5.5 32
[PE1-LoopBack10] quit
[PE1] ospf 100
```

```
[PE1-ospf-100] area 0
[PE1-ospf-100-area-0.0.0.0] sham-link 5.5.5.5 6.6.6.6 cost 1
[PE1-ospf-100-area-0.0.0.0] quit
[PE1-ospf-100] quit
```

Configure PE2.

```
[PE2] interface loopback 10
[PE2-LoopBack10] ip binding vpn-instance vpn1
[PE2-LoopBack10] ip address 6.6.6.6 32
[PE2-LoopBack10] quit
[PE2] ospf 100
[PE2-ospf-100] area 0
[PE2-ospf-100-area-0.0.0.0] sham-link 6.6.6.6 5.5.5.5 cost 1
[PE2-ospf-100-area-0.0.0.0] quit
[PE2-ospf-100] quit
```

Step 5 Verify the configuration.

After the configuration is complete, run the **display ip routing-table vpn-instance** command on the PEs. The command output shows that the routes to the remote CEs are BGP routes through the backbone network, and there are routes to the destination of the sham link.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
Destinations : 9          Routes : 9
Destination/Mask Proto Pre Cost Flags NextHop Interface
5.5.5.5/32 Direct 0 0 D 127.0.0.1 LoopBack10
6.6.6.6/32 IBGP 255 0 RD 3.3.3.9
GigabitEthernet2/0/0
20.1.1.0/24 OSPF 10 11 D 100.1.1.1
GigabitEthernet1/0/0
30.1.1.0/24 OSPF 10 12 D 100.1.1.1
GigabitEthernet1/0/0
100.1.1.0/24 Direct 0 0 D 100.1.1.2
GigabitEthernet1/0/0
100.1.1.2/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
100.1.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
120.1.1.0/24 IBGP 255 0 RD 3.3.3.9
GigabitEthernet2/0/0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Run the **display ip routing-table** command on the CEs. The command output shows that the cost of the OSPF route to the remote CE has changed to 3, and the next hop has changed to the interface connected to PE. That is, VPN traffic to the remote CE is forwarded through the backbone network.

The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
Destinations : 14          Routes : 14
Destination/Mask Proto Pre Cost Flags NextHop Interface
5.5.5.5/32 O_ASE 150 1 D 100.1.1.2
GigabitEthernet1/0/0
6.6.6.6/32 O_ASE 150 1 D 100.1.1.2
GigabitEthernet1/0/0
20.1.1.0/24 Direct 0 0 D 20.1.1.1
```

```
GigabitEthernet2/0/0
 20.1.1.1/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
 20.1.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
 30.1.1.0/24 OSPF 10 11 D 20.1.1.2
GigabitEthernet2/0/0
 100.1.1.0/24 Direct 0 0 D 100.1.1.1
GigabitEthernet1/0/0
 100.1.1.1/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
 100.1.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
 120.1.1.0/24 OSPF 10 3 D 100.1.1.2
GigabitEthernet1/0/0
 127.0.0.0/8 Direct 0 0 D 127.0.0.1 InLoopBack0
 127.0.0.1/32 Direct 0 0 D 127.0.0.1 InLoopBack0
127.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

NOTE

Cost of the OSPF route from CE1 to CE2 = Cost of the path from CE1 to PE1 + Cost of the sham link + Cost of the path from PE2 to CE2 = 1 + 1 + 1 = 3

Run the **tracert** command on CE1. The command output shows that the data sent from CE1 to CE2 passes through the interface connected to PE1. That is, VPN traffic is transmitted through the backbone network.

```
[CE1] tracert 120.1.1.1
tracert to 120.1.1.1(120.1.1.1), max hops: 30 ,packet length: 40,press
CTRL_C to break
 1 100.1.1.2 10 ms 1 ms 1 ms
 2 10.1.1.2 10 ms 1 ms 1 ms
 3 120.1.1.1 10 ms 2 ms 1 ms
[CE1] tracert 30.1.1.2
tracert to 30.1.1.2(30.1.1.2), max hops: 30 ,packet length: 40,press CTRL_C
to break
 1 20.1.1.2 10 ms 1 ms 1 ms
 2 30.1.1.2 10 ms 2 ms 1 ms
```

Run the **display ospf 100 sham-link** command on the PEs to check information about the sham link.

The information displayed on PE1 is used as an example.

```
[PE1] display ospf 100 sham-link

      OSPF Process 100 with Router ID 5.5.5.5
Sham Link:
Area          NeighborId      Source-IP        Destination-IP   State Cost
0.0.0.0       6.6.6.6         5.5.5.5          6.6.6.6          P-2-P 1
```

Run the **display ospf sham-link area 0** command. The command output shows that the neighbor relationship is in Full state.

```
[PE1] display ospf sham-link area 0

      OSPF Process 1 with Router ID 1.1.1.9

      OSPF Process 100 with Router ID 5.5.5.5

Sham-Link: 5.5.5.5 --> 6.6.6.6
Neighbor ID: 6.6.6.6, State: Full, GR status: Normal
Area: 0.0.0.0
Cost: 1 State: P-2-P, Type: Sham
Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1
```

Run the **display ospf routing** command on the CEs. The command output shows that the route to the remote CE is learned as an intra-area route.

```
[CE1] display ospf routing
      OSPF Process 1 with Router ID 100.1.1.1
      Routing Tables
Routing for Network
Destination      Cost   Type      NextHop      AdvRouter     Area
120.1.1.0/24    3      Transit   100.1.1.2    6.6.6.6       0.0.0.0
20.1.1.0/24     10     Transit   20.1.1.1     100.1.1.1     0.0.0.0
30.1.1.0/24     11     Transit   20.1.1.2     30.1.1.1     0.0.0.0
100.1.1.0/24    1      Transit   100.1.1.1    100.1.1.1     0.0.0.0
Routing for ASEs
Destination      Cost   Type      Tag           NextHop      AdvRouter
6.6.6.6/32      1      Type2     3489661028   100.1.1.2    5.5.5.5
5.5.5.5/32      1      Type2     3489661028   100.1.1.2    6.6.6.6

Total Nets: 6
Intra Area: 4  Inter Area: 0  ASE: 2  NSSA: 0
```

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:1
    vpn-target 1:1 export-extcommunity
    vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn1
  ip address 100.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
interface LoopBack10
  ip binding vpn-instance vpn1
  ip address 5.5.5.5 255.255.255.255
#
bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.9 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpn1
```

```
import-route direct
import-route ospf 100
#
ospf 1 router-id 1.1.1.9
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 10.1.1.0 0.0.0.255
#
ospf 100 router-id 5.5.5.5 vpn-instance vpn1
import-route bgp
domain-id 0.0.0.10
area 0.0.0.0
network 100.1.1.0 0.0.0.255
sham-link 5.5.5.5 6.6.6.6
#
return
```

● P configuration file

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 10.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 40.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1 router-id 2.2.2.9
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 10.1.1.0 0.0.0.255
network 40.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:2
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 120.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 40.1.1.2 255.255.255.0
mpls
mpls ldp
```

```
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
interface LoopBack10
ip binding vpn-instance vpn1
ip address 6.6.6.6 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpn1
import-route direct
import-route ospf 100
#
ospf 1 router-id 3.3.3.9
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 40.1.1.0 0.0.0.255
#
ospf 100 router-id 6.6.6.6 vpn-instance vpn1
import-route bgp
domain-id 0.0.0.10
area 0.0.0.0
network 120.1.1.0 0.0.0.255
sham-link 6.6.6.6 5.5.5.5
#
return
```

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 20.1.1.1 255.255.255.0
ospf cost 10
#
ospf 1
area 0.0.0.0
network 100.1.1.0 0.0.0.255
network 20.1.1.0 0.0.0.255
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 120.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 30.1.1.2 255.255.255.0
ospf cost 10
#
ospf 1
area 0.0.0.0
network 30.1.1.0 0.0.0.255
```

```
network 120.1.1.0 0.0.0.255
#
return
```

● RTA configuration file

```
#
sysname RTA
#
interface GigabitEthernet1/0/0
ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
#
ospf 1
area 0.0.0.0
network 20.1.1.0 0.0.0.255
network 30.1.1.0 0.0.0.255
#
return
```

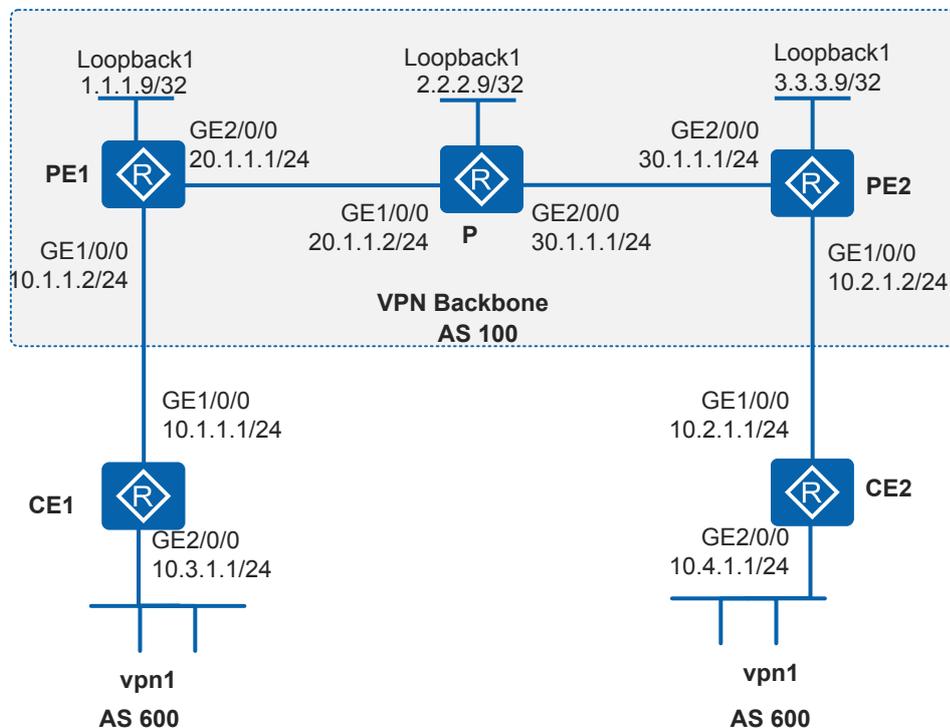
8.9.13 Example for Configuring BGP AS Number Substitution

Networking Requirements

As shown in [Figure 8-54](#), CE1 and CE2 belong to the same VPN. CE1 connects to PE1, and CE2 connects to PE2. Both CE1 and CE2 use AS number 600.

The PEs and CEs need to set up EBGP peer relationships to allow communication between VPN users.

Figure 8-54 Networking diagram for configuring BGP AS number substitution



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the P and PEs to ensure IP connectivity on the backbone network.
2. Configure basic MPLS capabilities and MPLS LDP on the P and PEs to set up MPLS LSP tunnels for VPN data transmission on the backbone network.
3. Set up an MP-IBGP peer relationship between PEs to exchange VPNv4 routes.
4. Configure a VPN instance and set the VPN target to 1:1 on PE1 and PE2 so that users in the VPN can communicate with each other. Bind the VPN instance to the PE interfaces connected to CEs to provide access for VPN users.
5. Set up EBGP peer relationships between the PEs and CEs and import routes of the CEs into routing tables of the PEs.
6. Configure BGP AS number substitution on the PEs to enable them to accept routes with the local AS number.

Procedure

Step 1 Configure basic BGP/MPLS IP VPN functions.

The configurations include the following:

- Configure OSPF on the MPLS backbone network so that the PEs and P can learn the routes to the loopback interface of each other.
- Configure basic MPLS capabilities and MPLS LDP on the backbone network to set up MPLS LSPs.
- Set up an MP-IBGP peer relationship between PEs to exchange VPNv4 routes.
- Configure the VPN instance of VPN1 on PE2 and bind the VPN instance to the interface connected to CE2.
- Configure the VPN instance of VPN1 on PE1 and bind the VPN instance to the interface connected to CE1.
- Set up BGP peer relationships between PE1 and CE1 and between PE2 and CE2 to import routes of CEs to PEs.

For detailed configuration, refer to [8.9.1 Example for Configuring BGP/MPLS IP VPN](#).

After the configuration is complete, run the **display ip routing-table** command on CE2 to check the routing table. The routing table on CE2 contains the route to the network segment (10.1.1.0/24) of interface that connects CE1 to PE1 but contains no route to the VPN (10.3.1.0/24) of CE1. This is the same on CE1.

```
[CE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
  Destinations : 11          Routes : 11
  Destination/Mask  Proto  Pre  Cost   Flags  NextHop    Interface
  10.1.1.0/24      EBGP   255  0      D      10.2.1.2
GigabitEthernet1/0/0
  10.2.1.0/24      Direct 0     0      D      10.2.1.1
GigabitEthernet1/0/0
  10.2.1.1/32      Direct 0     0      D      127.0.0.1
GigabitEthernet1/0/0
  10.2.1.255/32   Direct 0     0      D      127.0.0.1
```

```
GigabitEthernet1/0/0
 127.0.0.0/8 Direct 0 0 D 127.0.0.1 InLoopBack0
 127.0.0.1/32 Direct 0 0 D 127.0.0.1 InLoopBack0
127.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
 10.4.1.0/24 Direct 0 0 D 10.4.1.1
GigabitEthernet2/0/0
 10.4.1.1/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
 10.4.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet2/0/0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Run the **display ip routing-table vpn-instance** command on the PEs to check the routing table of the VPN instance. The VPN routing table has routes to the VPN of the CEs.

The information displayed on PE2 is used as an example.

```
[PE2] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
Destinations : 7 Routes : 7
Destination/Mask Proto Pre Cost Flags NextHop Interface
-----
10.1.1.0/24 IBGP 255 0 RD 1.1.1.9
GigabitEthernet2/0/0
10.2.1.0/24 Direct 0 0 D 10.2.1.2
GigabitEthernet1/0/0
10.2.1.2/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
10.2.1.255/32 Direct 0 0 D 127.0.0.1
GigabitEthernet1/0/0
10.3.1.0/24 IBGP 255 0 RD 1.1.1.9
GigabitEthernet2/0/0
10.4.1.0/24 EBGP 255 0 D 10.2.1.1
GigabitEthernet1/0/0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Run the **display bgp routing-table peer received-routes** command on CE2. The command output shows that CE2 did not accept the route to 10.3.1.0/24.

```
[CE2] display bgp routing-table peer 10.2.1.2 received-routes

BGP Local router ID is 10.2.1.1
Status codes: * - valid, > - best, d - damped,
h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 2
Network NextHop MED LocPrf PrefVal Path/Ogn
* > 10.1.1.0/24 10.2.1.2 0 100?
10.2.1.0/24 10.2.1.2 0 100?
```

Step 2 Configure BGP AS number substitution.

Configure BGP AS number substitution on the PEs.

Configure PE2. PE2 is used as an example.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 substitute-as
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

Check the routing information accepted by CE2 and routing table on CE2.

```
[CE2] display bgp routing-table peer 10.2.1.2 received-routes

BGP Local router ID is 10.2.1.1
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 3
  Network          NextHop          MED          LocPrf        PrefVal Path/Ogn
* > 10.1.1.0/24     10.2.1.2
  10.2.1.0/24     10.2.1.2         0
* > 10.3.1.0/24     10.2.1.2         0          100 100?
```

```
[CE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
  Destinations : 12          Routes : 12
  Destination/Mask Proto Pre Cost      Flags NextHop          Interface
  10.1.1.0/24     EBGP 255 0          D      10.2.1.2
GigabitEthernet1/0/0
  10.2.1.0/24     Direct 0 0          D      10.2.1.1
GigabitEthernet1/0/0
  10.2.1.1/32     Direct 0 0          D      127.0.0.1
GigabitEthernet1/0/0
  10.2.1.255/32   Direct 0 0          D      127.0.0.1
GigabitEthernet1/0/0
  10.3.1.0/24     EBGP 255 0          D      10.2.1.2
GigabitEthernet1/0/0
  127.0.0.0/8     Direct 0 0          D      127.0.0.1      InLoopBack0
  127.0.0.1/32   Direct 0 0          D      127.0.0.1      InLoopBack0
127.255.255.255/32 Direct 0 0          D      127.0.0.1      InLoopBack0
  10.4.1.0/24     Direct 0 0          D      10.4.1.1
GigabitEthernet2/0/0
  10.4.1.1/32     Direct 0 0          D      127.0.0.1
GigabitEthernet2/0/0
  10.4.1.255/32   Direct 0 0          D      127.0.0.1
GigabitEthernet2/0/0
255.255.255.255/32 Direct 0 0          D      127.0.0.1      InLoopBack0
```

After configuring BGP AS number substitution on PE1, you can find that CE1 and CE2 can successfully ping each other.

```
[CE1] ping -a 10.3.1.1 10.4.1.1
PING 10.4.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.4.1.1: bytes=56 Sequence=1 ttl=252 time=2 ms
  Reply from 10.4.1.1: bytes=56 Sequence=2 ttl=252 time=1 ms
  Reply from 10.4.1.1: bytes=56 Sequence=3 ttl=252 time=2 ms
  Reply from 10.4.1.1: bytes=56 Sequence=4 ttl=252 time=2 ms
  Reply from 10.4.1.1: bytes=56 Sequence=5 ttl=252 time=2 ms

--- 10.4.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/2 ms
```

----End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
```

```
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.3.1.1 255.255.255.0
#
bgp 600
 peer 10.1.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.1.1.2 enable
#
return
```

● PE1 configuration file

```
#
 sysname PE1
#
 ip vpn-instance vpn1
  ipv4-family
   route-distinguisher 100:1
   vpn-target 1:1 export-extcommunity
   vpn-target 1:1 import-extcommunity
#
 mpls lsr-id 1.1.1.9
 mpls
#
 mpls ldp
#
 interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn1
  ip address 10.1.1.2 255.255.255.0
#
 interface GigabitEthernet2/0/0
  ip address 20.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
 interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
 bgp 100
  peer 3.3.3.9 as-number 100
  peer 3.3.3.9 connect-interface LoopBack1
#
  ipv4-family unicast
   undo synchronization
   peer 3.3.3.9 enable
#
  ipv4-family vpnv4
   policy vpn-target
   peer 3.3.3.9 enable
#
  ipv4-family vpn-instance vpn1
   peer 10.1.1.1 as-number 600
   peer 10.1.1.1 substitute-as
   import-route direct
#
 ospf 1
  area 0.0.0.0
   network 1.1.1.9 0.0.0.0
   network 20.1.1.0 0.0.0.255
#
return
```

● P configuration file

```
#
 sysname P
```

```
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address ip address 20.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address ip address 30.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 20.1.1.0 0.0.0.255
network 30.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:2
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 30.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.9 enable
#
ipv4-family vpn-instance vpn1
peer 10.2.1.1 as-number 600
peer 10.2.1.1 substitute-as
import-route direct
```

```
#
ospf 1
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 30.1.1.0 0.0.0.255
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.4.1.1 255.255.255.0
#
bgp 600
 peer 10.2.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.2.1.2 enable
#
return
```

8.9.14 Example for Configuring the BGP SoO Attribute

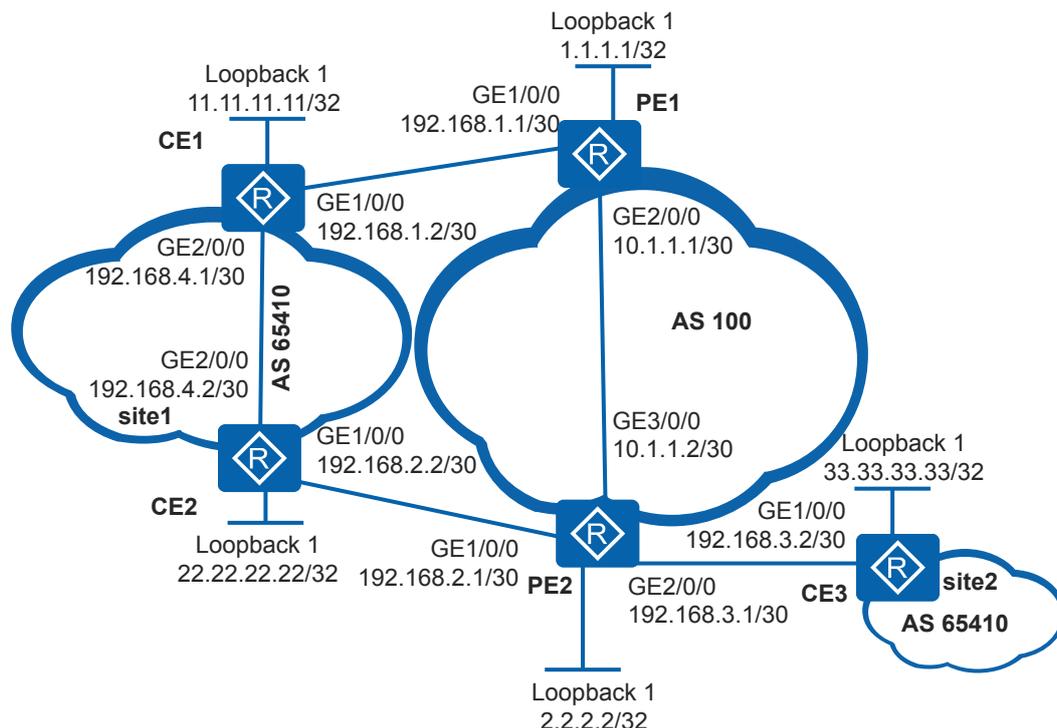
Networking Requirements

When multiple CEs in a VPN site connect to different PEs, VPN routes advertised from the CEs to the PEs may be sent back to the VPN site after the routes traverse the backbone network. This may cause routing loops in the VPN site.

As shown in [Figure 8-55](#), CE1 and CE2 belong to site 1; CE2 and CE3 connect to PE2. Site 1 and site 2 have the same AS number. The PEs and CEs run EBGP. PE1 uses MP-IGBP to advertise the routes learned from CE1 to PE2. Then PE2 advertises these routes to CE2 and CE3. However, CE2 has learned the routes through IGP in site 1. As a result, a routing loop may occur in site 1.

To prevent routing loops in site 1, configure the BGP Site of Origin (SoO) attribute on the PEs. When PE2 advertises routes to CE2, PE2 checks whether the SoO attribute of the routes is the same as the locally configured SoO attribute. If so, PE2 does not advertise these routes to CE2. PE2 can advertise the routes to CE3.

Figure 8-55 Networking diagram for configuring the BGP SoO attribute



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IP address for each interface and an IGP on the backbone network so that PEs can communicate.
2. Enable MPLS and MPLS LDP on the backbone network so that LDP LSPs can be established between the PEs.
3. Set up an MP-IBGP peer relationship between the PEs.
4. Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.
5. Set up EBGP peer relationships between the PEs and CEs and enable AS number substitution on the PEs.
6. Configure the BGP SoO attribute for the connected CEs on the PEs.

Procedure

- Step 1** Configure an IP address for each interface and an IGP on the backbone network so that PEs can learn routes to loopback interfaces of each other.

In this example, OSPF is configured.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.1 32
```

```
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 30
[PE1-GigabitEthernet2/0/0] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.3
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration on PE2 and CEs is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display ip routing-table** command on the PEs. The command output shows that the PEs have learned the route to loopback interfaces of each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 9          Routes : 9

Destination/Mask    Proto    Pre  Cost           Flags NextHop         Interface
-----
      1.1.1.1/32     Direct   0    0              D    127.0.0.1        LoopBack1
      2.2.2.2/32     OSPF     10   1              D    10.1.1.2
GigabitEthernet2/0/0
      10.1.1.0/30    Direct   0    0              D    10.1.1.1
GigabitEthernet2/0/0
      10.1.1.1/32    Direct   0    0              D    127.0.0.1
GigabitEthernet2/0/0
      10.1.1.3/32    Direct   0    0              D    127.0.0.1
GigabitEthernet2/0/0
      127.0.0.0/8     Direct   0    0              D    127.0.0.1        InLoopBack0
      127.0.0.1/32    Direct   0    0              D    127.0.0.1        InLoopBack0
      127.255.255.255/32 Direct   0    0              D    127.0.0.1        InLoopBack0
      255.255.255.255/32 Direct   0    0              D    127.0.0.1        InLoopBack0
```

Step 2 Enable MPLS and MPLS LDP on the backbone network to set up LDP LSPs.

Enable MPLS and MPLS LDP globally and on interfaces of the PE.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2 is the same as the configuration on PE1.

After the configuration is complete, run the **display mpls ldp lsp** command on the PEs. The command output shows the labels assigned to the routes to loopback interfaces on the peer PEs. The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp lsp
LDP LSP Information
```

```

-----
DestAddress/Mask   In/OutLabel      UpstreamPeer     NextHop          OutInterface
-----
1.1.1.1/32        3/NULL           2.2.2.2          127.0.0.1       InLoop0
*1.1.1.1/32       Liberal/1024     -                DS/2.2.2.2
2.2.2.2/32        NULL/3           -                10.1.1.2        GE2/0/0
2.2.2.2/32      1024/3          2.2.2.2          10.1.1.2        GE2/0/0
-----
TOTAL: 3 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
  
```

Step 3 Set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```

[PE1] bgp 100
[PE1-bgp] peer 2.2.2.2 as-number 100
[PE1-bgp] peer 2.2.2.2 connect-interface loopback1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.2 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
  
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here. For configuration details, refer to "Configuration Files".

After the configuration is complete, run the **display bgp peer** or **display bgp vpnv4 all peer** command on the PEs. The command output shows that BGP peer relationships have been established between the PEs. The information displayed on PE1 is used as an example.

```

[PE1] display bgp peer

BGP local router ID : 10.1.1.1
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down State
PrefRcv
2.2.2.2      4      100    187     186     0 02:44:06
Established 1
  
```

Step 4 On each PE, configure a VPN instance, enable the IPv4 address family in the instance, and bind the instance to the interfaces connected to the CEs.

Configure PE1.

```

[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:100
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet1/0/0] ip address 192.168.1.1 30
[PE1-GigabitEthernet1/0/0] quit
  
```

Configure PE2.

```

[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
  
```

```
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 100:100
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet1/0/0] ip address 192.168.2.1 30
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet2/0/0] ip address 192.168.3.1 30
[PE2-GigabitEthernet2/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances.

Step 5 Set up EBGp peer relationships between PEs and CEs, enable AS number substitution on the PEs, and configure PEs to import routes from CEs.

In this configuration example, the two VPN sites have the same AS number. Therefore, AS number substitution needs to be enabled on PE1 and PE2.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] peer 192.168.1.2 as-number 65410
[PE1-bgp-vpna] peer 192.168.1.2 substitute-as
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] quit
[PE1-bgp] quit
```

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 192.168.1.1 as-number 100
[CE1-bgp] network 11.11.11.11 32
[CE1-bgp] network 192.168.4.0 30
[CE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] peer 192.168.2.2 as-number 65410
[PE2-bgp-vpna] peer 192.168.3.2 as-number 65410
[PE2-bgp-vpna] peer 192.168.2.2 substitute-as
[PE2-bgp-vpna] peer 192.168.3.2 substitute-as
[PE2-bgp-vpna] import-route direct
[PE2-bgp-vpna] quit
[PE2-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65410
[CE2-bgp] peer 192.168.2.1 as-number 100
[CE2-bgp] network 22.22.22.22 32
[CE2-bgp] network 192.168.4.0 30
[CE2-bgp] quit
```

Configure CE3.

```
[CE3] bgp 65410
[CE3-bgp] peer 192.168.3.1 as-number 100
[CE3-bgp] network 33.33.33.33 32
[CE3-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance peer** command on the PEs. The command output shows that the status of EBGp peer relationships between

PEs and CEs is Established. This indicates that EBGP peer relationships have been established between PEs and CEs. The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpna peer

BGP local router ID : 10.1.1.1
Local AS number : 100

VPN-Instance vpna, router ID 10.1.1.1:
Total number of peers : 1                Peers in established state : 1

Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down    State
PrefRcv
-----
192.168.1.2   4          65410    224      231     0 03:02:12
Established 1
```

Run the **display bgp vpnv4 routing-table** command on the PEs. The command output shows the routes sent from the PEs to the PEs. The following shows the routes sent from PE2 to CE2.

```
[PE2] display bgp vpnv4 vpn-instance vpna routing-table peer 192.168.2.2
advertised-routes

BGP Local router ID is 2.2.2.2
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete

VPN-Instance vpna, Router ID 2.2.2.2:

Total Number of Routes: 6

Network          NextHop          MED          LocPrf          PrefVal Path/Ogn
-----
*>i 11.11.11.11/32 192.168.2.1          0          100 100i
*> 22.22.22.22/32 192.168.2.1          0          100 100i
*> 33.33.33.33/32 192.168.2.1          0          100 100i
*>i 192.168.1.0/30 192.168.2.1          0          100?
*> 192.168.2.0/30 192.168.2.1          0          100?
*> 192.168.3.0/30 192.168.2.1          0          100?
```

Step 6 Configure the BGP SoO attribute on the PEs.

CE1 and CE2 belong to the same site, so you need to set the same BGP SoO attribute value for the two CEs on PE1 and PE2. PE2 connects to two VPN sites, so you need to set different SoO attribute value for the CEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] peer 192.168.1.2 soo 100:101
[PE1-bgp-vpna] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] peer 192.168.2.2 soo 100:101
[PE2-bgp-vpna] peer 192.168.3.2 soo 100:102
[PE2-bgp-vpna] quit
[PE2-bgp] quit
```

Step 7 Verify the configuration.

After the configuration is complete, run the **display bgp vpnv4 routing-table** command on PE2 again. The command output shows that the routes sent from PE2 to CE2 have changed and the routes sent from PE2 to CE3 remain unchanged.

```
[PE2] display bgp vpnv4 vpn-instance vpna routing-table peer 192.168.2.2
advertised-routes

BGP Local router ID is 2.2.2.2
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

VPN-Instance vpna, Router ID 2.2.2.2:

Total Number of Routes: 4
   Network          NextHop          MED          LocPrf    PrefVal Path/Ogn
* > 33.33.33.33/32   192.168.2.1          0          0        100 100i
*>i 192.168.1.0/30   192.168.2.1          0          0        100?
*> 192.168.2.0/30   192.168.2.1          0          0        100?
*> 192.168.3.0/30   192.168.2.1          0          0        100?

[PE2] display bgp vpnv4 vpn-instance vpna routing-table peer 192.168.3.2
advertised-routes

BGP Local router ID is 2.2.2.2
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

VPN-Instance vpna, Router ID 2.2.2.2:

Total Number of Routes: 6
   Network          NextHop          MED          LocPrf    PrefVal Path/Ogn
*>i 11.11.11.11/32   192.168.3.1          0          0        100 100i
*> 22.22.22.22/32   192.168.3.1          0          0        100 100i
*>i 192.168.1.0/30   192.168.3.1          0          0        100?
*> 192.168.2.0/30   192.168.3.1          0          0        100?
*> 192.168.3.0/30   192.168.3.1          0          0        100?
*> 192.168.4.0/30   192.168.3.1          0          0        100 100i
```

Run the **display bgp vpnv4 routing-table** command on PE2. The command output shows the SoO attribute carried in the routes sent from PE2 to CE3.

```
[PE2] display bgp vpnv4 vpn-instance vpna routing-table 11.11.11.11 32

BGP local router ID : 2.2.2.2
Local AS number : 100

VPN-Instance vpna, Router ID 2.2.2.2:
Paths: 1 available, 1 best, 1 select
BGP routing table entry information of 11.11.11.11/32:
Label information (Received/Applied): 1029/NULL
From: 1.1.1.1 (1.1.1.1)
Route Duration: 00h11m51s
Relay Tunnel Out-Interface: GigabitEthernet3/0/0
Relay token: 0x3d
Original nexthop: 1.1.1.1
Qos information : 0x0
Ext-Community:RT <100 : 100>, SoO <100 : 101>
AS-path 65410, origin igp, MED 0, localpref 100, pref-val 0, valid, internal, best, select, active, pre 255, IGP cost 1
Advertised to such 1 peers:
192.168.3.2
```

The preceding command output shows that after the BGP SoO attribute is configured, the VPN routes received from CEs carry the SoO attribute, and PE2 does not send any route to CE2. This indicates that the configuration of the BGP SoO attribute has taken effect.

----End

Configuration Files

- **CE1 configuration file**

```
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 192.168.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 192.168.4.1 255.255.255.252
#
interface LoopBack1
 ip address 11.11.11.11 255.255.255.255
#
bgp 65410
 peer 192.168.1.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  network 11.11.11.11 255.255.255.255
  network 192.168.4.0 255.255.255.252
  peer 192.168.1.1 enable
#
return
```

- **CE2 configuration file**

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 192.168.2.2 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 192.168.4.2 255.255.255.252
#
interface LoopBack1
 ip address 22.22.22.22 255.255.255.255
#
bgp 65410
 peer 192.168.2.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  network 22.22.22.22 255.255.255.255
  network 192.168.4.0 255.255.255.252
  peer 192.168.2.1 enable
#
return
```

- **PE1 configuration file**

```
#
sysname PE1
#
ip vpn-instance vpna
 ipv4-family
  route-distinguisher 100:1
  vpn-target 100:100 export-extcommunity
  vpn-target 100:100 import-extcommunity
#
```

```
mpls lsr-id 1.1.1.1
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpna
 ip address 192.168.1.1 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.252
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.2 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 2.2.2.2 enable
#
 ipv4-family vpn-instance vpna
  import-route direct
  peer 192.168.1.2 as-number 65410
  peer 192.168.1.2 substitute-as
  peer 192.168.1.2 soo 100:101
#
ospf 1
 area 0.0.0.0
  network 1.1.1.1 0.0.0.0
  network 10.1.1.0 0.0.0.3
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
 ipv4-family
  route-distinguisher 100:2
  vpn-target 100:100 export-extcommunity
  vpn-target 100:100 import-extcommunity
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpna
 ip address 192.168.2.1 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpna
 ip address 192.168.3.1 255.255.255.252
#
interface GigabitEthernet3/0/0
 ip address 10.1.1.2 255.255.255.252
 mpls
 mpls ldp
```

```
#
interface LoopBack1
 ip address 2.2.2.2 255.255.255.255
#
bgp 100
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.1 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
#
 ipv4-family vpn-instance vpna
  import-route direct
  peer 192.168.2.2 as-number 65410
  peer 192.168.2.2 substitute-as
  peer 192.168.2.2 soo 100:101
  peer 192.168.3.2 as-number 65410
  peer 192.168.3.2 substitute-as
  peer 192.168.3.2 soo 100:102
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 10.1.1.0 0.0.0.3
#
return
```

- CE3 configuration file

```
#
 sysname CE3
#
interface GigabitEthernet1/0/0
 ip address 192.168.3.2 255.255.255.252
#
interface LoopBack1
 ip address 33.33.33.33 255.255.255.255
#
bgp 65410
 peer 192.168.3.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  network 33.33.33.33 255.255.255.255
  peer 192.168.3.1 enable
#
return
```

8.9.15 Example for Configuring CE Dual-homing

Networking Requirements

It is a trend to transmit all telecommunication services on an IP network. Key services such as 3G/NGN, IPTV streaming media, and VPN services require high reliability on networks. In addition to improving the reliability of the network devices, you can improve the link reliability by configuring fast route convergence, fault detection, fast reroute, and route backup.

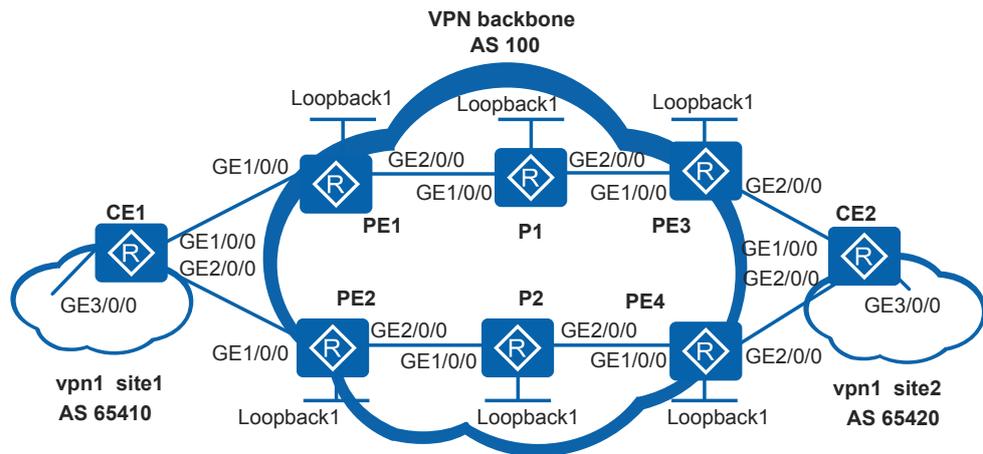
On the access layer, the CE dual-homing networking is a common method to improve the network reliability. A dual-homed CE connects to two PEs that belong to the same VPN as the

CE. In this networking, the CE connects to the backbone network through two links. The two links work in load balancing mode or active/standby mode.

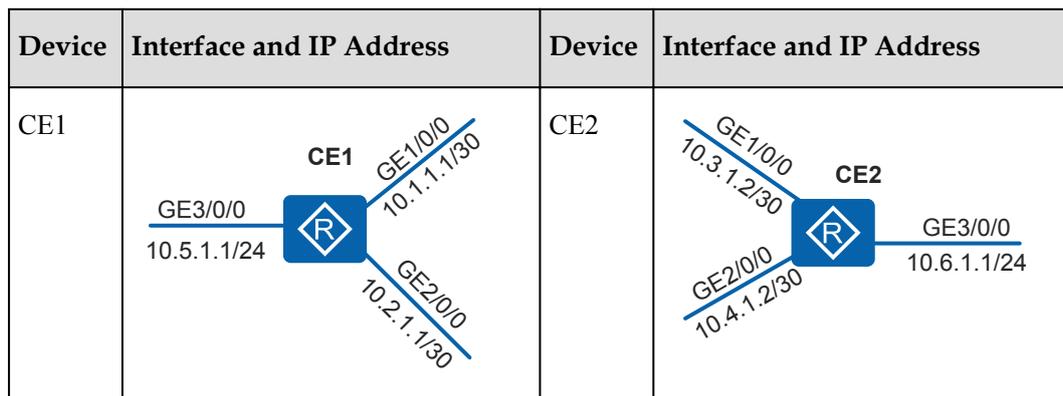
As shown in **Figure 8-56**, CE1 is located in site1 of vpn1, and CE2 is located in site2 of vpn1. CE1 connects to PE1 and PE2, and CE2 connects to PE3 and PE4.

If the data traffic volume from CE1 to CE2 is large but traffic volume from CE2 to CE1 is small, the data traffic from CE1 to CE2 can be transmitted in load balancing mode. The data traffic from CE2 to CE1 is transmitted through PE4, and PE3 only works as a backup.

Figure 8-56 Networking diagram for configuring CE dual-homing



Device	Interface and IP Address	Device	Interface and IP Address
PE1	Loopback1 1.1.1.1/32 GE1/0/0 10.1.1.2/30 GE2/0/0 100.1.1.1/30 PE1	PE2	PE2 GE1/0/0 10.2.1.2/30 GE2/0/0 100.2.1.1/30 Loopback1 2.2.2.2/32
P1	Loopback1 5.5.5.5/32 GE1/0/0 100.1.1.2/30 GE2/0/0 100.3.1.1/30 P1	P2	P2 GE1/0/0 100.2.1.2/30 GE2/0/0 100.4.1.1/30 Loopback1 6.6.6.6/32
PE3	Loopback1 3.3.3.3/32 GE1/0/0 100.3.1.2/30 GE2/0/0 10.3.1.1/30 PE3	PE4	Loopback1 4.4.4.4/32 GE1/0/0 100.4.1.2/30 GE2/0/0 10.4.1.1/30 PE4



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP/MPLS IP VPN functions.
2. In the BGP view of CE1, configure load balancing for traffic sent to CE2.
3. Increase the MED value of the BGP-VPN route on PE3 to ensure that the next hop of the route selected by CE2 to the customer network connected to CE1 is PE4.

Procedure

Step 1 Configure an IGP on the MPLS backbone network so that PEs and Ps can communicate with each other.

Configure PE1.

Set IP addresses of interfaces. The IP addresses of the loopback interfaces must use a 32-bit mask.

```
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.1 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 100.1.1.1 255.255.255.252
[PE1-GigabitEthernet2/0/0] quit
```

Configure the ISIS protocol to advertise routes of the interfaces.

```
[PE1] isis 1
[PE1-isis-1] network-entity 10.0000.0000.0001.00
[PE1-isis-1] quit
[PE1] interface loopback 1
[PE1-LoopBack1] isis enable 1
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] isis enable 1
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2, PE3, PE4, P1, and P2 is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display ip routing-table** command. The command output shows that PE1 and PE3 can learn the routes of Loopback1 interface of each other; PE2 and PE4 can learn routes of Loopback1 interface of each other.

Step 2 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.

Configure PE1.

Enable MPLS and LDP in the system view, set the LSR ID to the IP address of the loopback interface, and trigger the LSP.

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
```

Enable MPLS and LDP on the interface connected to the backbone network.

```
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2, PE3, PE4, P1, and P2 is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, LDP sessions can be set up between PE1 and P1, and between PE3 and P1. Run the **display mpls ldp session** command. The command output shows that the status of the sessions is Operational. Run the **display mpls ldp lsp** command. Information about the established LDP LSPs is displayed.

The information displayed on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
5.5.5.5:0              Operational DU   Passive 0000:07:02 1688/1688
-----
TOTAL: 1 session(s) Found.

[PE1] display mpls ldp lsp

LDP LSP Information
-----
DestAddress/Mask      In/OutLabel  UpstreamPeer  NextHop      OutInterface
-----
1.1.1.1/32            3/NULL      5.5.5.5       127.0.0.1    InLoop0
*1.1.1.1/32           Liberal/1024
3.3.3.3/32           NULL/1025   -             100.1.1.2    GE2/0/0
3.3.3.3/32           1025/1025  5.5.5.5       100.1.1.2    GE2/0/0
5.5.5.5/32           NULL/3      -             100.1.1.2    GE2/0/0
5.5.5.5/32           1024/3      5.5.5.5       100.1.1.2    GE2/0/0
-----
TOTAL: 5 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
```

Step 3 Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

Configure PE1.

Create a VPN instance and set the RD and VPN target of the VPN instance. The VPN target set on the local PE must match the VPN target of the MP-BGP peer PE so that the sites of the same VPN can communicate with each other.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
```

Bind the VPN instance to the interface connected to the CE and set the IP address of the interface.

```
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 255.255.255.252
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

Create a VPN instance and set the RD and VPN target of the VPN instance. The VPN target set on the local PE must match the VPN target of the MP-BGP peer PE so that the sites of the same VPN can communicate with each other.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
```

Bind the VPN instance to the interface connected to the CE and set the IP address of the interface.

```
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet1/0/0] ip address 10.2.1.2 255.255.255.252
[PE2-GigabitEthernet1/0/0] quit
```

Configure PE3.

Create a VPN instance and set the RD and VPN target of the VPN instance. The VPN target set on the local PE must match the VPN target of the MP-BGP peer PE so that the sites of the same VPN can communicate with each other.

```
[PE3] ip vpn-instance vpn1
[PE3-vpn-instance-vpn1] ipv4-family
[PE3-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:3
[PE3-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE3-vpn-instance-vpn1-af-ipv4] quit
[PE3-vpn-instance-vpn1] quit
```

Bind the VPN instance to the interface connected to the CE and set the IP address of the interface.

```
[PE3] interface gigabitethernet 2/0/0
[PE3-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE3-GigabitEthernet2/0/0] ip address 10.3.1.1 255.255.255.252
[PE3-GigabitEthernet2/0/0] quit
```

Configure PE4.

Create a VPN instance and set the RD and VPN target of the VPN instance. The VPN target set on the local PE must match the VPN target of the MP-BGP peer PE so that the sites of the same VPN can communicate with each other.

```
[PE4] ip vpn-instance vpn1
[PE4-vpn-instance-vpn1] ipv4-family
[PE4-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:4
[PE4-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE4-vpn-instance-vpn1-af-ipv4] quit
[PE4-vpn-instance-vpn1] quit
```

Bind the VPN instance to the interface connected to the CE and set the IP address of the interface.

```
[PE4] interface gigabitEthernet 2/0/0
[PE4-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE4-GigabitEthernet2/0/0] ip address 10.4.1.1 255.255.255.252
[PE4-GigabitEthernet2/0/0] quit
```

Assign IP addresses to interfaces on CEs according to [Figure 8-56](#).

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitEthernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
[CE1] interface gigabitEthernet 2/0/0
[CE1-GigabitEthernet2/0/0] ip address 10.2.1.1 24
[CE1-GigabitEthernet2/0/0] quit
[CE1] interface gigabitEthernet 3/0/0
[CE1-GigabitEthernet3/0/0] ip address 10.5.1.1 24
[CE1-GigabitEthernet3/0/0] quit
```

The configuration on other CEs is similar to the configuration on Spoke-CE1 and is not mentioned here.

After the configuration is complete, run the **display ip vpn-instance verbose** command on the PEs to check the configuration of VPN instances.

The information displayed on PE1 is used as an example.

```
[PE1] display ip vpn-instance verbose
Total VPN-Instances configured : 1
Total IPv4 VPN-Instances configured : 1
Total IPv6 VPN-Instances configured : 0

VPN-Instance Name and ID : vpn1, 1
  Interfaces : GigabitEthernet1/0/0
Address family ipv4
Create date : 2012/07/25 00:58:17
Up time : 0 days, 17 hours, 38 minutes and 53 seconds
Route Distinguisher : 100:1
Export VPN Targets : 1:1
Import VPN Targets : 1:1
Label Policy : label per route
Log Interval : 5
```

Step 4 Set up MP-IBGP peer relationships between the PEs.

Configure PE1.

Specify PE3 as the IGBP peer and use the IP address of the loopback interface to set up an IBGP connection with the peer.

```
[PE1] bgp 100  
[PE1-bgp] peer 3.3.3.3 as-number 100  
[PE1-bgp] peer 3.3.3.3 connect-interface loopback 1
```

Enter the VPNv4 address family view and enable the local PE to exchange VPN routing information with the IGBP peer.

```
[PE1-bgp] ipv4-family vpnv4  
[PE1-bgp-af-vpnv4] peer 3.3.3.3 enable  
[PE1-bgp-af-vpnv4] quit  
[PE1-bgp] quit
```

Configure PE3.

Specify PE1 as the IGBP peer and use the IP address of the loopback interface to set up an IGBP connection with the peer.

```
[PE3] bgp 100  
[PE3-bgp] peer 1.1.1.1 as-number 100  
[PE3-bgp] peer 1.1.1.1 connect-interface loopback 1
```

Enter the VPNv4 address family view and enable the local PE to exchange VPN routing information with the IGBP peer.

```
[PE3-bgp] ipv4-family vpnv4  
[PE3-bgp-af-vpnv4] peer 1.1.1.1 enable  
[PE3-bgp-af-vpnv4] quit  
[PE3-bgp] quit
```

Configure PE2.

Specify PE4 as the IGBP peer and use the IP address of the loopback interface to set up an IGBP connection with the peer.

```
[PE2] bgp 100  
[PE2-bgp] peer 4.4.4.4 as-number 100  
[PE2-bgp] peer 4.4.4.4 connect-interface loopback 1
```

Enter the VPNv4 address family view and enable the local PE to exchange VPN routing information with the IGBP peer.

```
[PE2-bgp] ipv4-family vpnv4  
[PE2-bgp-af-vpnv4] peer 4.4.4.4 enable  
[PE2-bgp-af-vpnv4] quit  
[PE2-bgp] quit
```

Configure PE4.

Specify PE2 as the IGBP peer and use the IP address of the loopback interface to set up an IGBP connection with the peer.

```
[PE4] bgp 100  
[PE4-bgp] peer 2.2.2.2 as-number 100  
[PE4-bgp] peer 2.2.2.2 connect-interface loopback 1
```

Enter the VPNv4 address family view and enable the local PE to exchange VPN routing information with the IGBP peer.

```
[PE4-bgp] ipv4-family vpnv4  
[PE4-bgp-af-vpnv4] peer 2.2.2.2 enable  
[PE4-bgp-af-vpnv4] quit  
[PE4-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and are in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 all peer
BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
3.3.3.3       4   100      70       81     0 01:00:23  Established    3
```

Step 5 Configure EBGP between the PE and the CEs to import the VPN routes.

Configure CE1.

Enable BGP, specify PE1 and PE2 as EBGP peers, and import direct routes.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] peer 10.2.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1.

Enable BGP, specify CE1 as the EBGP peer, and import direct routes.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure PE2.

Enable BGP, specify CE1 as the EBGP peer, and import direct routes.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 as-number 65410
[PE2-bgp-vpn1] import-route direct
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

Configure CE2.

Enable BGP, specify PE3 and PE4 as EBGP peers, and import direct routes.

```
[CE2] bgp 65420
[CE2-bgp] peer 10.3.1.1 as-number 100
[CE2-bgp] peer 10.4.1.1 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE3.

Enable BGP, specify CE2 as the EBGP peer, and import direct routes.

```
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpn1
[PE3-bgp-vpn1] peer 10.3.1.2 as-number 65420
[PE3-bgp-vpn1] import-route direct
[PE3-bgp-vpn1] quit
[PE3-bgp] quit
```

Configure PE4.

Enable BGP, specify CE2 as the EBGP peer, and import direct routes.

```
[PE4] bgp 100
[PE4-bgp] ipv4-family vpn-instance vpn1
[PE4-bgp-vpn1] peer 10.4.1.2 as-number 65420
[PE4-bgp-vpn1] import-route direct
[PE4-bgp-vpn1] quit
[PE4-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 vpn-instance vpn-instancename peer** command on the PEs. The command output shows that the BGP peer relationships have been set up between the PEs and CEs and are in Established state. Each PE can ping its connected CE.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.1
Local AS number : 100

VPN-Instance vpn1, Router ID 1.1.1.1:
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
-----
10.1.1.1      4 65410    408     435     0 06:16:09 Established      5

[PE1] ping -vpn-instance vpn1 10.1.1.1
PING 10.1.1.1 : 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=80 ms
  Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=20 ms
  Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=30 ms
  Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=50 ms
  Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=30 ms

--- 10.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 20/42/80 ms
```

Step 6 On CE1, configure load balancing for the traffic sent from CE1 to CE2.

```
[CE1] bgp 65410
[CE1-bgp] ipv4-family unicast
[CE1-bgp-af-ipv4] maximum load-balancing 2
[CE1-bgp-af-ipv4] quit
[CE1-bgp] quit
```

Step 7 Configure a routing policy on PE3 to increase the MED value of the BGP routes advertised to CE2. Then the traffic sent from CE2 to CE1 is forwarded by PE4, and PE3 is a backup of PE4.

```
[PE3] route-policy policyl1 permit node 10
[PE3-route-policy] apply cost 120
[PE3-route-policy] quit
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpn1
[PE3-bgp-vpn1] peer 10.3.1.2 route-policy policyl1 export
[PE3-bgp-vpn1] quit
[PE3-bgp] quit
```

Check the BGP routing table on CE2. In the routing table, the route to 10.5.1.0/30 advertised by PE3 has a MED value of 120, larger than the MED value of the route advertised by PE4 (the default MED value is 0). Therefore, CE2 selects the route advertised by PE4.

```
[CE2] display bgp routing-table

Total Number of Routes: 11
BGP Local router ID is 10.2.1.1
```

```

Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete

```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.5.1.0/24	10.4.1.1			0	100 65410?
*		10.3.1.1	120		0	100 65410?
*>	10.6.1.0/24	0.0.0.0	0		0	?
*>	10.1.1.0/30	10.3.1.1	120		0	100?
*		10.4.1.1			0	100 65410?
*>	10.2.1.0/30	10.4.1.1			0	100?
*		10.3.1.1	120		0	100 65410?
*>	10.3.1.0/30	0.0.0.0	0		0	?
*		10.3.1.1	120		0	100?
*>	10.4.1.0/30	0.0.0.0	0		0	?
*		10.4.1.1	0		0	100?
*>	127.0.0.0	0.0.0.0	0		0	?
*>	127.0.0.1/30	0.0.0.0	0		0	?

Step 8 Verify the configuration.

#If the configuration is successful:

#The **display ip routing-table** command on CE1 displays the routes to the customer network connected to CE2. The routes work in load balancing mode.

```

[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
Destinations : 16      Routes : 17

```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.5.1.0/24	Direct	0	0	D	10.5.1.1	
GigabitEthernet3/0/0						
10.5.1.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet3/0/0						
10.5.1.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet3/0/0						
10.6.1.0/24	EBGP	255	0	D	10.1.1.2	
GigabitEthernet1/0/0						
	EBGP	255	0	D	10.2.1.2	
GigabitEthernet2/0/0						
10.1.1.0/30	Direct	0	0	D	10.1.1.1	
GigabitEthernet1/0/0						
10.1.1.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet1/0/0						
10.1.1.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet1/0/0						
10.2.1.0/30	Direct	0	0	D	10.2.1.1	
GigabitEthernet2/0/0						
10.2.1.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet2/0/0						
10.2.1.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet2/0/0						
10.3.1.0/30	EBGP	255	0	D	10.1.1.2	
GigabitEthernet1/0/0						
10.4.1.0/30	EBGP	255	0	D	10.2.1.2	
GigabitEthernet2/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0
255.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0

The **display ip routing-table** command on CE2 displays the routes to the customer network connected to CE1. The next hop of the route is 10.4.1.1, IP address of the interface that connects PE4 to CE2.

```
[CE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 16          Routes : 16

Destination/Mask    Proto  Pre  Cost           Flags NextHop         Interface
-----
      10.5.1.0/24    EGBP   255  0              D   10.4.1.1
GigabitEthernet2/0/0
      10.6.1.0/24    Direct  0     0              D   10.6.1.1
GigabitEthernet3/0/0
      10.6.1.1/32    Direct  0     0              D   127.0.0.1
GigabitEthernet3/0/0
      10.6.1.255/32  Direct  0     0              D   127.0.0.1
GigabitEthernet3/0/0
      10.1.1.0/30    EGBP   255  120           D   10.3.1.1
GigabitEthernet1/0/0
      10.2.1.0/30    EGBP   255  0              D   10.4.1.1
GigabitEthernet2/0/0
      10.3.1.0/30    Direct  0     0              D   10.3.1.2
GigabitEthernet1/0/0
      10.3.1.2/32    Direct  0     0              D   127.0.0.1
GigabitEthernet1/0/0
      10.3.1.3/32    Direct  0     0              D   127.0.0.1
GigabitEthernet1/0/0
      10.4.1.0/30    Direct  0     0              D   10.4.1.2
GigabitEthernet2/0/0
      10.4.1.2/32    Direct  0     0              D   127.0.0.1
GigabitEthernet2/0/0
      10.4.1.3/32    Direct  0     0              D   127.0.0.1
GigabitEthernet2/0/0
      127.0.0.0/8     Direct  0     0              D   127.0.0.1      InLoopBack0
      127.0.0.1/32   Direct  0     0              D   127.0.0.1      InLoopBack0
127.255.255.255/32 Direct  0     0              D   127.0.0.1      InLoopBack0
255.255.255.255/32 Direct  0     0              D   127.0.0.1      InLoopBack0
```

---End

Configuration Files

- CE1 configuration file

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.252
#
interface GigabitEthernet2/0/0
ip address 10.2.1.1 255.255.255.252
#
interface GigabitEthernet3/0/0
ip address 10.5.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
peer 10.2.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
maximum load-balancing 2
peer 10.1.1.2 enable
peer 10.2.1.2 enable
#
return
```

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:1
    vpn-target 1:1 export-extcommunity
    vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.1
mpls
#
mpls ldp
#
isis 1
  network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn1
  ip address 10.1.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
  ip address 100.1.1.1 255.255.255.252
  isis enable 1
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.1 255.255.255.255
  isis enable 1
#
bgp 100
  peer 3.3.3.3 as-number 100
  peer 3.3.3.3 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 3.3.3.3 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 3.3.3.3 enable
#
  ipv4-family vpn-instance vpn1
    peer 10.1.1.1 as-number 65410
    import-route direct
#
return
```

- PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:2
    vpn-target 1:1 export-extcommunity
    vpn-target 1:1 import-extcommunity
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
isis 1
  network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
```

```
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
ip address 100.2.1.1 255.255.255.252
isis enable 1
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.2 255.255.255.255
isis enable 1
#
bgp 100
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 4.4.4.4 enable
#
ipv4-family vpnv4
policy vpn-target
peer 4.4.4.4 enable
#
ipv4-family vpn-instance vpn1
peer 10.2.1.1 as-number 65410
import-route direct
#
return
```

● P1 configuration file

```
#
sysname P1
#
mpls lsr-id 5.5.5.5
mpls
#
mpls ldp
#
isis 1
network-entity 10.0000.0000.0005.00
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.252
isis enable 1
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 100.3.1.1 255.255.255.252
isis enable 1
mpls
mpls ldp
#
interface LoopBack1
ip address 5.5.5.5 255.255.255.255
isis enable 1
#
return
```

● P2 configuration file

```
#
sysname P2
#
mpls lsr-id 6.6.6.6
mpls
#
mpls ldp
#
```

```
isis 1
 network-entity 10.0000.0000.0006.00
 #
 interface GigabitEthernet1/0/0
  ip address 100.2.1.2 255.255.255.252
  isis enable 1
  mpls
  mpls ldp
 #
 interface GigabitEthernet2/0/0
  ip address 100.4.1.1 255.255.255.252
  isis enable 1
  mpls
  mpls ldp
 #
 interface LoopBack1
  ip address 6.6.6.6 255.255.255.255
  isis enable 1
 #
 return
```

● PE3 configuration file

```
#
sysname PE3
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:3
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
 #
 mpls lsr-id 3.3.3.3
 mpls
 #
 mpls ldp
 #
 isis 1
  network-entity 10.0000.0000.0003.00
 #
 interface GigabitEthernet1/0/0
  ip address 100.3.1.2 255.255.255.252
  isis enable 1
  mpls
  mpls ldp
 #
 interface GigabitEthernet2/0/0
  ip binding vpn-instance vpn1
  ip address 10.3.1.1 255.255.255.252
 #
 interface LoopBack1
  ip address 3.3.3.3 255.255.255.255
  isis enable 1
 #
 bgp 100
  peer 1.1.1.1 as-number 100
  peer 1.1.1.1 connect-interface LoopBack1
 #
  ipv4-family unicast
  undo synchronization
  peer 1.1.1.1 enable
 #
  ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
 #
  ipv4-family vpn-instance vpn1
  peer 10.3.1.2 as-number 65420
  peer 10.3.1.2 route-policy policy1 export
  import-route direct
 #
```

```
route-policy policy1 permit node 10
  apply cost 120
#
return
```

- PE4 configuration file

```
#
sysname PE4
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:4
    vpn-target 1:1 export-extcommunity
    vpn-target 1:1 import-extcommunity
#
mpls lsr-id 4.4.4.4
mpls
#
mpls ldp
#
isis 1
  network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
  ip address 100.4.1.2 255.255.255.252
  isis enable 1
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpn1
  ip address 10.4.1.1 255.255.255.252
#
interface LoopBack1
  ip address 4.4.4.4 255.255.255.255
  isis enable 1
#
bgp 100
  peer 2.2.2.2 as-number 100
  peer 2.2.2.2 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 2.2.2.2 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 2.2.2.2 enable
#
  ipv4-family vpn-instance vpn1
    peer 10.4.1.2 as-number 65420
    import-route direct
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
  ip address 10.3.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
  ip address 10.4.1.2 255.255.255.252
#
interface GigabitEthernet3/0/0
  ip address 10.6.1.1 255.255.255.0
#
bgp 65420
  peer 10.3.1.1 as-number 100
```

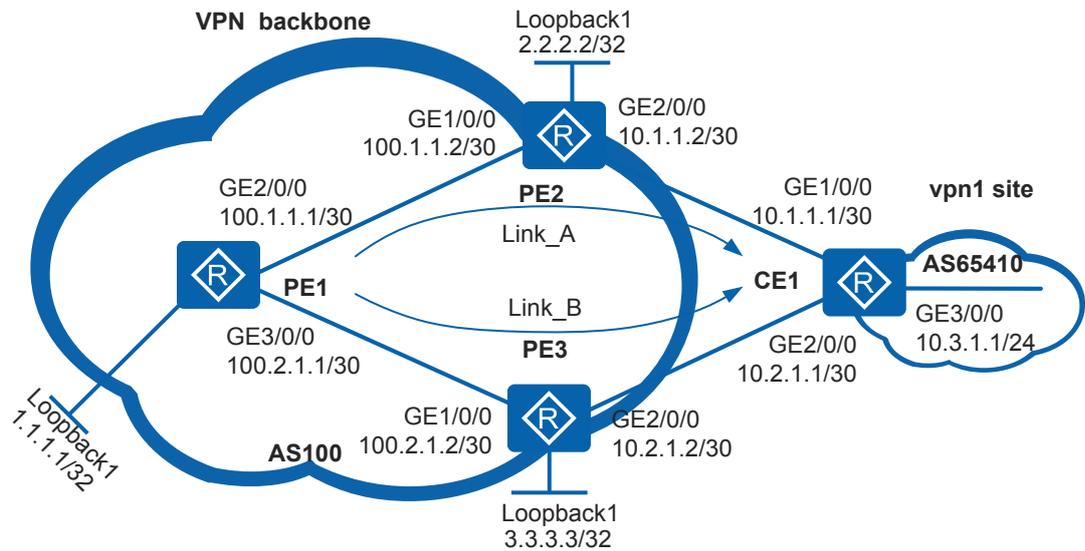
```
peer 10.4.1.1 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.3.1.1 enable
peer 10.4.1.1 enable
#
return
```

8.9.16 Example for Configuring VPN FRR

Networking Requirements

As shown in [Figure 8-57](#), CE1 dual-homing networking is deployed to improve reliability of VPN data transmission. Link_A is the primary link, and Link_B is the backup link. The customer wants to transmit VPN services through the primary link and hopes that VPN traffic can be quickly switched to the backup link when the primary link fails.

Figure 8-57 Networking diagram for configuring VPN FRR



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on PE1, PE2, and PE3 to implement interworking on the backbone network.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.
3. Configure a VPN instance on PE1, PE2, and PE3. On PE2 and PE3, bind the VPN instance to the interfaces connected to CE1.
4. Set up EBGP peer relationships between PE2 and CE1 and between PE3 and CE1. Set up MP-IBGP peer relationships between the PEs.

5. On PE1, configure a routing policy for VPN FRR, configure the backup next hop, and enable VPN FRR. When VPN FRR is not required, run the **undo vpn frr** command to disable this function.
6. Configure multi-hop BFD on PE1 and PE2.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-57](#).

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.1 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 100.1.1.1 30
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip address 100.2.1.1 30
[PE1-GigabitEthernet3/0/0] quit
```

The configuration on PE2, PE3, and CE1 is similar to the configuration on PE1 and is not mentioned here.

Step 2 Configure OSPF on the MPLS backbone network for IP connectivity between the PEs on the backbone network.

Configure PE1.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 100.1.1.0 0.0.0.3
[PE1-ospf-1-area-0.0.0.0] network 100.2.1.0 0.0.0.3
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration on PE2 and PE3 is similar to the configuration on PE1 and is not mentioned here.

Step 3 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls
[PE1-GigabitEthernet3/0/0] mpls ldp
[PE1-GigabitEthernet3/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 2.2.2.2
```

```
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

Configure PE3.

```
[PE3] mpls lsr-id 3.3.3.3
[PE3] mpls
[PE3-mpls] quit
[PE3] mpls ldp
[PE3-mpls-ldp] quit
[PE3] interface gigabitethernet 1/0/0
[PE3-GigabitEthernet1/0/0] mpls
[PE3-GigabitEthernet1/0/0] mpls ldp
[PE3-GigabitEthernet1/0/0] quit
```

Run the **display mpls lsp** command on the PEs. The command output shows that LSPs are established between PE1 and PE2 and between PE1 and PE3. The information displayed on PE1 is used as an example.

```
[PE1] display mpls lsp
-----
LSP Information: LDP LSP
-----
FEC                In/Out Label      In/Out IF          Vrf Name
1.1.1.1/32         3/NULL           -/-
3.3.3.3/32         NULL/3           -/GE3/0/0
3.3.3.3/32         1025/3           -/GE3/0/0
2.2.2.2/32         NULL/3           -/GE2/0/0
2.2.2.2/32         1024/3           -/GE2/0/0
```

Step 4 Configure VPN instances on PEs and bind the instances to the interfaces connected to CE1.

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet2/0/0] ip address 10.1.1.2 30
[PE2-GigabitEthernet2/0/0] quit
```

Configure PE3.

```
[PE3] ip vpn-instance vpn1
[PE3-vpn-instance-vpn1] ipv4-family
[PE3-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:3
[PE3-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE3-vpn-instance-vpn1-af-ipv4] quit
[PE3-vpn-instance-vpn1] quit
[PE3] interface gigabitethernet 2/0/0
```

```
[PE3-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE3-GigabitEthernet2/0/0] ip address 10.2.1.2 30
[PE3-GigabitEthernet2/0/0] quit
```

Step 5 Import direct VPN routes to PE1. Set up EBGP peer relationships between PE2 and CE1 and between PE3 and CE1 to import VPN routes.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE2-bgp-vpn1] import-route direct
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

Configure PE3.

```
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpn1
[PE3-bgp-vpn1] peer 10.2.1.1 as-number 65410
[PE3-bgp-vpn1] import-route direct
[PE3-bgp-vpn1] quit
[PE3-bgp] quit
```

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] peer 10.2.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] network 10.3.1.0 24
[CE1-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on PE2 and PE3. The command output shows that PE2 and PE3 have set up EBGP peer relationships with CE1. The peer relationships are in Established state.

The information displayed on PE2 is used as an example.

```
[PE2] display bgp vpnv4 all peer

BGP local router ID : 2.2.2.2
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down    State
PrefRcv

Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 2.2.2.2:
10.1.1.1      4          65410    966      968      0 16:01:19
Established   5
```

Step 6 Set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.2 as-number 100
```

```
[PE1-bgp] peer 2.2.2.2 connect-interface loopback 1
[PE1-bgp] peer 3.3.3.3 as-number 100
[PE1-bgp] peer 3.3.3.3 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.2 enable
[PE1-bgp-af-vpnv4] peer 3.3.3.3 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.1 as-number 100
[PE2-bgp] peer 1.1.1.1 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.1 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

Configure PE3.

```
[PE3] bgp 100
[PE3-bgp] peer 1.1.1.1 as-number 100
[PE3-bgp] peer 1.1.1.1 connect-interface loopback 1
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpnv4] peer 1.1.1.1 enable
[PE3-bgp-af-vpnv4] quit
[PE3-bgp] quit
```

Run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that an MP-IBGP peer relationship has been set up between the PEs and is in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

Peer          V    AS  MsgRcvd  MsgSent   OutQ  Up/Down      State PrefRcv
-----
2.2.2.2       4   100     20      17        0  00:13:26  Established    5
3.3.3.3       4   100     24      19        0  00:17:18  Established    5
```

Step 7 Configure the VPN FRR routing policy.

```
[PE1] ip ip-prefix vpn_frr_list permit 2.2.2.2 32
[PE1] route-policy vpn_frr_rp permit node 10
[PE1-route-policy] if-match ip next-hop ip-prefix vpn_frr_list
[PE1-route-policy] apply backup-nexthop 3.3.3.3
[PE1-route-policy] quit
```

Step 8 Configure multi-hop BFD.

Configure multi-hop BFD on PE1.

```
[PE1] bfd
[PE1-bfd] quit
[PE1] bfd for_vpn_frr bind peer-ip 2.2.2.2
[PE1-bfd-session-for_vpn_frr] discriminator local 10
[PE1-bfd-session-for_vpn_frr] discriminator remote 20
[PE1-bfd-session-for_vpn_frr] min-tx-interval 100
[PE1-bfd-session-for_vpn_frr] min-rx-interval 100
[PE1-bfd-session-for_vpn_frr] commit
[PE1-bfd-session-for_vpn_frr] quit
```

Configure multi-hop BFD on PE2.

```
[PE2] bfd
[PE2-bfd] quit
```

```
[PE2] bfd for_vpn_frr bind peer-ip 1.1.1.1
[PE2-bfd-session-for_vpn_frr] discriminator local 20
[PE2-bfd-session-for_vpn_frr] discriminator remote 10
[PE2-bfd-session-for_vpn_frr] min-tx-interval 100
[PE2-bfd-session-for_vpn_frr] min-rx-interval 100
[PE2-bfd-session-for_vpn_frr] commit
[PE2-bfd-session-for_vpn_frr] quit
```

After the configuration is complete, run the **display bfd session all verbose** command on PE1 and PE2. The command output shows that a multi-hop BFD session is established and the status of the BFD session is Up.

Step 9 Enable VPN FRR.

Enable VPN FRR on PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] vpn frr route-policy vpn_frr_rp
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
```

Step 10 Verify the configuration.

Check the backup next hop, backup label, and backup tunnel ID on PE1.

```
[PE1] display ip routing-table vpn-instance vpn1 10.3.1.0 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Table : vpn1
Summary Count : 1

Destination: 10.3.1.0/24
  Protocol: IBGP                Process ID: 0
  Preference: 255                Cost: 0
    NextHop: 2.2.2.2            Neighbour: 2.2.2.2
      State: Active Adv Relied    Age: 00h15m06s
      Tag: 0                      Priority: low
      Label: 15361                QoSInfo: 0x0
  IndirectID: 0x13
  RelayNextHop: 100.1.1.2        Interface: GigabitEthernet2/0/0
  TunnelID: 0x31                 Flags: RD
    BkNextHop: 3.3.3.3          BkInterface: GigabitEthernet3/0/0
    BkLabel: 15362              SecTunnelID: 0x0
  BkPETunnelID: 0x32            BkPESecTunnelID: 0x0
  BkIndirectID: 0x15
```

Run the **shutdown** command on GE1/0/0 of PE2 to simulate a link failure.

```
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] shutdown
[PE2-GigabitEthernet1/0/0] quit
```

Run the **display ip routing-table vpn-instance** command on the PE1 again. The command output shows that the next hop of the route to 10.3.1.0/24 is 3.3.3.3.

```
[PE1] display ip routing-table vpn-instance vpn1 10.3.1.0 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Table : vpn1
Summary Count : 1

Destination: 10.3.1.0/24
  Protocol: IBGP                Process ID: 0
  Preference: 255                Cost: 0
    NextHop: 3.3.3.3            Neighbour: 3.3.3.3
```

```
State: Active Adv Relied      Age: 00h15m06s
Tag: 0                        Priority: low
Label: 15362                  QoSInfo: 0x0
IndirectID: 0x15
RelayNextHop: 100.2.1.2      Interface: GigabitEthernet3/0/0
TunnelID: 0x32               Flags: RD
```

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:1
    vpn frr route-policy vpn_frr_rp
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
mpls lsr-id 1.1.1.1
mpls
#
mpls ldp
#
interface GigabitEthernet2/0/0
  ip address 100.1.1.1 255.255.255.252
  mpls
  mpls ldp
#
interface GigabitEthernet3/0/0
  ip address 100.2.1.1 255.255.255.252
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.1 255.255.255.255
#
bfd for_vpn_frr bind peer-ip 2.2.2.2
  discriminator local 10
  discriminator remote 20
  min-tx-interval 100
  min-rx-interval 100
  commit
#
bgp 100
  peer 2.2.2.2 as-number 100
  peer 2.2.2.2 connect-interface LoopBack1
  peer 3.3.3.3 as-number 100
  peer 3.3.3.3 connect-interface LoopBack1
#
  ipv4-family unicast
    undo synchronization
    peer 2.2.2.2 enable
    peer 3.3.3.3 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 2.2.2.2 enable
    peer 3.3.3.3 enable
#
  ipv4-family vpn-instance vpn1
    import-route direct
#
ospf 1
  area 0.0.0.0
```

```
network 100.1.1.0 0.0.0.3
network 100.2.1.0 0.0.0.3
network 1.1.1.1 0.0.0.0
#
ip ip-prefix vpn_frr_list index 10 permit 2.2.2.2 32
#
route-policy vpn_frr_rp permit node 10
if-match ip next-hop ip-prefix vpn_frr_list
apply backup-nexthop 3.3.3.3
#
return
```

- PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:2
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.252
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpn1
ip address 10.1.1.2 255.255.255.252
#
interface LoopBack1
ip address 2.2.2.2 255.255.255.255
#
bfd for_vpn_frr bind peer-ip 1.1.1.1
discriminator local 20
discriminator remote 10
min-tx-interval 100
min-rx-interval 100
commit
#
bgp 100
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.1 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.1 enable
#
ipv4-family vpn-instance vpn1
peer 10.1.1.1 as-number 65410
import-route direct
#
ospf 1
area 0.0.0.0
network 100.1.1.0 0.0.0.3
network 2.2.2.2 0.0.0.0
#
return
```

- PE3 configuration file

```
#
 sysname PE3
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:3
  vpn-target 111:1 export-extcommunity
  vpn-target 111:1 import-extcommunity
#
 mpls lsr-id 3.3.3.3
 mpls
#
 mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 100.2.1.2 255.255.255.252
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 10.2.1.2 255.255.255.252
#
interface LoopBack1
 ip address 3.3.3.3 255.255.255.255
#
 bgp 100
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.1 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.2.1.1 as-number 65410
  import-route direct
#
 ospf 1
 area 0.0.0.0
 network 100.2.1.0 0.0.0.3
 network 3.3.3.3 0.0.0.0
#
Return
```

● CE1 configuration file

```
#
 sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 10.2.1.1 255.255.255.252
#
interface GigabitEthernet3/0/0
 ip address 10.3.1.1 255.255.255.0
#
 bgp 65410
 peer 10.1.1.2 as-number 100
 peer 10.2.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  network 10.3.1.0 255.255.255.0
  import-route direct
```

```
peer 10.1.1.2 enable
peer 10.2.1.2 enable
#
return
```

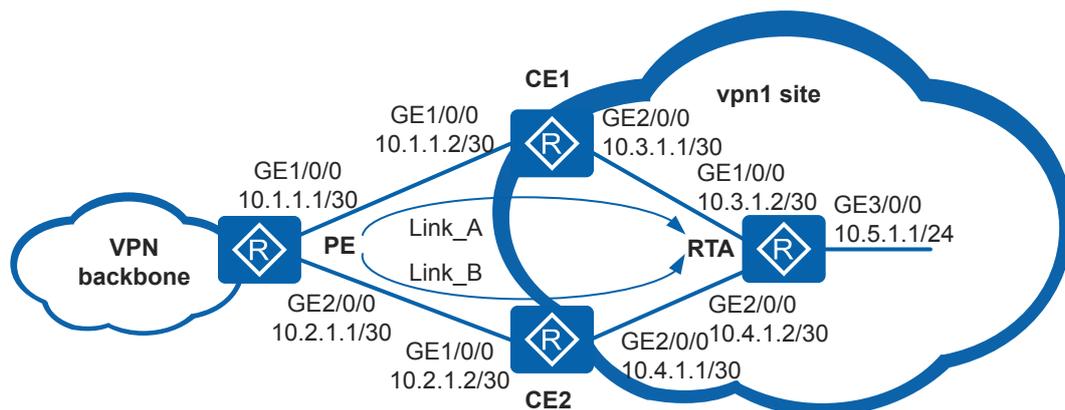
8.9.17 Example for Configuring IP FRR for VPN Routes

Networking Requirements

When multiple CEs in a site connect to the same PE, the PE learns multiple IP VPN routes with the same VPN prefix. To use one of IP VPN routes as the primary route and the other as backup routes, configure IP FRR for VPN routes. Then the PE generates primary and backup routes to the VPN prefix. When the link of the primary route fails, IP traffic on the VPN is quickly switched to the link of a backup route.

As shown in [Figure 8-58](#), the PE has two OSPF routes to RTA. The route on Link_A is the optimal route, and the route on Link_B is the suboptimal route. IP FRR for VPN routes needs to be configured on the PE to quickly switch IP traffic on the VPN to Link_B when Link_A fails.

Figure 8-58 Networking diagram for configuring IP FRR for VPN routes



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable basic OSPF functions on each router so that routes to RTA can be advertised to CE1 and CE2.
2. On the PE, configure VPN instance vpn1, bind GE1/0/0 and GE2/0/0 to vpn1, and configure OSPF multi-instance.
3. Set the cost on GE2/0/0 of the PE and RTA both to a large value so that OSPF preferentially selects Link_A.
4. Configure IP FRR for VPN routes on the PE.
5. Configure BFD to detect the link status.

Procedure

Step 1 Assign IP addresses to interfaces.

Assign IP addresses to the interfaces on RTA.

```
<Huawei> system-view
[Huawei] sysname RTA
[RTA] interface gigabitethernet 1/0/0
[RTA-GigabitEthernet1/0/0] ip address 10.3.1.2 30
[RTA-GigabitEthernet1/0/0] quit
[RTA] interface gigabitethernet 2/0/0
[RTA-GigabitEthernet2/0/0] ip address 10.4.1.2 30
[RTA-GigabitEthernet2/0/0] quit
[RTA] interface gigabitethernet 3/0/0
[RTA-GigabitEthernet3/0/0] ip address 10.5.1.1 30
[RTA-GigabitEthernet3/0/0] quit
```

The configuration on PE, CE1, and CE2 is similar to the configuration on RTA and is not mentioned here.

Step 2 Configure OSPF on CE1, CE2, and RTA.

Configure CE1.

```
[CE1] ospf 1
[CE1-ospf] area 0
[CE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.3
[CE1-ospf-1-area-0.0.0.0] network 10.3.1.0 0.0.0.3
[CE1-ospf-1-area-0.0.0.0] quit
[CE1-ospf-1] quit
```

The configuration on CE2 and RTA is similar to the configuration on CE1 and is not mentioned here.

After the configuration is complete, CE1, CE2, and RTA can learn interface addresses from each other. The information displayed on CE1 is used as an example.

```
[CE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
Destinations : 13          Routes : 13

Destination/Mask  Proto  Pre  Cost           Flags  NextHop           Interface
-----
10.1.1.0/30      Direct  0    0                D    10.1.1.2
GigabitEthernet1/0/0
10.1.1.2/32      Direct  0    0                D    127.0.0.1
GigabitEthernet1/0/0
10.1.1.3/32      Direct  0    0                D    127.0.0.1
GigabitEthernet1/0/0
10.3.1.0/30      Direct  0    0                D    10.3.1.1
GigabitEthernet2/0/0
10.3.1.1/32      Direct  0    0                D    127.0.0.1
GigabitEthernet2/0/0
10.3.1.3/32      Direct  0    0                D    127.0.0.1
GigabitEthernet2/0/0
10.2.1.0/30      OSPF   10   3                D    10.3.1.2
GigabitEthernet2/0/0
10.4.1.0/30      OSPF   10   2                D    10.3.1.2
GigabitEthernet2/0/0
10.5.1.0/24      OSPF   10   2                D    10.3.1.2
GigabitEthernet2/0/0
127.0.0.0/8      Direct  0    0                D    127.0.0.1          InLoopBack0
127.0.0.1/32     Direct  0    0                D    127.0.0.1          InLoopBack0
```

```
127.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Step 3 Configure a VPN instance and OSPF multi-instance on the PE.

On the PE, configure VPN instance vpn1 and bind GE1/0/0 and GE2/0/0 to vpn1.

```
[PE] ip vpn-instance vpn1
[PE-vpn-instance-vpn1] ipv4-family
[PE-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE-vpn-instance-vpn1-af-ipv4] quit
[PE-vpn-instance-vpn1] quit
[PE] interface gigabitethernet 1/0/0
[PE-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE-GigabitEthernet1/0/0] ip address 10.1.1.1 30
[PE-GigabitEthernet1/0/0] quit
[PE] interface gigabitethernet 2/0/0
[PE-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE-GigabitEthernet2/0/0] ip address 10.2.1.1 30
[PE-GigabitEthernet2/0/0] quit
```

Configure OSPF multi-instance on the PE.

```
[PE] ospf vpn-instance vpn1
[PE-ospf-1] area 0
[PE-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.3
[PE-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.3
[PE-ospf-1-area-0.0.0.0] quit
[PE-ospf-1] quit
```

Step 4 Set the cost on the OSPF interface.

Set the cost on GE2/0/0 of the PE to 100 so that OSPF preferentially selects Link_A.

```
[PE] interface gigabitethernet 2/0/0
[PE-GigabitEthernet2/0/0] ospf cost 100
[PE-GigabitEthernet2/0/0] quit
```

Set the cost on GE2/0/0 of RTA to 100 so that OSPF preferentially selects Link_A.

```
[RTA] interface gigabitethernet 2/0/0
[RTA-GigabitEthernet2/0/0] ospf cost 100
[RTA-GigabitEthernet2/0/0] quit
```

Step 5 Configure a routing policy.

Configure a routing policy, a backup next hop, and a backup outbound interface on the PE.
 Configure an if-match clause.

```
[PE] ip ip-prefix frr1 permit 10.5.1.0 24
[PE] route-policy ip_frr_rp permit node 10
[PE-route-policy] if-match ip-prefix frr1
[PE-route-policy] apply backup-nexthop 10.2.1.2
[PE-route-policy] apply backup-interface gigabitethernet 2/0/0
[PE-route-policy] quit
```

Step 6 Configure association between BFD and IP FRR.

Configure the PE.

```
[PE] bfd
[PE-bfd] quit
[PE] bfd for_ip_frr bind peer-ip 10.1.1.2 vpn-instance vpn1 interface
gigabitethernet 1/0/0
[PE-bfd-session-for_ip_frr] discriminator local 10
[PE-bfd-session-for_ip_frr] discriminator remote 20
[PE-bfd-session-for_ip_frr] min-tx-interval 100
[PE-bfd-session-for_ip_frr] min-rx-interval 100
```

```
[PE-bfd-session-for_ip_frr] commit
[PE-bfd-session-for_ip_frr] quit
```

Configure CE1.

```
[CE1] bfd
[CE1-bfd] quit
[CE1] bfd for_ip_frr bind peer-ip 10.1.1.1 interface gigabitethernet 1/0/0
[CE1-bfd-session-for_ip_frr] discriminator local 20
[CE1-bfd-session-for_ip_frr] discriminator remote 10
[CE1-bfd-session-for_ip_frr] min-tx-interval 100
[CE1-bfd-session-for_ip_frr] min-rx-interval 100
[CE1-bfd-session-for_ip_frr] commit
[CE1-bfd-session-for_ip_frr] quit
```

Run the **display bfd session all verbose** command on the PE and CE1. The command output shows that the BFD session status is Up.

Step 7 Enable IP FRR for VPN routes.

```
[PE] ip vpn-instance vpn1
[PE-vpn-instance-vpn1] ipv4-family
[PE-vpn-instance-vpn1-af-ipv4] ip frr route-policy ip_frr_rp
[PE-vpn-instance-vpn1-af-ipv4] quit
[PE-vpn-instance-vpn1] quit
```

Step 8 Verify the configurations.

Run the **display ip routing-table vpn-instance** command on the PE. The command output shows that the next hop of the route to 10.5.1.0/24 is 10.1.1.2, and the route has a backup next hop and a backup outbound interface.

```
[PE] display ip routing-table vpn-instance vpn1 10.5.1.0 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Table : vpn1
Summary Count : 1
Destination: 10.5.1.0/24
  Protocol: OSPF                Process ID: 1
  Preference: 10                Cost: 3
  NextHop: 10.1.1.2             Neighbour: 0.0.0.0
    State: Active Adv           Age: 00h00m03s
    Tag: 0                       Priority: low
    Label: NULL                  QoSInfo: 0x0
  IndirectID: 0x0
  RelayNextHop: 0.0.0.0         Interface: GigabitEthernet1/0/0
  TunnelID: 0x0                 Flags: D
  BkNextHop: 10.2.1.2          BkInterface: GigabitEthernet2/0/0
    BkLabel: NULL                SecTunnelID: 0x0
  BkPETunnelID: 0x0            BkPESecTunnelID: 0x0
  BkIndirectID: 0x0
```

Run the **shutdown** command on GE1/0/0 of CE1 to simulate a link failure.

```
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] shutdown
[CE1-GigabitEthernet1/0/0] quit
```

Run the **display ip routing-table vpn-instance** command on the PE again. The command output shows that the next hop of the route to 10.5.1.0/24 is 10.2.1.2.

```
[PE] display ip routing-table vpn-instance vpn1 10.5.1.0 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Table : vpn1
Summary Count : 1
Destination: 10.5.1.0/24
```

```

    Protocol: OSPF                Process ID: 1
    Preference: 10                Cost: 102
      NextHop: 10.2.1.2          Neighbour: 0.0.0.0
        State: Active Adv       Age: 00h01m03s
          Tag: 0                 Priority: low
            Label: NULL         QoSInfo: 0x0
      IndirectID: 0x0
    RelayNextHop: 0.0.0.0        Interface: GigabitEthernet2/0/0
      TunnelID: 0x0              Flags: D
        BkNextHop: 10.2.1.2     BkInterface: GigabitEthernet2/0/0
          BkLabel: NULL         SecTunnelID: 0x0
    BkPETunnelID: 0x0           BkPESecTunnelID: 0x0
    BkIndirectID: 0x0
  
```

----End

Configuration Files

- PE configuration file

```

#
sysname PE
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:1
    ip frr route-policy ip_frr_rp
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
bfd
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn1
  ip address 10.1.1.1 255.255.255.252
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpn1
  ip address 10.2.1.1 255.255.255.252
  ospf cost 100
#
ospf 1 vpn-instance vpn1
  area 0.0.0.0
    network 10.1.1.0 0.0.0.3
    network 10.2.1.0 0.0.0.3
#
ip ip-prefix frr1 index 10 permit 10.5.1.0 24
#
route-policy ip_frr_rp permit node 10
  if-match ip-prefix frr1
  apply backup-nexthop 10.2.1.2
  apply backup-interface GigabitEthernet2/0/0
#
bfd for_ip_frr bind peer-ip 10.1.1.2 vpn-instance vpn1 interface
GigabitEthernet 1/0/0
  discriminator local 10
  discriminator remote 20
  min-tx-interval 100
  min-rx-interval 100
  commit
#
return
  
```

- CE1 configuration file

```

#
sysname CE1
#
bfd
#
  
```

```
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 10.3.1.1 255.255.255.252
#
ospf 1
 area 0.0.0.0
  network 10.1.1.0 0.0.0.3
  network 10.3.1.0 0.0.0.3
#
bfd for_ip_frr bind peer-ip 10.1.1.1 interface GigabitEthernet 1/0/0
 discriminator local 20
 discriminator remote 10
 min-tx-interval 100
 min-rx-interval 100
 commit
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 10.4.1.1 255.255.255.252
#
ospf 1
 area 0.0.0.0
  network 10.2.1.0 0.0.0.3
  network 10.4.1.0 0.0.0.3
#
return
```

- RTA configuration file

```
#
sysname RTA
#
interface GigabitEthernet1/0/0
 ip address 10.3.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
 ip address 10.4.1.2 255.255.255.252
 ospf cost 100
#
interface GigabitEthernet3/0/0
 ip address 10.5.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 10.3.1.0 0.0.0.3
  network 10.4.1.0 0.0.0.3
 area 0.0.0.2
  network 10.5.1.0 0.0.0.255
#
return
```

8.9.18 Example for Configuring VPN GR

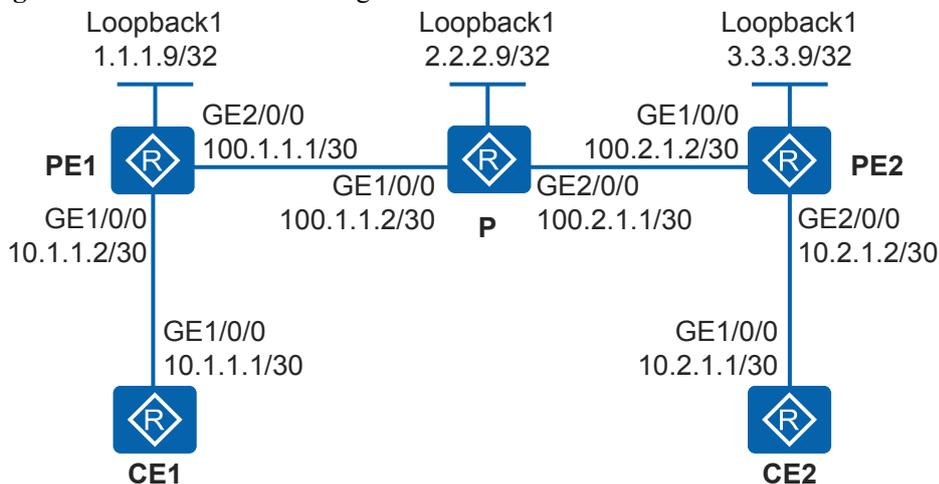
Networking Requirements

 **NOTE**

Only the AR3260-S can be used in this scenario.

As shown in **Figure 8-59**, CE1 and CE2 belong to the same VPN. PE1, P, PE2 on the backbone network belong to the same AS and use the IS-IS protocol to exchange routing information. CE1 connects to PE1, and CE2 connects to PE2. BGP runs between CE1 and PE1, and OSPF runs between CE2 and PE2.

Figure 8-59 VPN GR networking



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP/MPLS IP VPN functions.
2. Configure IGP GR, BGP GR, and LDP GR on the backbone network. Configure GR for the routing protocols running between the PE and CE devices to ensure uninterrupted VPN traffic forwarding when an active/standby switchover occurs on any of the CE, PE, and P devices.

Procedure

Step 1 Configure IP addresses for the interfaces on the backbone network.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 100.1.1.1 30
[PE1-GigabitEthernet2/0/0] quit
```

The configurations of PE2 and P are similar to the configuration of PE1, and are not mentioned here.

Step 2 Configure basic BGP/MPLS IP VPN functions on the backbone network.

Configure IS-IS as the IGP on the backbone network, enable LDP on PE1 and PE2, and set up an MP-IBGP peer relationship between PE1 and PE2.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] isis 1
[PE1-isis-1] network-entity 10.0000.0000.0001.00
[PE1-isis-1] quit
[PE1] interface loopback 1
[PE1-LoopBack1] isis enable 1
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 100.1.1.1 30
[PE1-GigabitEthernet2/0/0] isis enable 1
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] isis 1
[P-isis-1] network-entity 10.0000.0000.0002.00
[P-isis-1] quit
[P] interface loopback 1
[P-LoopBack1] isis enable 1
[P-LoopBack1] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] isis enable 1
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] isis enable 1
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] isis 1
[PE2-isis-1] network-entity 10.0000.0000.0003.00
[PE2-isis-1] quit
[PE2] interface loopback 1
[PE2-LoopBack1] isis enable 1
[PE2-LoopBack1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] isis enable 1
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
[PE2] bgp 100
```

```
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

After the configuration is complete, run the **display isis peer** command on PE1 or PE2. You can see that the IS-IS neighbor relationship is in Up state. Run the **display bgp vpnv4 all peer** command, and you can see that the BGP peer relationship has been set up and is in **Established** state. Run the **display mpls ldp session** command, and you can see that an LDP session has been set up and the session status is **Operational**.

Step 3 Configure a VPN instance on the PE devices and bind the instance to the interfaces connected to the CE devices.

Configure VPN instance **vpn1** on PE1 and bind it to the interface connected to CE1. Configure VPN instance **vpn1** on PE2 and bind it to the interface connected to CE2. Set up an EBGP peer relationship between CE1 and PE1. Set up an OSPF neighbor relationship between CE2 and PE2.

Configure CE1.

```
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 30
[CE1-GigabitEthernet1/0/0] quit
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 30
[PE1-GigabitEthernet1/0/0] quit
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet2/0/0] ip address 10.2.1.2 30
[PE2-GigabitEthernet2/0/0] quit
[PE2] ospf 2 vpn-instance vpn1
[PE2-ospf-2] area 0
[PE2-ospf-2-area-0.0.0.0] network 10.2.1.0 0.0.0.3
[PE2-ospf-2-area-0.0.0.0] quit
[PE2-ospf-2] import-route bgp
[PE2-ospf-2] quit
```

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] import-route ospf 2
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

Configure CE2.

```
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 10.2.1.1 30
[CE2-GigabitEthernet1/0/0] quit
[CE2] ospf 2
[CE2-ospf-2] area 0
[CE2-ospf-2-area-0.0.0.0] network 10.2.1.0 0.0.0.3
[CE2-ospf-2-area-0.0.0.0] quit
[CE2-ospf-2] import-route direct
[CE2-ospf-2] quit
```

The basic BGP/MPLS IP VPN configuration is complete, and CE1 and CE2 can communicate with each other.

Step 4 Configure IGP GR on the backbone network.

Configure IGP GR on PE1, P, and PE2.

Configure PE1.

```
[PE1] isis 1
[PE1-isis-1] graceful-restart
[PE1-isis-1] quit
```

Configure P.

```
[P] isis 1
[P-isis-1] graceful-restart
[P-isis-1] quit
```

Configure PE2.

```
[PE2] isis 1
[PE2-isis-1] graceful-restart
[PE2-isis-1] quit
```

Run the **display isis graceful-restart status** command on PE1, P, and PE2. The command output shows that IS-IS GR has been configured successfully.

The display on PE1 is used as an example:

```
[PE1] display isis graceful-restart status

Restart information for ISIS(1)
-----

IS-IS(1) Level-1 Restart Status
Restart Interval: 300
SA Bit Supported
  Total Number of Interfaces = 2
  Restart Status: RESTART COMPLETE

IS-IS(1) Level-2 Restart Status
Restart Interval: 300
SA Bit Supported
  Total Number of Interfaces = 2
  Restart Status: RESTART COMPLETE
```

Step 5 Configure MPLS LDP GR on the backbone network.

Configure MPLS LDP GR on PE1, P, and PE2.

Configure PE1.

```
[PE1] mpls ldp
[PE1-mpls-ldp] graceful-restart
[PE1-mpls-ldp] quit
```

Configure P.

```
[P] mpls ldp
[P-mpls-ldp] graceful-restart
[P-mpls-ldp] quit
```

Configure PE2.

```
[PE2] mpls ldp
[PE2-mpls-ldp] graceful-restart
[PE2-mpls-ldp] quit
```

Step 6 Configure GR for the routing protocols running between the PE and CE devices.

Configure BGP GR on PE1 and CE1. Configure OSPF GR on PE2 and CE2.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] graceful-restart
[PE1-bgp] quit
```

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] graceful-restart
[CE1-bgp] quit
```

Configure PE2.

```
[PE2] ospf 2 vpn-instance vpn1
[PE2-ospf-2] opaque-capability enable
[PE2-ospf-2] graceful-restart
[PE2-ospf-2] quit
```

Configure CE2.

```
[CE2] ospf 2
[CE2-ospf-2] opaque-capability enable
[CE2-ospf-2] graceful-restart
[CE2-ospf-2] quit
```

Run the **display ospf brief** command on PE2 or CE2. The command output shows that OSPF GR has been configured successfully.

The display on PE2 is used as an example:

```
[PE2] display ospf brief

      OSPF Process 2 with Router ID 10.2.1.2
      OSPF Protocol Information

RouterID: 10.2.1.2          Border Router:  AREA  AS
ECA-route-type: 0x0306
Route Tag: 3489661028
PE Router, Multi-VPN-Instance is enabled
Opaque Capable
Global DS-TE Mode: Non-Standard IETF Mode
Graceful-restart capability: planned and un-planned, totally
Helper support capability  : enabled
      filter capability    : disabled
      policy capability    : strict lsa check, planned and un-planned
Applications Supported: MPLS Traffic-Engineering
```

```
Spf-schedule-interval: max 10000ms, start 500ms, hold 1000ms
Default ASE parameters: Metric: 1 Tag: 1 Type: 2
Route Preference: 10
ASE Route Preference: 150
SPF Computation Count: 17
RFC 1583 Compatible
Retransmission limitation is disabled
Area Count: 1 Nssa Area Count: 0
Exchange/Loading Neighbors: 0
Process total up interface count: 1
Process valid up interface count: 1

Area: 0.0.0.0 (MPLS TE not enabled)
AuthType: None Area flag: Normal
SPF scheduled Count: 17
Exchange/Loading Neighbors: 0
Router ID conflict state: Normal
Area interface up count: 1

Interface: 10.2.1.2 (GigabitEthernet2/0/0)
Cost: 1 State: DR Type: Broadcast MTU: 1500
Priority: 1
Designated Router: 10.2.1.2
Backup Designated Router: 10.2.1.1
Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1
```

Step 7 Configure BGP GR on the PE devices.

BGP GR has been configured in step 6, so you only need to configure BGP GR on PE2 in this step.

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] graceful-restart
[PE2-bgp] quit
```

Run the **display bgp vpnv4 all peer verbose** command on PE1. The command output shows that IBGP GR has taken effect between PE1 and PE2, and EBGP GR has taken effect between PE1 and CE1.

```
[PE1] display bgp vpnv4 all peer verbose

BGP Peer is 3.3.3.9, remote AS 100
Type: IBGP link
BGP version 4, Remote router ID 3.3.3.9
Update-group ID: 1
BGP current state: Established, Up for 00h01m04s
BGP current event: RecvKeepalive
BGP last state: OpenConfirm
BGP Peer Up count: 3
Received total routes: 3
Received active routes total: 3
Received mac routes: 0
Advertised total routes: 2
Port: Local - 179 Remote - 56400
Configured: Connect-retry Time: 32 sec
Configured: Min Hold Time: 0 sec
Configured: Active Hold Time: 180 sec Keepalive Time:60 sec
Received : Active Hold Time: 180 sec
Negotiated: Active Hold Time: 180 sec Keepalive Time:60 sec
Peer optional capabilities:
Peer supports bgp multi-protocol extension
Peer supports bgp route refresh capability
Peer supports bgp 4-byte-as capability
Graceful Restart Capability: advertised and received
Restart Timer Value received from Peer: 150 seconds
Address families preserved for peer in GR:
IPv4 Unicast (was preserved)
```

```

VPNv4 (was preserved)
  Address family IPv4 Unicast: advertised and received
  Address family VPNv4: advertised and received
Received: Total 7 messages
  Update messages          4
  Open messages            1
  KeepAlive messages       2
  Notification messages    0
  Refresh messages        0
Sent: Total 8 messages
  Update messages          3
  Open messages            2
  KeepAlive messages       3
  Notification messages    0
  Refresh messages        0
Authentication type configured: None
Last keepalive received: 2013/09/15 19:43:15
Last keepalive sent      : 2013/09/15 19:43:15
Last update received: 2013/09/15 19:42:15
Last update sent        : 2013/09/15 19:42:15
Minimum route advertisement interval is 0 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Connect-interface has been configured
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured

  IPv4-family for VPN instance:  vpn1

BGP Peer is 10.1.1.1, remote AS 65410
Type: EBGP link
BGP version 4, Remote router ID 10.1.1.1
Update-group ID: 1
BGP current state: Established, Up for 00h05m43s
BGP current event: KATimerExpired
BGP last state: OpenConfirm
BGP Peer Up count: 2
Received total routes: 2
Received active routes total: 0
Received mac routes: 0
Advertised total routes: 3
Port: Local - 179 Remote - 49695
Configured: Connect-retry Time: 32 sec
Configured: Min Hold Time: 0 sec
Configured: Active Hold Time: 180 sec Keepalive Time:60 sec
Received : Active Hold Time: 180 sec Keepalive Time:60 sec
Negotiated: Active Hold Time: 180 sec Keepalive Time:60 sec
Peer optional capabilities:
Peer supports bgp multi-protocol extension
Peer supports bgp route refresh capability
Peer supports bgp 4-byte-as capability
Graceful Restart Capability: advertised and received
Restart Timer Value received from Peer: 150 seconds
Address families preserved for peer in GR:
  IPv4 Unicast (was preserved)
  Address family IPv4 Unicast: advertised and received
Received: Total 10 messages
  Update messages          3
  Open messages            1
  KeepAlive messages       6
  Notification messages    0
  Refresh messages        0
Sent: Total 15 messages
  Update messages          6
  Open messages            2
  KeepAlive messages       7
  Notification messages    0
```

```
Refresh messages 0
Authentication type configured: None
Last keepalive received: 2013/09/15 19:42:37
Last keepalive sent : 2013/09/15 19:42:37
Last update received: 2013/09/15 19:37:37
Last update sent : 2013/09/15 19:42:15
Minimum route advertisement interval is 30 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

Step 8 Verify the configuration.

Run the **display switchover state** command on PE1 to check the status of the slave SRU. The following information is displayed:

```
[PE1] display switchover state
Slot 15 HA FSM State(master): realtime or routine backup.
Slot 14 HA FSM State(slave): receiving realtime or routine data.
```

Perform an active/standby switchover on PE1.

```
[PE1] slave switchover
Are you sure to switch over? (y/n) [n]:y
```

Communication between the site connected to CE1 and the site connected to CE2 is not interrupted.

NOTE

Communication between the sites may be interrupted when two or more neighboring devices among CE1, PE1, PE2, and CE2 perform an active/standby switchover at the same time.

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpn1
  ipv4-family
    route-distinguisher 100:1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
  graceful-restart
#
isis 1
  graceful-restart
  network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
  ip binding vpn-instance vpn1
  ip address 10.1.1.2 255.255.255.252
#
interface GigabitEthernet2/0/0
  ip address 100.1.1.1 255.255.255.252
  isis enable 1
```

```
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
isis enable 1
#
bgp 100
 graceful-restart
 peer 3.3.3.9 as-number 100
 peer 3.3.3.9 connect-interface LoopBack1
#
 ipv4-family vpnv4
  policy vpn-target
  peer 3.3.3.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.1.1.1 as-number 65410
#
return
```

- P configuration file

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
 graceful-restart
#
isis 1
 graceful-restart
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.252
isis enable 1
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 100.2.1.1 255.255.255.252
isis enable 1
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
isis enable 1
#
return
```

- PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:2
  vpn-target 111:1 export-extcommunity
  vpn-target 111:1 import-extcommunity
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
 graceful-restart
#
isis 1
```

```
 graceful-restart
 network-entity 10.0000.0000.0003.00
 #
 interface GigabitEthernet1/0/0
 ip address 100.2.1.2 255.255.255.252
 isis enable 1
 mpls
 mpls ldp
 #
 interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 10.2.1.2 255.255.255.252
 #
 interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
 isis enable 1
 #
 bgp 100
 graceful-restart
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
 #
 ipv4-family vpnv4
 policy vpn-target
 peer 1.1.1.9 enable
 #
 ipv4-family vpn-instance vpn1
 import-route ospf 2
 #
 ospf 2 vpn-instance vpn1
 import-route bgp
 opaque-capability enable
 graceful-restart
 area 0.0.0.0
 network 10.2.1.0 0.0.0.3
 #
 return
```

● CE1 configuration file

```
 #
 sysname CE1
 #
 interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.252
 #
 bgp 65410
 graceful-restart
 peer 10.1.1.2 as-number 100
 #
 ipv4-family
 unicast
 undo
 synchronization
 import-route
 direct
 peer 10.1.1.2
 enable
 #
 return
```

● CE2 configuration file

```
 #
 sysname CE2
 #
 interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.252
 #
 ospf 2
```

```
import-route direct
opaque-capability enable
graceful-restart
area 0.0.0.0
 network 10.2.1.0 0.0.0.3
#
return
```

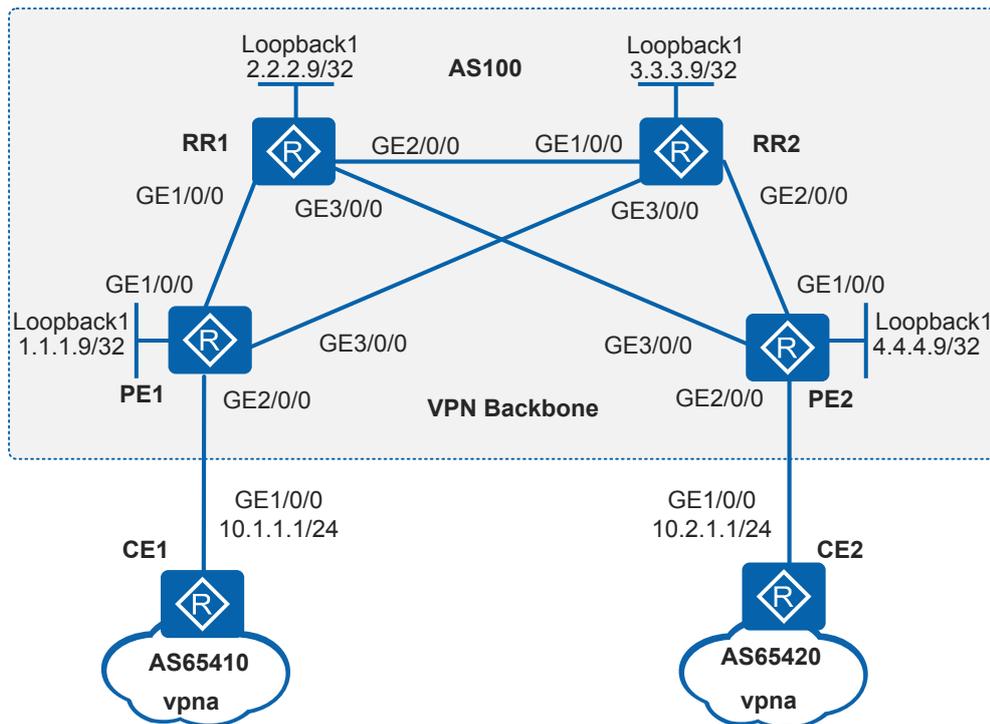
8.9.19 Example for Configuring Double RRs to Optimize the VPN Backbone Layer

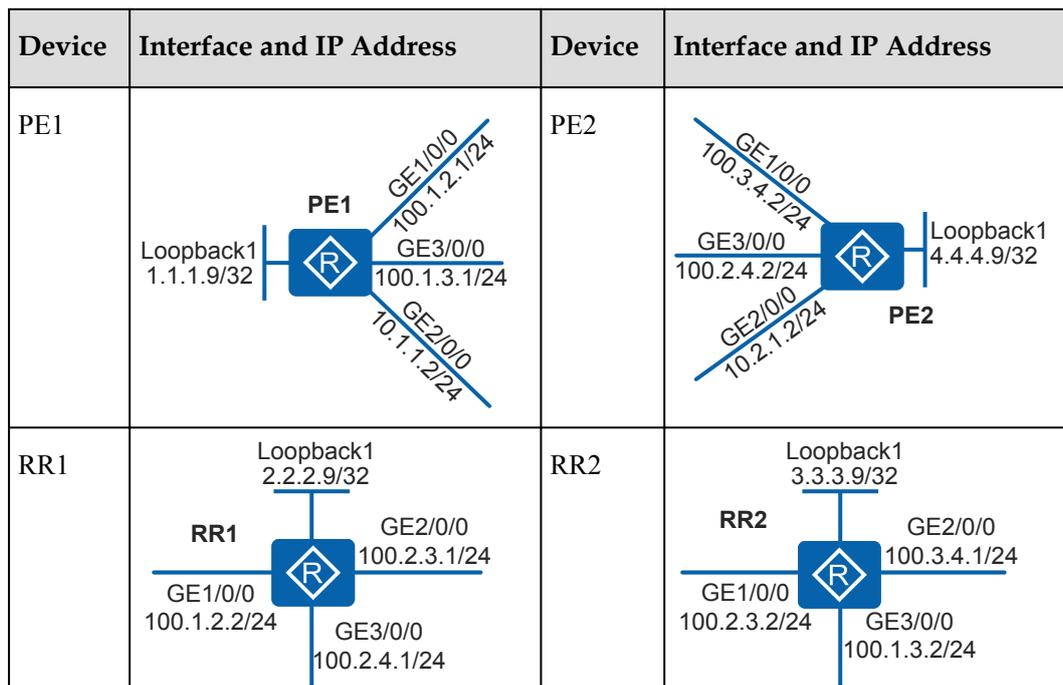
Networking Requirements

When deploying a VPN, you can configure double route reflectors (RRs) on the VPN. To achieve this, you need to select two RRs from the PEs in the same AS on the backbone network and ensure that the two RRs back up each other and reflect routes of the public network and VPNv4.

As shown in [Figure 8-60](#), PE1, PE2, RR1, and RR2 are located in AS 100 on the backbone network. CE1 and CE2 belong to vpna. Select RR1 and RR2 as the RRs of the VPN.

Figure 8-60 Networking diagram for configuring double RRs on a VPN





Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IGP protocol on the MPLS backbone network for IP connectivity.
2. Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up MPLS LSPs.
3. Configure VPN instances on PE1 and PE2 and bind the instances to the interfaces connected to the CEs. Configure the same VPN target for the VPN instances to enable users in the same VPN to communicate with each other.
4. Set up EBGP peer relationships between the PEs and CEs and import VPN routes into BGP.
5. Set up MP-IBGP peer relationships between PEs and RRs. The PEs do not need to set up an MP-IBGP peer relationship.
6. Configure the same reflector cluster ID for RR1 and RR2 so that they back up each other.
7. Configure RR1 and RR2 to accept all VPNv4 routes without filtering the routes based on VPN targets, because RR1 and RR2 must save all VPNv4 routes and advertise them to PEs.

NOTE

On a VPN with double RRs, ensure that each RR has at least two paths to a PE and the paths do not share the same network segment or node. If there is only one path between the RRs and PEs or if the paths share the same network segment or node, double RRs cannot improve network reliability.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-60](#).

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 100.1.2.1 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip address 100.1.3.1 24
[PE1-GigabitEthernet3/0/0] quit
```

The configuration on PE2, RRs, CE1, and CE2 is similar to the configuration on PE1 and is not mentioned here.

Step 2 Configure an IGP protocol on the MPLS backbone network for IP connectivity.

Configure PE1.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 100.1.2.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 100.1.3.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration on PE2 and RRs is similar to the configuration on PE1 and is not mentioned here.

NOTE

The IP addresses of loopback interfaces that are used as LSR IDs need to be advertised.

After the configuration is complete, the devices on the backbone network can learn the loopback interface addresses from each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
          Destinations : 17          Routes : 19

Destination/Mask    Proto    Pre  Cost           Flags    NextHop         Interface
-----
      1.1.1.9/32     Direct   0    0                D    127.0.0.1       LoopBack1
      2.2.2.9/32     OSPF    10    1                D    100.1.2.2
GigabitEthernet1/0/0
      3.3.3.9/32     OSPF    10    1                D    100.1.3.2
GigabitEthernet3/0/0
      4.4.4.9/32     OSPF    10    2                D    100.1.3.2
GigabitEthernet1/0/0
      OSPF    10    2                D    100.1.2.2
GigabitEthernet3/0/0
      100.1.2.0/24    Direct   0    0                D    100.1.2.1
GigabitEthernet1/0/0
      100.1.2.1/32    Direct   0    0                D    127.0.0.1
GigabitEthernet1/0/0
      100.1.2.255/32  Direct   0    0                D    127.0.0.1
GigabitEthernet1/0/0
      100.1.3.0/24    Direct   0    0                D    100.1.3.1
GigabitEthernet3/0/0
      100.1.3.1/32    Direct   0    0                D    127.0.0.1
GigabitEthernet3/0/0
      100.1.3.255/32  Direct   0    0                D    127.0.0.1
```

```
GigabitEthernet3/0/0
  100.2.3.0/24 OSPF 10 2 D 100.1.3.2
GigabitEthernet3/0/0
  OSPF 10 2 D 100.1.2.2
GigabitEthernet1/0/0
  100.2.4.0/24 OSPF 10 2 D 100.1.2.2
GigabitEthernet1/0/0
  100.3.4.0/24 OSPF 10 2 D 100.1.3.2
GigabitEthernet3/0/0
  127.0.0.0/8 Direct 0 0 D 127.0.0.1 InLoopBack0
  127.0.0.1/32 Direct 0 0 D 127.0.0.1 InLoopBack0
127.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
```

Step 3 Configure basic MPLS capabilities and MPLS LDP on the MPLS backbone network to set up LDP LSPs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls
[PE1-GigabitEthernet3/0/0] mpls ldp
[PE1-GigabitEthernet3/0/0] quit
```

The configuration on PE2 and RRs is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display mpls ldp session** command on the PEs and RRs. The State field in the command output displays as Operational.

The information displayed on PE1 and RR1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0             Operational DU   Passive  0000:00:01  8/8
3.3.3.9:0             Operational DU   Passive  0000:00:00  4/4
-----
TOTAL: 2 session(s) Found.
[RR1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
1.1.1.9:0             Operational DU   Active   000:00:02  11/11
3.3.3.9:0             Operational DU   Passive  000:00:01  8/8
4.4.4.9:0             Operational DU   Passive  000:00:00  4/4
-----
TOTAL: 3 session(s) Found.
```

Step 4 Configure VPN instances on the PEs.

Configure PE1.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 1:1 both
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

Step 5 Set up EBGp peer relationships between the PEs and CEs and import VPN routes into BGP.

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] quit
```

The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] quit
[PE1-bgp] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

Step 6 Set up MP-IBGP peer relationships between PEs and RRs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[PE1-bgp] peer 3.3.3.9 as-number 100
[PE1-bgp] peer 3.3.3.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.9 enable
[PE1-bgp-af-vpnv4] peer 3.3.3.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure RR1.

```
[RR1] bgp 100
[RR1-bgp] group rr1 internal
[RR1-bgp] peer rr1 connect-interface loopback 1
[RR1-bgp] peer 1.1.1.9 group rr1
[RR1-bgp] peer 3.3.3.9 group rr1
[RR1-bgp] peer 4.4.4.9 group rr1
[RR1-bgp] ipv4-family vpnv4
[RR1-bgp-af-vpnv4] peer rr1 enable
[RR1-bgp-af-vpnv4] peer 1.1.1.9 group rr1
[RR1-bgp-af-vpnv4] peer 3.3.3.9 group rr1
[RR1-bgp-af-vpnv4] peer 4.4.4.9 group rr1
[RR1-bgp-af-vpnv4] quit
[RR1-bgp] quit
```

Configure RR2.

```
[RR2] bgp 100
[RR2-bgp] group rr2 internal
[RR2-bgp] peer rr2 connect-interface loopback 1
[RR2-bgp] peer 1.1.1.9 group rr2
[RR2-bgp] peer 2.2.2.9 group rr2
[RR2-bgp] peer 4.4.4.9 group rr2
[RR2-bgp] ipv4-family vpnv4
[RR2-bgp-af-vpnv4] peer rr2 enable
[RR2-bgp-af-vpnv4] peer 1.1.1.9 group rr2
[RR2-bgp-af-vpnv4] peer 2.2.2.9 group rr2
[RR2-bgp-af-vpnv4] peer 4.4.4.9 group rr2
[RR2-bgp-af-vpnv4] quit
[RR2-bgp] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that the PEs have set up IBGP peer relationships with RRs, and the peer relationships are in Established state. The PEs also set up EBGP peer relationships with the CEs.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 3                Peers in established state : 3
Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
2.2.2.9       4   100    2         4         0   00:00:31   Established  0
3.3.3.9       4   100    3         5         0   00:01:23   Established  0

Peer of IPv4-family for vpn instance :

VPN-Instance vpna, Router ID 1.1.1.9:
10.1.1.1     4   65410  79        82         0   01:13:29   Established  0
```

Step 7 Configure route reflection on RR1 and RR2.

Configure RR1.

```
[RR1] bgp 100
[RR1-bgp] ipv4-family vpnv4
[RR1-bgp-af-vpnv4] reflector cluster-id 100
[RR1-bgp-af-vpnv4] peer rr1 reflect-client
[RR1-bgp-af-vpnv4] undo policy vpn-target
[RR1-bgp-af-vpnv4] quit
[RR1-bgp] quit
```

Configure RR2.

```
[RR2] bgp 100
[RR2-bgp] ipv4-family vpnv4
[RR2-bgp-af-vpnv4] reflector cluster-id 100
[RR2-bgp-af-vpnv4] peer rr2 reflect-client
[RR2-bgp-af-vpnv4] undo policy vpn-target
[RR2-bgp-af-vpnv4] quit
[RR2-bgp] quit
```

Step 8 Verify the configuration.

Check the VPN routing table on a PE. The routing table contains a route to the remote CE.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
      Destinations : 8          Routes : 8

Destination/Mask  Proto  Pre  Cost           Flags  NextHop           Interface
-----
 10.1.1.0/24      Direct  0    0                D    10.1.1.2
GigabitEthernet2/0/0
 10.1.1.2/32      Direct  0    0                D    127.0.0.1
GigabitEthernet2/0/0
 10.1.1.255/32    Direct  0    0                D    127.0.0.1
GigabitEthernet2/0/0
 10.2.1.0/24      IBGP    255  0                RD    4.4.4.9
GigabitEthernet3/0/0
 127.0.0.0/8      Direct  0    0                D    127.0.0.1          InLoopBack0
 127.0.0.1/32     Direct  0    0                D    127.0.0.1          InLoopBack0
127.255.255.255/32 Direct  0    0                D    127.0.0.1          InLoopBack0
255.255.255.255/32 Direct  0    0                D    127.0.0.1          InLoopBack0
```

If CE1 and CE2 can ping each other, the route reflection function has been configured successfully.

Run the **shutdown** command in the view of GE3/0/0 on PE1 and GE3/0/0 on PE2. CE1 and CE2 can still ping each other, indicating that the RRs are successfully configured.

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpna
  ipv4-family
  route-distinguisher 100:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 100.1.2.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip binding vpn-instance vpna
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
ip address 100.1.3.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
```

```
peer 3.3.3.9 as-number 100
peer 3.3.3.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
peer 3.3.3.9 enable
#
ipv4-family vpnv4
policy vpn-target
peer 2.2.2.9 enable
peer 3.3.3.9 enable
#
ipv4-family vpn-instance vpna
peer 10.1.1.1 as-number 65410
import-route direct
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 100.1.2.0 0.0.0.255
network 100.1.3.0 0.0.0.255
#
return
```

● RR1 configuration file

```
#
sysname RR1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 100.1.2.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 100.2.3.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet3/0/0
ip address 100.2.4.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
bgp 100
peer 1.1.1.9 as-number 100
peer 3.3.3.9 as-number 100
peer 4.4.4.9 as-number 100
group rr1 internal
peer rr1 connect-interface LoopBack1
#
ipv4-family unicast
undo
synchronization
peer rr1
enable
peer 1.1.1.9
enable
peer 1.1.1.9 group
rr1
peer 3.3.3.9
enable
```

```
peer 3.3.3.9 group rr1
peer 4.4.4.9
enable
peer 4.4.4.9 group rr1
#
ipv4-family vpnv4
 reflector cluster-id 100
 undo policy vpn-target
 peer rr1 enable
 peer rr1 reflect-client
 peer 1.1.1.9 enable
 peer 1.1.1.9 group rr1
 peer 3.3.3.9 enable
 peer 3.3.3.9 group rr1
 peer 4.4.4.9 enable
 peer 4.4.4.9 group rr1
#
ospf 1
 area 0.0.0.0
  network 100.1.2.0 0.0.0.255
  network 100.2.3.0 0.0.0.255
  network 100.2.4.0 0.0.0.255
  network 2.2.2.9 0.0.0.0
#
return
```

● RR2 configuration file

```
#
sysname RR2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 100.2.3.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 100.3.4.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet3/0/0
 ip address 100.1.3.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 2.2.2.9 as-number 100
 peer 4.4.4.9 as-number 100
 group rr2 internal
 peer rr2 connect-interface LoopBack1
#
ipv4-family unicast
 undo
synchronization
 peer rr2
enable
 peer 1.1.1.9
enable
 peer 1.1.1.9 group
rr2
 peer 3.3.3.9
```

```
enable
 peer 3.3.3.9 group rr2
 peer 4.4.4.9
enable
 peer 4.4.4.9 group rr2
#
ipv4-family vpnv4
 reflector cluster-id 100
 undo policy vpn-target
 peer rr2 enable
 peer rr2 reflect-client
 peer 1.1.1.9 enable
 peer 1.1.1.9 group rr2
 peer 2.2.2.9 enable
 peer 2.2.2.9 group rr2
 peer 4.4.4.9 enable
 peer 4.4.4.9 group rr2
#
ospf 1
 area 0.0.0.0
 network 100.2.3.0 0.0.0.255
 network 100.3.4.0 0.0.0.255
 network 100.1.3.0 0.0.0.255
 network 3.3.3.9 0.0.0.0
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
 ipv4-family
  route-distinguisher 100:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 100.3.4.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpna
 ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 100.2.4.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 4.4.4.9 255.255.255.255
#
bgp 100
 peer 2.2.2.9 as-number 100
 peer 2.2.2.9 connect-interface LoopBack1
 peer 3.3.3.9 as-number 100
 peer 3.3.3.9 connect-interface LoopBack1
#
ipv4-family unicast
 undo synchronization
 peer 2.2.2.9 enable
 peer 3.3.3.9 enable
#
```

```
ipv4-family vpnv4
 policy vpn-target
 peer 3.3.3.9 enable
 peer 2.2.2.9 enable
#
ipv4-family vpn-instance vpna
 peer 10.2.1.1 as-number 65420
 import-route direct
#
ospf 1
 area 0.0.0.0
 network 4.4.4.9 0.0.0.0
 network 100.3.4.0 0.0.0.255
 network 100.2.4.0 0.0.0.255
#
return
```

- CE1 configuration file

```
#
 sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
bgp 65410
 peer 10.1.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  peer 10.1.1.2 enable
#
return
```

- CE2 configuration file

```
#
 sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
bgp 65420
 peer 10.2.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  peer 10.2.1.2 enable
#
return
```

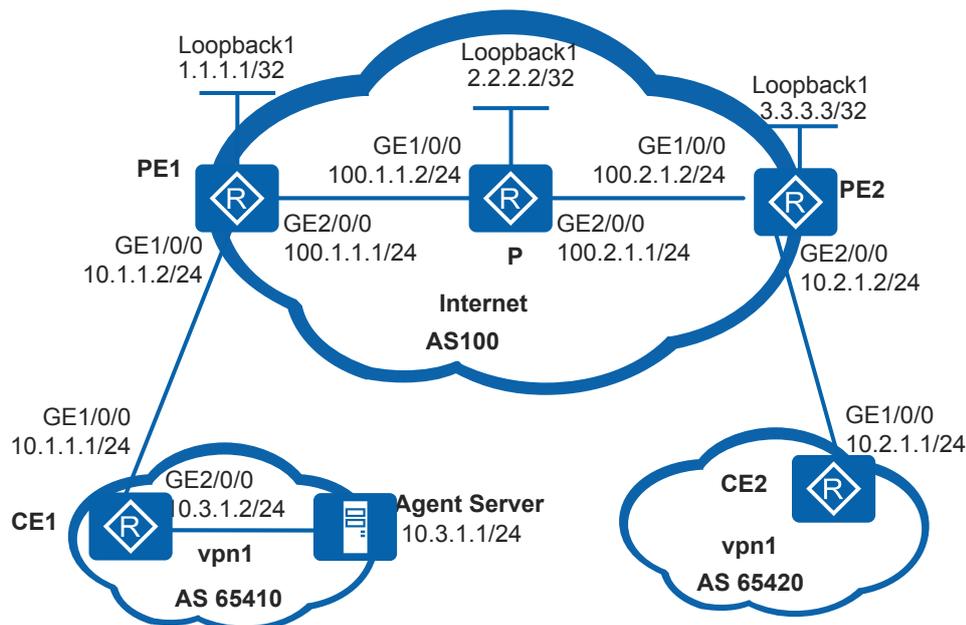
8.9.20 Example for Connecting a VPN to the Internet

Networking Requirements

As shown in [Figure 8-61](#), CE1 and CE2 need to communicate with each other, and users connected to CE1 need to connect to the Internet.

To enable users connected to CE1 to access the Internet, connect an agent server to CE1 and configure a public IP address for the agent server. Then users connected to CE1 can access the Internet through the agent server. In this example, the P represents on the Internet.

Figure 8-61 Networking diagram for connecting a VPN to the Internet



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP/MPLS IP VPN functions.
2. Configure three static routes:
 - On CE1, create a default route and specify PE1 as the next hop.
 - On PE1, configure a default route from the VPN to the Internet and specify P as the next hop. This route enables traffic to be transmitted from the agent server to the Internet.
 - On PE1, configure a static route from the Internet to the agent server and specify CE1 as the next hop. Configure IGP to advertise the static route to the Internet. This route enables traffic to be transmitted from the Internet to the agent server.

Procedure

Step 1 Assign IP addresses to interfaces according to [Figure 8-61](#).

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.1 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 100.1.1.1 24
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2, P, CE1, and CE2 is similar to the configuration on PE1 and is not mentioned here.

Step 2 Configure an IGP protocol on the MPLS backbone network for IP connectivity.

Configure PE1.

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 100.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration on PE2 and P is similar to the configuration on PE1 and is not mentioned here.

NOTE

The IP addresses of loopback interfaces that are used as LSR IDs need to be advertised.

After the configuration is complete, the devices on the backbone network can learn the loopback interface addresses from each other.

Step 3 Set up MPLS LDP LSPs and an MP-IBGP peer relationship between the devices on the backbone network.

Enable MPLS LDP on PE1 to set up MPLS LDP LSPs.

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

The configuration on PE2 and P is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display mpls ldp session** command on P. The command output shows that the LDP sessions between PE1 and P, and between PE2 and P are in Operational state.

The information displayed on P is used as an example.

```
[P] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
1.1.1.1:0             Operational DU   Active    0000:00:00  2/2
3.3.3.3:0             Operational DU   Active    0000:23:08  5556/5555
-----
TOTAL: 2 session(s) Found.
```

Configure an MP-IBGP peer on PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 3.3.3.3 as-number 100
[PE1-bgp] peer 3.3.3.3 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
```

```
[PE1-bgp-af-ipv4] peer 3.3.3.3 enable
[PE1-bgp-af-ipv4] quit
[PE1-bgp] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

Run the **display bgp vpnv4 all peer** command on PE1 and PE2. The command output shows that an MP-IBGP peer relationship has been set up between the PEs and is in Established state. The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
3.3.3.3       4   100      6         8     0  00:03:48  Established  2
```

Step 4 Create VPN instances and set up EBGP peer relationships.

Create VPN instance vpn1 on the PEs and bind it to the interfaces connected to CEs. The information displayed on PE1 is used as an example.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

Set up EBGP peer relationships between PE1 and CE1 and between PE2 and CE2 so that routes of the CEs can be advertised to the PEs. The configuration on CE1 and PE1 is used as an example.

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] import-route static
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

After the configuration is complete, run the **display ip vpn-instance** command on the PEs. In the command output, vpn1 is displayed in the VPN-Instance Name field.

The information displayed on PE1 is used as an example.

```
[PE1] display ip vpn-instance
Total VPN-Instances configured      : 1
Total IPv4 VPN-Instances configured : 1
Total IPv6 VPN-Instances configured : 0
```

VPN-Instance Name	RD	Address-family
vpn1	100:1	IPv4

Run the **display bgp vpnv4 all peer** command on the PEs. The command output shows that the IBGP and EBGp peer relationships are all in Established state.

The information displayed on PE1 is used as an example.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 2                Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
3.3.3.3	4	100	127	134	0	01:39:44	Established	2

Peer of IPv4-family for vpn instance :

```
VPN-Instance vpn1, Router ID 1.1.1.1:
10.1.1.1      4 65410      107      110      0 01:26:33 Established      3
```

Step 5 Configure static routes to enable VPN users to access the Internet.

On CE1, create a default route and specify PE1 as the next hop.

```
[CE1] ip route-static 0.0.0.0 0 10.1.1.2
```

Configure PE1.

Configure a default route from the agent server to the Internet and specify P as the next hop. Specify the **public** keyword in the command to use the public IP address of P as the next hop address.

```
[PE1] ip route-static vpn-instance vpn1 0.0.0.0 0 100.1.1.2 public
```

NOTE

If the CEs and PEs are connected through an Ethernet network, you must specify the next hop when configuring the static route.

Configure a static route from the Internet to the agent server and specify CE1 as the next hop.

```
[PE1] ip route-static 10.3.1.0 24 vpn-instance vpn1 10.1.1.1
```

Advertise the preceding static route to the Internet using an IGP (OSPF in this example).

```
[PE1] ospf 1
[PE1-ospf-1] import-route static
[PE1-ospf-1] quit
```

Configure the agent server. Set the IP address of the agent server to 10.3.1.1/24 and the default gateway address of the agent server to 10.3.1.2/24 (address of CE1). In addition, the agent server must run the agent software.

Step 6 Verify the configuration.

Run the **display ip routing-table vpn-instance vpn1** command on PE1 to check the VPN routing table of vpn1. The VPN routing table has a default route with the next hop address 100.1.1.2 and the outbound interface GE2/0/0.

```
[PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
  Destinations : 7          Routes : 7
  Destination/Mask  Proto Pre Cost      Flags NextHop      Interface
  0.0.0.0/0         Static 60   0          RD   100.1.1.2
GigabitEthernet2/0/0
  10.1.1.0/24       Direct 0     0          D    10.1.1.2
GigabitEthernet1/0/0
  10.1.1.2/32       Direct 0     0          D    127.0.0.1
GigabitEthernet1/0/0
  10.1.1.255/32     Direct 0     0          D    127.0.0.1
GigabitEthernet1/0/0
  10.2.1.0/24       IBGP   255   0          RD   3.3.3.3
GigabitEthernet2/0/0
  10.3.1.0/24       EBGP   255   0          D    10.1.1.1
GigabitEthernet1/0/0
  255.255.255.255/32 Direct 0     0          D    127.0.0.1      InLoopBack0
```

Run the **display ip routing-table** command on PE1 to check the IP routing table on PE1. The routing table has a route to the agent server, in which the next hop address is 10.1.1.1.

```
[PE1] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
  Destinations : 12        Routes : 12
  Destination/Mask  Proto Pre Cost      Flags NextHop      Interface
  1.1.1.1/32        Direct 0     0          D    127.0.0.1      LoopBack1
  2.2.2.2/32        OSPF  10    1          D    100.1.1.2
GigabitEthernet2/0/0
  3.3.3.3/32        OSPF  10    2          D    100.1.1.2
GigabitEthernet2/0/0
  100.1.1.0/24       Direct 0     0          D    100.1.1.1
GigabitEthernet2/0/0
  100.1.1.1/32       Direct 0     0          D    127.0.0.1
GigabitEthernet2/0/0
  100.1.1.255/32     Direct 0     0          D    127.0.0.1
GigabitEthernet2/0/0
  100.2.1.0/24       OSPF  10    2          D    100.1.1.2
GigabitEthernet2/0/0
  10.3.1.0/24       Static 60    0          RD   10.1.1.1
GigabitEthernet1/0/0
  127.0.0.0/8        Direct 0     0          D    127.0.0.1      InLoopBack0
  127.0.0.1/32       Direct 0     0          D    127.0.0.1      InLoopBack0
  127.255.255.255/32 Direct 0     0          D    127.0.0.1      InLoopBack0
  255.255.255.255/32 Direct 0     0          D    127.0.0.1      InLoopBack0
```

P can ping the agent server.

```
[P] ping 10.3.1.1
PING 10.3.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.1: bytes=56 Sequence=1 ttl=253 time=1 ms
  Reply from 10.3.1.1: bytes=56 Sequence=2 ttl=253 time=1 ms
  Reply from 10.3.1.1: bytes=56 Sequence=3 ttl=253 time=1 ms
  Reply from 10.3.1.1: bytes=56 Sequence=4 ttl=253 time=1 ms
  Reply from 10.3.1.1: bytes=56 Sequence=5 ttl=253 time=1 ms

--- 10.3.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

The agent server can access the P on the Internet.

----End

Configuration Files

- CE1 configuration file

```
#
 sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.3.1.2 255.255.255.0
#
bgp 65410
 peer 10.1.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.1.1.2 enable
#
ip route-static 0.0.0.0 0.0.0.0 10.1.1.2
#
return
```

- PE1 configuration file

```
#
 sysname PE1
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
 mpls lsr-id 1.1.1.1
 mpls
#
 mpls ldp
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpn1
 ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 100.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 3.3.3.3 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 3.3.3.3 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.1.1.1 as-number 65410
  import-route static
  import-route direct
#
ospf 1
```

```
import-route static
area 0.0.0.0
 network 1.1.1.1 0.0.0.0
 network 100.1.1.0 0.0.0.255
#
ip route-static 10.3.1.0 255.255.255.0 vpn-instance vpn1 10.1.1.1
ip route-static vpn-instance vpn1 0.0.0.0 0.0.0.0 100.1.1.2 public
#
return
```

● P configuration file

```
#
sysname P
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 100.2.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 2.2.2.2 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 100.1.1.0 0.0.0.255
  network 100.2.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:2
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
mpls lsr-id 3.3.3.3
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 100.2.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance vpn1
 ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
 ip address 3.3.3.3 255.255.255.255
#
bgp 100
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack1
```

```
#
ipv4-family unicast
 undo synchronization
 peer 1.1.1.1 enable
#
ipv4-family vpnv4
 policy vpn-target
 peer 1.1.1.1 enable
#
ipv4-family vpn-instance vpn1
 peer 10.2.1.1 as-number 65420
 import-route direct
#
ospf 1
 area 0.0.0.0
 network 3.3.3.3 0.0.0.0
 network 100.2.1.0 0.0.0.255
#
return
```

- CE2 configuration file

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
bgp 65420
 peer 10.2.1.2 as-number 100
#
ipv4-family unicast
 undo synchronization
 import-route direct
 peer 10.2.1.2 enable
#
return
```

8.9.21 Example for Configuring BGP/MPLS IP VPN to Use a GRE Tunnel

Networking Requirements

 **NOTE**

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot be used in this scenario.

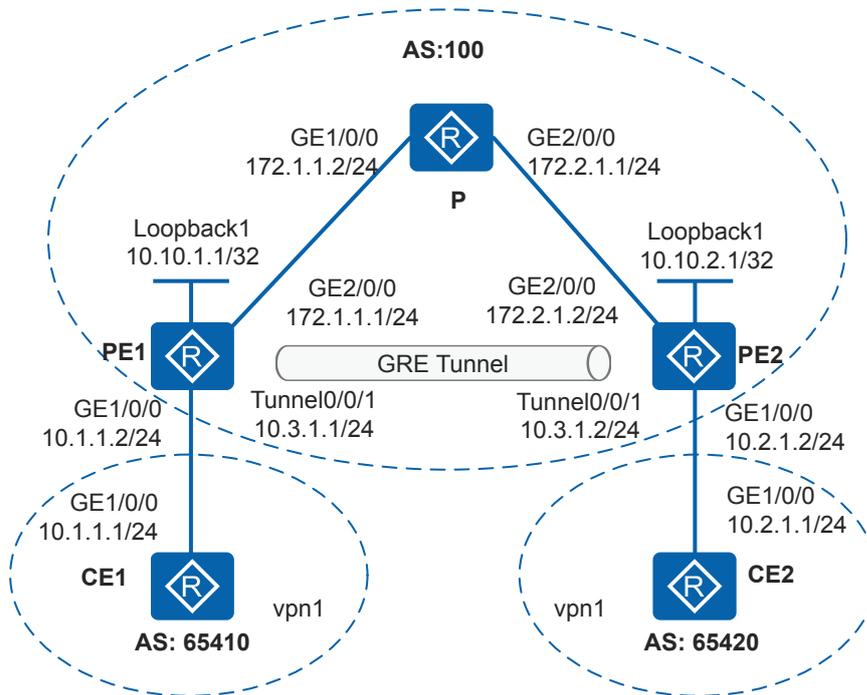
In [Figure 8-62](#):

- Branch 1 connects to the VPN backbone network through CE1 and PE1.
- Branch 2 connects to the VPN backbone network through CE2 and PE2.

On the backbone network, PEs provide MPLS functions, and the P does not provide MPLS functions.

The enterprise wants to establish a GRE tunnel between the PEs and use IP to forward VPN packets over the IP network.

Figure 8-62 Networking diagram for configuring BGP/MPLS IP VPN to use a GRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the PEs and P to implement IP connectivity on the backbone network.
2. Create a GRE tunnel between PEs so that VPN packets can be transmitted over the GRE tunnel.
3. Configure VPN instances on PEs and bind each PE interface connected to a CE to a VPN instance.
4. Because the P device does not support MPLS functions, an LSP cannot be used to transmit VPN packets. Configure a tunnel policy on the PEs to specify that VPN packets are transmitted over a GRE tunnel, and apply the tunnel policy.
5. Establish EBGP peer relationships between PEs and CEs to exchange routes so that a CE can learn routes from the peer CE and CE1 can communicate with CE2.

Procedure

Step 1 Configure an IP address for each interface.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
```

```
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24  
[CE1-GigabitEthernet1/0/0] quit
```

Configure IP addresses for interfaces on PE1 except the interface to be bound to a VPN instance. This is because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view  
[Huawei] sysname PE1  
[PE1] interface gigabitethernet 2/0/0  
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 24  
[PE1-GigabitEthernet2/0/0] quit  
[PE1] interface loopback 1  
[PE1-LoopBack1] ip address 10.10.1.1 32  
[PE1-LoopBack1] quit
```

Configure the P device.

```
<Huawei> system-view  
[Huawei] sysname P  
[P] interface gigabitethernet 1/0/0  
[P-GigabitEthernet1/0/0] ip address 172.1.1.2 24  
[P-GigabitEthernet1/0/0] quit  
[P] interface gigabitethernet 2/0/0  
[P-GigabitEthernet2/0/0] ip address 172.2.1.1 24  
[P-GigabitEthernet2/0/0] quit
```

Configure IP addresses for interfaces on PE2 except the interface to be bound to a VPN instance. This is because all configurations on this interface are deleted when the interface is bound to a VPN instance.

```
<Huawei> system-view  
[Huawei] sysname PE2  
[PE2] interface gigabitethernet 2/0/0  
[PE2-GigabitEthernet2/0/0] ip address 172.2.1.2 24  
[PE2-GigabitEthernet2/0/0] quit  
[PE2] interface loopback 1  
[PE2-LoopBack1] ip address 10.10.2.1 32  
[PE2-LoopBack1] quit
```

Configure CE2.

```
<Huawei> system-view  
[Huawei] sysname CE2  
[CE2] interface gigabitethernet 1/0/0  
[CE2-GigabitEthernet1/0/0] ip address 10.2.1.1 24  
[CE2-GigabitEthernet1/0/0] quit
```

Step 2 Configure IGP on the MPLS backbone network to implement interworking between PEs.

Configure PE1.

```
[PE1] ospf 1  
[PE1-ospf-1] area 0  
[PE1-ospf-1-area-0.0.0.0] network 10.10.1.1 0.0.0.0  
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255  
[PE1-ospf-1-area-0.0.0.0] quit  
[PE1-ospf-1] quit
```

Configure the P device.

```
[P] ospf 1  
[P-ospf-1] area 0  
[P-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255  
[P-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255  
[P-ospf-1-area-0.0.0.0] quit  
[P-ospf-1] quit
```

Configure PE2.

```
[PE2] ospf 1
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 10.10.2.1 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

After the configurations are complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

Step 3 Configure a GRE tunnel.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] source loopback 1
[PE1-Tunnel0/0/1] destination 10.10.2.1
[PE1-Tunnel0/0/1] ip address 10.3.1.1 24
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] tunnel-protocol gre
[PE2-Tunnel0/0/1] source loopback 1
[PE2-Tunnel0/0/1] destination 10.10.1.1
[PE2-Tunnel0/0/1] ip address 10.3.1.2 24
[PE2-Tunnel0/0/1] quit
```

Step 4 Enable basic MPLS functions on the PEs.

Configure PE1.

```
[PE1] mpls lsr-id 10.10.1.1
[PE1] mpls
[PE1-mpls] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 10.10.2.1
[PE2] mpls
[PE2-mpls] quit
```

Step 5 Configure VPN instances on PEs and bind each interface that connects a PE to a CE to a VPN instance. Apply tunnel policies on the PEs to specify the GRE tunnel used to forward VPN packets.

Configure PE1.

```
[PE1] tunnel-policy gre1
[PE1-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE1-tunnel-policy-gre1] quit
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE1-vpn-instance-vpn1-af-ipv4] tnl-policy gre1
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] tunnel-policy gre1
[PE2-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE2-tunnel-policy-gre1] quit
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE2-vpn-instance-vpn1-af-ipv4] tnl-policy gre1
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet1/0/0] ip address 10.2.1.2 24
[PE2-GigabitEthernet1/0/0] quit
```

After the configurations are complete, run the **display ip vpn-instance verbose** command on PEs to view the configurations of VPN instances. Each PE can ping its local CE.

NOTE

If a PE has multiple interfaces bound to the same VPN instance, specify a source IP address by setting **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping a remote CE. If the source IP address is not specified, the ping operation fails.

Step 6 Set up EBGP peer relationships between the PEs and CEs and import VPN routes to EBGP.

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65420
[CE2-bgp] peer 10.2.1.2 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 as-number 65420
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

After the configurations are complete, run the **display bgp vpnv4 vpn-instance peer** command on PEs. You can see that BGP peer relationships have been established between PEs and CEs and are in **Established** state.

The command output on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 10.10.1.1
Local AS number : 100
```

```

VPN-Instance vpn1, Router ID 10.10.1.1:
Total number of peers : 1                Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.1.1.1     4          65410    6         3       0 00:01:14
Established    3
  
```

Step 7 Set up an MP-IBGP peer relationship between PEs.

Configure PE1.

```

[PE1] bgp 100
[PE1-bgp] peer 10.10.2.1 as-number 100
[PE1-bgp] peer 10.10.2.1 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 10.10.2.1 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
  
```

Configure PE2.

```

[PE2] bgp 100
[PE2-bgp] peer 10.10.1.1 as-number 100
[PE2-bgp] peer 10.10.1.1 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.10.1.1 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
  
```

After the configurations are complete, run the **display bgp vpnv4 all peer** command on a PE. The command output shows that the BGP peer relationships have been established between the PEs and are in the **Established** state.

```

[PE1] display bgp vpnv4 all peer

BGP local router ID : 10.10.1.1
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.10.2.1     4          100     4         7       0 00:02:54
Established    0

  Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 10.10.1.1:
  10.1.1.1     4          65410    122        119       0 01:57:43
Established    3
  
```

Step 8 Verify the configuration.

After the configuration is complete, CEs can learn routes to each other. CEs can successfully ping each other.

The command output on CE1 is used as an example.

```

[CE1] display ip routing-table 10.2.1.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto   Pre  Cost           Flags NextHop           Interface
-----
10.2.1.0/24        EBGP    255  0              D    10.1.1.2
  
```

```
GigabitEthernet1/0/0

[CE1] ping 10.2.1.1
  PING 10.2.1.1: 56 data bytes, press CTRL_C to break
    Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=253 time=10 ms
    Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=253 time=1 ms

--- 10.2.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/2/10 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
bgp 65410
peer 10.1.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.1.1.2 enable
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:1
tnl-policy gre1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
#
mpls lsr-id 10.10.1.1
mpls
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.1.1.1 255.255.255.0
#
interface LoopBack1
ip address 10.10.1.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.3.1.1 255.255.255.0
tunnel-protocol gre
source LoopBack1
destination 10.10.2.1
#
tunnel-policy gre1
```

```
tunnel select-seq gre load-balance-number 1
#
bgp 100
peer 10.10.2.1 as-number 100
peer 10.10.2.1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 10.10.2.1 enable
#
ipv4-family vpnv4
policy vpn-target
peer 10.10.2.1 enable
#
ipv4-family vpn-instance vpn1
peer 10.1.1.1 as-number 65410
import-route direct
#
ospf 1
area 0.0.0.0
network 10.10.1.1 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

- Configuration file of the P device

```
#
sysname P
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.2.1.1 255.255.255.0
#
ospf 1
area 0.0.0.0
network 172.1.1.0 0.0.0.255
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
ip vpn-instance vpn1
ipv4-family
route-distinguisher 100:2
tnl-policy gre1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
#
mpls lsr-id 10.10.2.1
mpls
#
interface GigabitEthernet1/0/0
ip binding vpn-instance vpn1
ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
ip address 172.2.1.2 255.255.255.0
#
interface LoopBack1
ip address 10.10.2.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.3.1.2 255.255.255.0
tunnel-protocol gre
source LoopBack1
```

```
destination 10.10.1.1
#
tunnel-policy gre1
tunnel select-seq gre load-balance-number 1
#
bgp 100
peer 10.10.1.1 as-number 100
peer 10.10.1.1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 10.10.1.1 enable
#
ipv4-family vpnv4
policy vpn-target
peer 10.10.1.1 enable
#
ipv4-family vpn-instance vpn1
peer 10.2.1.1 as-number 65420
#
ospf 1
area 0.0.0.0
network 10.10.2.1 0.0.0.0
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.2.1.1 255.255.255.0
#
bgp 65420
peer 10.2.1.2 as-number 100
#
ipv4-family unicast
undo synchronization
import-route direct
peer 10.2.1.2 enable
#
return
```

8.9.22 Example for Configuring L3VPN Using LDP Signaling over GRE

Networking Requirements

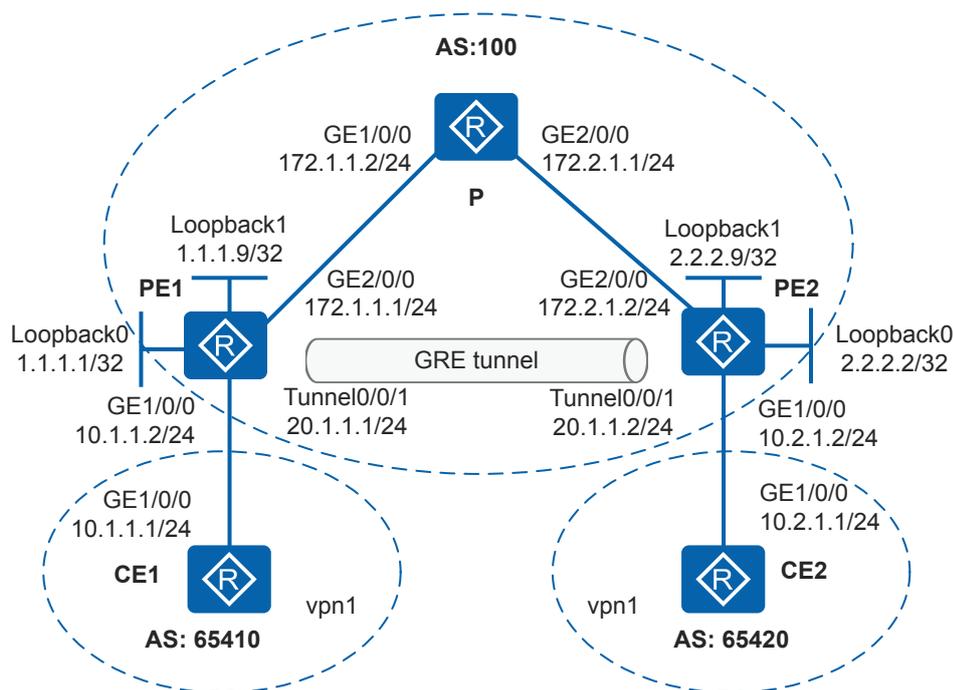
In [Figure 8-63](#):

- Branch 1 connects to the VPN backbone network through CE1 and PE1.
- Branch 2 connects to the VPN backbone network through CE2 and PE2.

On the backbone network, PEs provide MPLS functions, and the P does not provide MPLS functions.

The enterprise wants to deploy BGP/MPLS IP VPN between PE1 and PE2 and use LDP LSPs to transmit VPN data so that CE1 can communicate with CE2.

Figure 8-63 Networking for configuring L3VPN using LDP signaling over



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between the PEs and P to implement IP connectivity on the backbone network.
2. Configure basic MPLS functions and MPLS LDP on PEs so that MPLS LSPs can be established to transmit VPN data.
3. Because the P device does not support MPLS functions and an LSP is required to transmit VPN data, use LDP over GRE so that a GRE tunnel is set up between PEs to transmit services over LDP LSPs. Create GRE tunnel interfaces on PEs, specify source and destination addresses of the tunnel, and establish a GRE tunnel between PEs to implement interworking on the MPLS network.
4. Enable MPLS LDP on tunnel interfaces to implement LDP over GRE and establish MPLS LSPs.
5. Configure VPN instances on PEs and bind each PE interface connected to a CE to a VPN instance.
6. Establish an MP-IBGP peer relationship between PE1 and PE2, and establish EBGP peer relationships between PEs and CEs and import VPN routes, so that CE1 can communicate with CE2.

NOTE

The IP address of Loopback1 interface is used as the LSR ID, that is, LDP uses this IP address to establish a session. A GRE tunnel interface must have an IP address configured, and uses addresses of Loopback0 interfaces as source and destination addresses. The source and destination addresses, and physical interface are advertised by an IGP, and the IP address of Loopback1 interface and tunnel interface address are advertised by another IGP or static route. If a static route is used, specify the tunnel interface as the outbound interface.

Procedure

- Step 1** Configure OSPF between the PEs and P to implement IP connectivity on the backbone network.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 1.1.1.1 32
[PE1-LoopBack0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configurations of PE2 and P are similar to the configuration of PE1, and are not mentioned here.

After the configurations are complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

- Step 2** Enable basic MPLS functions and MPLS LDP on PEs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 2.2.2.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
```

- Step 3** Create GRE tunnel interfaces on PEs, and specify source and destination addresses of the tunnel.

Create and configure GRE tunnel interfaces on PE1 and PE2, and establish a GRE tunnel between PEs to implement interworking on the MPLS network.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] ip address 20.1.1.1 24
[PE1-Tunnel0/0/1] source loopback 0
[PE1-Tunnel0/0/1] destination 2.2.2.2
[PE1-Tunnel0/0/1] quit
[PE1] ospf 11
```

```
[PE1-ospf-11] area 0
[PE1-ospf-11-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-11-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[PE1-ospf-11-area-0.0.0.0] quit
[PE1-ospf-11] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] tunnel-protocol gre
[PE2-Tunnel0/0/1] ip address 20.1.1.2 24
[PE2-Tunnel0/0/1] source loopback 0
[PE2-Tunnel0/0/1] destination 1.1.1.1
[PE2-Tunnel0/0/1] quit
[PE2] ospf 11
[PE2-ospf-11] area 0
[PE2-ospf-11-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[PE2-ospf-11-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[PE2-ospf-11-area-0.0.0.0] quit
[PE2-ospf-11] quit
```

After the configurations are complete, a GRE tunnel is set up between PE1 and PE2. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

Step 4 Enable MPLS LDP on tunnel interfaces of PEs.

Enable MPLS LDP on tunnel interfaces of PE1 and PE2 so that MPLS LSPs can be established.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] mpls
[PE1-Tunnel0/0/1] mpls ldp
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] mpls
[PE2-Tunnel0/0/1] mpls ldp
[PE2-Tunnel0/0/1] quit
```

After the configurations are complete, an LDP session can be set up between PE1 and PE2. Run the **display mpls ldp session** command. You can see that the **Status** field is **Operational** in the command output.

Step 5 Configure a VPN instance on each PE and connect CEs to PEs.

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/0] ip address 10.1.1.2 24
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpn1
[PE2-vpn-instance-vpn1] ipv4-family
[PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:2
```

```
[PE2-vpn-instance-vpn1-af-ipv4] vpn-target 100:1 both
[PE2-vpn-instance-vpn1-af-ipv4] quit
[PE2-vpn-instance-vpn1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip binding vpn-instance vpn1
[PE2-GigabitEthernet1/0/0] ip address 10.2.1.2 24
[PE2-GigabitEthernet1/0/0] quit
```

Configure IP addresses for CE interfaces according to [Figure 8-63](#).

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
```

The configuration of CE2 is similar to that of CE1, and is not mentioned here.

After the configurations are complete, run the **display ip vpn-instance verbose** command on PEs to view the configurations of VPN instances. Each PE can successfully ping the connected CE.

NOTE

If multiple interfaces on a PE is bound to the same VPN instance, specify a source IP addresses by specifying **-a source-ip-address** in the **ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address** command to ping the remote CE. Otherwise, the ping operation fails.

Step 6 Set up EBGP peer relationships between the PEs and CEs and import VPN routes to EBGP.

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] peer 10.1.1.1 as-number 65410
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65420
[CE2-bgp] peer 10.2.1.2 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn1
[PE2-bgp-vpn1] peer 10.2.1.1 as-number 65420
[PE2-bgp-vpn1] quit
[PE2-bgp] quit
```

After the configurations are complete, run the **display bgp vpnv4 vpn-instance peer** command on PEs. You can see that BGP peer relationships have been established between PEs and CEs and are in **Established** state.

The display on PE1 is used as an example.

```
[PE1] display bgp vpnv4 vpn-instance vpn1 peer

BGP local router ID : 1.1.1.9
Local AS number : 100

VPN-Instance vpn1, Router ID 1.1.1.9:
Total number of peers : 1                Peers in established state : 1

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  10.1.1.1     4          65410    6         3       0 00:01:14
Established    3
```

Step 7 Set up an MP-IBGP peer relationship between PEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.9 enable
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.9 as-number 100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[PE2-bgp-af-vpnv4] quit
[PE2-bgp] quit
```

After the configurations are complete, run the **display bgp vpnv4 all peer** command on a PE. You can see that the BGP peer relationship between PEs is in Established state.

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  2.2.2.9       4          100     4         7       0 00:02:54
Established    0

  Peer of IPv4-family for vpn instance :

VPN-Instance vpn1, Router ID 1.1.1.9:
  10.1.1.1     4          65410   122       119     0 01:57:43
Established    3
```

Step 8 Verify the configuration.

After the configurations are complete, CEs can learn routes to the interface of each other, and can ping each other successfully.

The display on CE1 is used as an example.

```
[CE1] display ip routing-table 10.2.1.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
```

```

Summary Count : 1
Destination/Mask    Proto    Pre    Cost    Flags NextHop          Interface
      10.2.1.0/24  EBGP    255    0            D   10.1.1.2
GigabitEthernet1/0/0

[CE1] ping 10.2.1.1
  PING 10.2.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.2.1.1: bytes=56 Sequence=1 ttl=253 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=2 ttl=253 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=3 ttl=253 time=1 ms
  Reply from 10.2.1.1: bytes=56 Sequence=4 ttl=253 time=10 ms
  Reply from 10.2.1.1: bytes=56 Sequence=5 ttl=253 time=1 ms

--- 10.2.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/2/10 ms
  
```

---End

Configuration Files

- CE1 configuration file

```

#
 sysname CE1
#
 interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
 bgp 65410
 peer 10.1.1.2 as-number 100
#
 ipv4-family unicast
 undo synchronization
 import-route direct
 peer 10.1.1.2 enable
#
 return
  
```

- PE1 configuration file

```

#
 sysname PE1
#
 ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 100:1 export-extcommunity
  vpn-target 100:1 import-extcommunity
#
 mpls lsr-id 1.1.1.9
 mpls
#

 mpls
 ldp
#
 interface GigabitEthernet1/0/0
 ip binding vpn-instance vpn1
 ip address 10.1.1.2 255.255.255.0
#
 interface GigabitEthernet2/0/0
 ip address 172.1.1.1 255.255.255.0
#
 interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
  
```

```
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
interface Tunnel0/0/1
 ip address 20.1.1.1 255.255.255.0
 tunnel-protocol gre
 source LoopBack0
 destination 2.2.2.2
 mpls
 mpls ldp
#
bgp 100
 peer 2.2.2.9 as-number 100
 peer 2.2.2.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
 peer 2.2.2.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
 peer 2.2.2.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.1.1.1 as-number 65410
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 1.1.1.1 0.0.0.0
  network 172.1.1.0 0.0.0.255
#
ospf 11
 area 0.0.0.0
  network 1.1.1.9 0.0.0.0
  network 20.1.1.0 0.0.0.255
#
return
```

● P configuration file

```
#
 sysname P
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 172.2.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 172.1.1.0 0.0.0.255
  network 172.2.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
 sysname PE2
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 100:2
  vpn-target 100:1 export-extcommunity
  vpn-target 100:1 import-extcommunity
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
```

```
#
interface GigabitEthernet1/0/0
 ip binding vpn-instance vpn1
 ip address 10.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 172.2.1.2 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
interface Tunnel0/0/1
 ip address 20.1.1.2 255.255.255.0
 tunnel-protocol gre
 source LoopBack0
 destination 1.1.1.1
 mpls
 mpls ldp
#
bgp 100
 peer 1.1.1.9 as-number 100
 peer 1.1.1.9 connect-interface LoopBack1
#
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.9 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.9 enable
#
 ipv4-family vpn-instance vpn1
  peer 10.2.1.1 as-number 65420
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 172.2.1.0 0.0.0.255
#
ospf 11
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 20.1.1.0 0.0.0.255
#
return
```

● CE2 configuration file

```
#
 sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 10.2.1.1 255.255.255.0
#
bgp 65420
 peer 10.2.1.2 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.2.1.2 enable
#
return
```

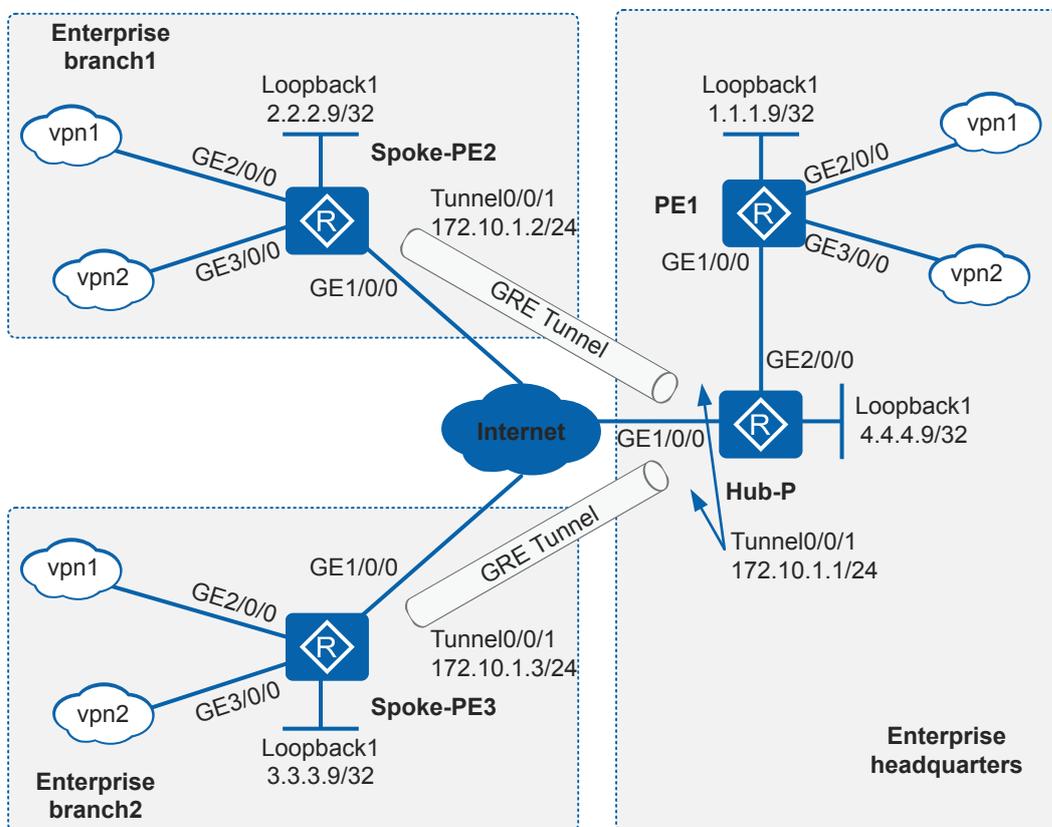
8.9.23 Example for Configuring L3VPN with LDP Signals Carried by DSVPN

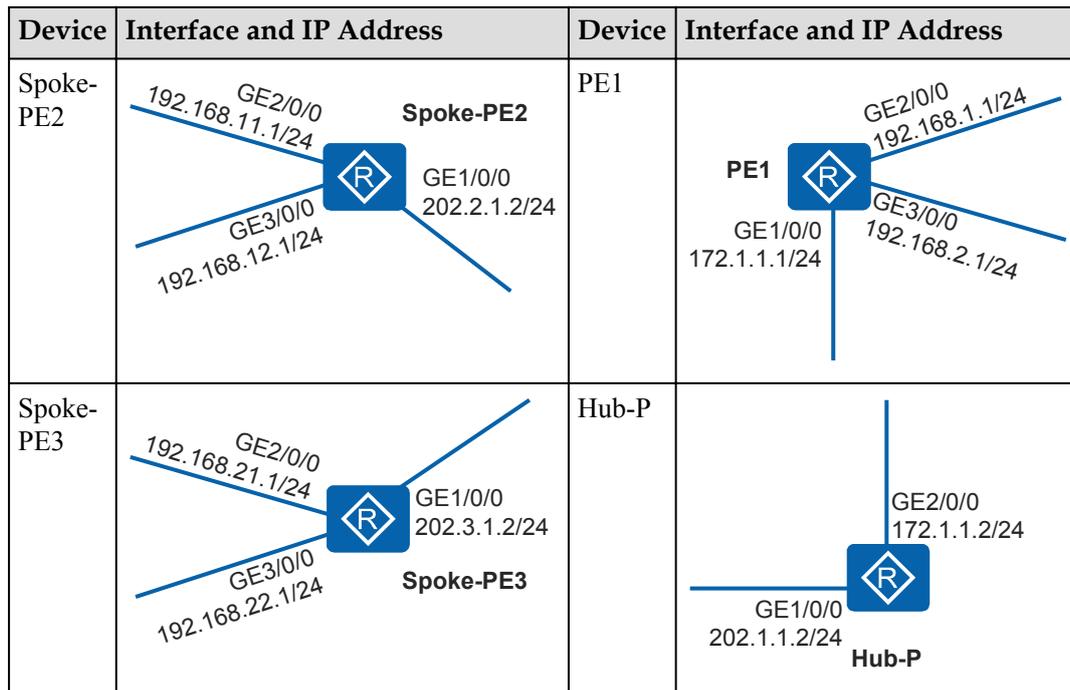
Networking Requirements

As shown in **Figure 8-64**, a large-scale enterprise has deployed a production network **vpn1** and an office network **vpn2** in the headquarters and branches respectively. The enterprise establishes an IP/MPLS backbone network in its headquarters, and its branches located in different areas use Spoke-PE to connect to the IP/MPLS backbone network through the Internet. In this example, the backbone network has only a Hub-P and PE1 and the enterprise has only two branches. Spoke-PE2 and Spoke-PE3 in branches dynamically obtain their public addresses. (Configurations related to dynamic address allocation is omitted in this example and public addresses are manually specified.) Because the Internet cannot provide the MPLS function for the enterprise, the production networks and office networks in branches cannot communicate with those in the headquarters.

The enterprise wants to expand the IP/MPLS backbone network, deploy BGP/MPLS IP VPN in the headquarters and branches, and use LDP LSP to transmit data from **vpn1** and **vpn2** to implement secure interconnection between the headquarters and branches and between branches. VPN data between branches needs to be forwarded by the headquarters so that the headquarters can monitor traffic in real-time.

Figure 8-64 Networking diagram for configuring L3VPN with LDP signals carried by DSVPN





Configuration Roadmap

To expand the IP/MPLS backbone network and deploy BGP/MPLS IP VPN for an enterprise, you need to add the Spoke-PE devices in the branches to the IP/MPLS backbone network in the headquarters. MPLS LDP packets between the headquarters and branches need to be transmitted over GRE tunnels because the Internet cannot provide the MPLS function. As there are a large number of branches and devices in the branches dynamically obtain their public addresses, DSVPN is used to establish GRE tunnels between the headquarters and branches. As a result, L3VPN with LDP signals carried by DSVPN can meet the requirements of the enterprise.

The configuration roadmap for L3VPN with LDP signals carried by DSVPN is as follows:

1. Configure branch devices to save only summarized routes to the headquarters, configure OSPF on Hub-P and Spoke-PEs to advertise routes, and set the OSPF network type to point-to-multipoint (P2MP), so that all VPN data between branches is forwarded by the headquarters.
2. Enable MPLS LDP on tunnel interfaces of Spoke-PE2, Spoke-PE3, and Hub-P and set up MPLS LSP tunnels to implement LDP over mGRE.
3. Configure L3VPN on Spoke-PE2, Spoke-PE3, and PE1 to implement secure interconnection between the headquarters and branches and between branches. Because there are a large number of branches, a route reflector can be used to reduce the number of MP-IBGP connections between PEs.

NOTE

Do not configure NHRP redirection on the Hub because LDP over mGRE does not need to establish tunnels for direct communication between branches.

Procedure

- Step 1** Configure interface IP addresses and OSPF on Hub-P and PE1 to implement interconnection on the IP/MPLS backbone network.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.1 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration of Hub-P is similar to that of PE1, and is not mentioned here.

After the configuration is complete, an OSPF neighbor relationship can be set up between Hub-P and PE1. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that Hub-P and PE1 have learnt the routes to Loopback1 of each other.

Step 2 Configure interface IP addresses and static routes on Hub-P, Spoke-PE2, and Spoke-PE3 to ensure that public routes are reachable.

Because Hub-P, Spoke-PE2, and Spoke-PE3 are directly connected to the Internet, IP addresses and default static routes are manually specified here.

Configure Spoke-PE2.

```
<Huawei> system-view
[Huawei] sysname Spoke-PE2
[Spoke-PE2] interface gigabitethernet 1/0/0
[Spoke-PE2-GigabitEthernet1/0/0] ip address 202.2.1.2 24
[Spoke-PE2-GigabitEthernet1/0/0] quit
[Spoke-PE2] interface loopback 1
[Spoke-PE2-LoopBack1] ip address 2.2.2.9 32
[Spoke-PE2-LoopBack1] quit
[Spoke-PE2] ip route-static 0.0.0.0 0 202.2.1.1
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

After the configuration is complete, devices can ping each other and public routes are reachable.

Step 3 Create tunnel interfaces and configure DSVPN on Hub-P, Spoke-PE2, and Spoke-PE3.

1. Create an mGRE interface, configure an IP address, and specify a source tunnel interface.

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] ip address 172.10.1.2 24
[Spoke-PE2-Tunnel0/0/1] tunnel-protocol gre p2mp
[Spoke-PE2-Tunnel0/0/1] source gigabitethernet 1/0/0
[Spoke-PE2-Tunnel0/0/1] quit
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

2. Configure OSPF to advertise the MPLS LSR ID as DSVPN subnet information through the tunnel interface.

Configure Spoke-PE2.

```
[Spoke-PE2] ospf 1
[Spoke-PE2-ospf-1] area 0
[Spoke-PE2-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[Spoke-PE2-ospf-1-area-0.0.0.0] network 172.10.1.0 0.0.0.255
[Spoke-PE2-ospf-1-area-0.0.0.0] quit
[Spoke-PE2-ospf-1] quit
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

3. Configure NHRP and set the OSPF network type to P2MP. Do not configure NHRP redirection on the Hub-P.

Configure Hub-P.

```
[Hub-P] interface tunnel 0/0/1
[Hub-P-Tunnel0/0/1] nhrp entry multicast dynamic
[Hub-P-Tunnel0/0/1] ospf network-type p2mp
[Hub-P-Tunnel0/0/1] ospf dr-priority 100
[Hub-P-Tunnel0/0/1] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] nhrp entry 172.10.1.1 202.1.1.2 register
[Spoke-PE2-Tunnel0/0/1] ospf network-type p2mp
[Spoke-PE2-Tunnel0/0/1] ospf dr-priority 0
[Spoke-PE2-Tunnel0/0/1] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] interface tunnel 0/0/1
[Spoke-PE3-Tunnel0/0/1] nhrp entry 172.10.1.1 202.1.1.2 register
[Spoke-PE3-Tunnel0/0/1] ospf network-type p2mp
[Spoke-PE3-Tunnel0/0/1] ospf dr-priority 0
[Spoke-PE3-Tunnel0/0/1] quit
```

After the configuration is complete, run the **display nhrp peer all** command on Hub-P to view registration information about Spoke-PE2 and Spoke-PE3.

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.10.1.2     32    202.2.1.2      172.10.1.2   dynamic   route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time   : 00:02:36
Expire time    : 01:57:24
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.10.1.3     32    202.3.1.2      172.10.1.3   dynamic   route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time   : 00:00:04
Expire time    : 01:59:56
-----
Number of nhrp peers: 2
```

Run the **display ip routing-table** command on all devices on the IP/MPLS backbone network. You can see that all devices have learnt the routes to Loopback1 of other devices.

- Step 4** Enable basic MPLS functions and MPLS LDP on Spoke-PE2, Spoke-PE3, Hub-P, and PE1.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
```

The configurations of Spoke-PE2, Spoke-PE3 and Hub-P are similar to that of PE1, and are not mentioned here.

Step 5 Enable MPLS LDP on the interfaces of Spoke-PE2, Spoke-PE3, Hub-P, and PE1.

Enable MPLS LDP on interfaces of Hub-P and PE1 that are directly connected to each other and enable MPLS LDP on tunnel interfaces of Spoke-PE2, Spoke-PE3 and Hub-P to establish MPLS LSP tunnels.

Configure PE1.

```
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure Hub-P.

```
[Hub-P] interface gigabitethernet 2/0/0
[Hub-P-GigabitEthernet2/0/0] mpls
[Hub-P-GigabitEthernet2/0/0] mpls ldp
[Hub-P-GigabitEthernet2/0/0] quit
[Hub-P] interface tunnel 0/0/1
[Hub-P-Tunnel0/0/1] mpls
[Hub-P-Tunnel0/0/1] mpls ldp
[Hub-P-Tunnel0/0/1] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] mpls
[Spoke-PE2-Tunnel0/0/1] mpls ldp
[Spoke-PE2-Tunnel0/0/1] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] interface tunnel 0/0/1
[Spoke-PE3-Tunnel0/0/1] mpls
[Spoke-PE3-Tunnel0/0/1] mpls ldp
[Spoke-PE3-Tunnel0/0/1] quit
```

After the configuration is complete, PE1, Spoke-PE2, and Spoke-PE3 can establish LDP sessions with Hub-P. Run the **display mpls ldp session** command. You can see that the MPLS LDP session status is **Operational**.

Step 6 Configure VPN instances on Spoke-PE2, Spoke-PE3, and PE1 and bind VPN instances to interfaces.

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] ipv4-family
[PE1-vpn-instance-vpn2-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[PE1-vpn-instance-vpn2-af-ipv4] quit
[PE1-vpn-instance-vpn2] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet2/0/0] ip address 192.168.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
```

```
[PE1-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
[PE1-GigabitEthernet3/0/0] ip address 192.168.2.1 24
[PE1-GigabitEthernet3/0/0] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] ip vpn-instance vpn1
[Spoke-PE2-vpn-instance-vpn1] ipv4-family
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 200:1
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] quit
[Spoke-PE2-vpn-instance-vpn1] quit
[Spoke-PE2] ip vpn-instance vpn2
[Spoke-PE2-vpn-instance-vpn2] ipv4-family
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] route-distinguisher 200:2
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] quit
[Spoke-PE2-vpn-instance-vpn2] quit
[Spoke-PE2] interface gigabitethernet 2/0/0
[Spoke-PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[Spoke-PE2-GigabitEthernet2/0/0] ip address 192.168.11.1 24
[Spoke-PE2-GigabitEthernet2/0/0] quit
[Spoke-PE2] interface gigabitethernet 3/0/0
[Spoke-PE2-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
[Spoke-PE2-GigabitEthernet3/0/0] ip address 192.168.12.1 24
[Spoke-PE2-GigabitEthernet3/0/0] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] ip vpn-instance vpn1
[Spoke-PE3-vpn-instance-vpn1] ipv4-family
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] route-distinguisher 300:1
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] quit
[Spoke-PE3-vpn-instance-vpn1] quit
[Spoke-PE3] ip vpn-instance vpn2
[Spoke-PE3-vpn-instance-vpn2] ipv4-family
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] route-distinguisher 300:2
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] quit
[Spoke-PE3-vpn-instance-vpn2] quit
[Spoke-PE3] interface gigabitethernet 2/0/0
[Spoke-PE3-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[Spoke-PE3-GigabitEthernet2/0/0] ip address 192.168.21.1 24
[Spoke-PE3-GigabitEthernet2/0/0] quit
[Spoke-PE3] interface gigabitethernet 3/0/0
[Spoke-PE3-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
[Spoke-PE3-GigabitEthernet3/0/0] ip address 192.168.22.1 24
[Spoke-PE3-GigabitEthernet3/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on each device to view the configuration of VPN instances.

Step 7 Set up MP-IBGP peer relationships between Spoke-PE2, Spoke-PE3, and PE1.

Configure PE1 as a route reflector. Spoke-PE2 and Spoke-PE3 can set up MP-IBGP peer relationships with PE1.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] group rr1 internal
[PE1-bgp] peer rr1 connect-interface loopback 1
[PE1-bgp] peer 2.2.2.9 group rr1
[PE1-bgp] peer 3.3.3.9 group rr1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer rr1 enable
[PE1-bgp-af-vpnv4] peer 2.2.2.9 group rr1
[PE1-bgp-af-vpnv4] peer 3.3.3.9 group rr1
```

```
[PE1-bgp-af-vpnv4] reflector cluster-id 100
[PE1-bgp-af-vpnv4] peer rrl reflect-client
[PE1-bgp-af-vpnv4] undo policy vpn-target
[PE1-bgp-af-vpnv4] quit
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] ipv4-family vpn-instance vpn2
[PE1-bgp-vpn2] import-route direct
[PE1-bgp-vpn2] quit
[PE1-bgp] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] bgp 100
[Spoke-PE2-bgp] peer 1.1.1.9 as-number 100
[Spoke-PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[Spoke-PE2-bgp] ipv4-family vpnv4
[Spoke-PE2-bgp-af-vpnv4] peer 1.1.1.9 enable
[Spoke-PE2-bgp-af-vpnv4] quit
[Spoke-PE2-bgp] ipv4-family vpn-instance vpn1
[Spoke-PE2-bgp-vpn1] import-route direct
[Spoke-PE2-bgp-vpn1] quit
[Spoke-PE2-bgp] ipv4-family vpn-instance vpn2
[Spoke-PE2-bgp-vpn2] import-route direct
[Spoke-PE2-bgp-vpn2] quit
[Spoke-PE2-bgp] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] bgp 100
[Spoke-PE3-bgp] peer 1.1.1.9 as-number 100
[Spoke-PE3-bgp] peer 1.1.1.9 connect-interface loopback 1
[Spoke-PE3-bgp] ipv4-family vpnv4
[Spoke-PE3-bgp-af-vpnv4] peer 1.1.1.9 enable
[Spoke-PE3-bgp-af-vpnv4] quit
[Spoke-PE3-bgp] ipv4-family vpn-instance vpn1
[Spoke-PE3-bgp-vpn1] import-route direct
[Spoke-PE3-bgp-vpn1] quit
[Spoke-PE3-bgp] ipv4-family vpn-instance vpn2
[Spoke-PE3-bgp-vpn2] import-route direct
[Spoke-PE3-bgp-vpn2] quit
[Spoke-PE3-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on Spoke-PE2, Spoke-PE3, and PE1. You can see that Spoke-PE2, Spoke-PE3, and PE1 have set up BGP peer relationships with PE1 and are in **Established** state.

The display on PE1 is used as an example:

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
 2.2.2.9      4      100      5        12      0 00:02:00
Established
 3.3.3.9      4      100      5        11      0 00:01:02
Established
```

Step 8 Verify the configuration.

After the configuration is complete, Spoke-PE2, Spoke-PE3, and PE1 can learn the routes to **vpn1** and **vpn2** of each other.

The display on PE1 is used as an example:

```
[PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
      Destinations : 6          Routes : 6

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
 192.168.1.0/24    Direct  0    0        D   192.168.1.1
GigabitEthernet2/0/0
 192.168.1.1/32    Direct  0    0        D   127.0.0.1
GigabitEthernet2/0/0
 192.168.1.255/32  Direct  0    0        D   127.0.0.1
GigabitEthernet2/0/0
 192.168.11.0/24   IBGP    255  0        RD   2.2.2.9
GigabitEthernet1/0/0
 192.168.21.0/24   IBGP    255  0        RD   3.3.3.9
GigabitEthernet1/0/0
255.255.255.255/32 Direct  0    0        D   127.0.0.1      InLoopBack0

[PE1] display ip routing-table vpn-instance vpn2
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn2
      Destinations : 6          Routes : 6

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
 192.168.2.0/24    Direct  0    0        D   192.168.2.1
GigabitEthernet3/0/0
 192.168.2.1/32    Direct  0    0        D   127.0.0.1
GigabitEthernet3/0/0
 192.168.2.255/32  Direct  0    0        D   127.0.0.1
GigabitEthernet3/0/0
 192.168.12.0/24   IBGP    255  0        RD   2.2.2.9
GigabitEthernet1/0/0
 192.168.22.0/24   IBGP    255  0        RD   3.3.3.9
GigabitEthernet1/0/0
255.255.255.255/32 Direct  0    0        D   127.0.0.1      InLoopBack0
```

Devices in the same VPN can successfully ping each other, whereas devices in different VPNs cannot.

The display on Spoke-PE2 is used as an example:

```
[Spoke-PE2] ping -vpn-instance vpn1 -a 192.168.11.1 192.168.1.1
PING 192.168.1.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.1.1: bytes=56 Sequence=1 ttl=254 time=10 ms
  Reply from 192.168.1.1: bytes=56 Sequence=2 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=3 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=4 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=5 ttl=254 time=1 ms

--- 192.168.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/2/10 ms

[Spoke-PE2] ping -vpn-instance vpn2 -a 192.168.12.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=254 time=10 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=254 time=1 ms

--- 192.168.2.1 ping statistics ---
```

```
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 1/2/10 ms
```

----End

Configuration Files

NOTE

This example does not provide configuration files of devices on the Internet.

- PE1 configuration file

```
#
 sysname PE1
#
ip vpn-instance
vpn1
  ipv4-
  family
    route-distinguisher
    100:1
    vpn-target 111:1 export-
    extcommunity
    vpn-target 111:1 import-
    extcommunity
#

ip vpn-instance
vpn2
  ipv4-
  family
    route-distinguisher
    100:2
    vpn-target 222:2 export-
    extcommunity
    vpn-target 222:2 import-
    extcommunity
#

mpls lsr-id
1.1.1.9
mpls

#

mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1
 255.255.255.0

mpls

mpls ldp
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance
vpn1
 ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance
vpn2
 ip address 192.168.2.1 255.255.255.0
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
```

```
#
bgp
100
  group rr1
  internal
  peer rr1 connect-interface
  LoopBack1
  peer 2.2.2.9 as-number
  100
  peer 2.2.2.9 group
  rr1
  peer 3.3.3.9 as-number
  100
  peer 3.3.3.9 group
  rr1

#

  ipv4-family
  unicast
  undo
  synchronization
  peer rr1
  enable
  peer 2.2.2.9
  enable
  peer 2.2.2.9 group
  rr1
  peer 3.3.3.9
  enable
  peer 3.3.3.9 group
  rr1

#

  ipv4-family
  vpnv4
  reflector cluster-id
  100
  undo policy vpn-
  target
  peer rr1
  enable
  peer rr1 reflect-
  client
  peer 2.2.2.9
  enable
  peer 2.2.2.9 group
  rr1
  peer 3.3.3.9
  enable
  peer 3.3.3.9 group
  rr1

#

  ipv4-family vpn-instance
  vpn1
  import-route
  direct

#

  ipv4-family vpn-instance
  vpn2
  import-route
  direct
#
ospf 1
```

```
area 0.0.0.0
 network 1.1.1.9
0.0.0.0
 network 172.1.1.0 0.0.0.255
#
return
```

● Hub-P configuration file

```
#
 sysname Hub-P
#
mpls lsr-id
4.4.4.9
mpls
#

mpls
 ldp
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 172.1.1.2
255.255.255.0

mpls

 mpls ldp
#
interface
 LoopBack1
 ip address 4.4.4.9
255.255.255.255
#

interface
 Tunnel0/0/1
 ip address 172.10.1.1
255.255.255.0
 tunnel-protocol gre
 p2mp
 source
 GigabitEthernet1/0/0
 ospf network-type
 p2mp
 ospf dr-priority
100

mpls

 mpls
 ldp
 nhrp entry multicast
dynamic
#

ospf
1
 area
0.0.0.0
 network 4.4.4.9
0.0.0.0
 network 172.1.1.0
0.0.0.255
 network 172.10.1.0
0.0.0.255
#
```

```
ip route-static 0.0.0.0 0.0.0.0 202.1.1.1
#
return
```

● Spoke-PE2 configuration file

```
#
 sysname Spoke-PE2
#
ip vpn-instance
vpn1
  ipv4-
  family
    route-distinguisher
    200:1
    vpn-target 111:1 export-
    extcommunity
    vpn-target 111:1 import-
    extcommunity
#
ip vpn-instance
vpn2
  ipv4-
  family
    route-distinguisher
    200:2
    vpn-target 222:2 export-
    extcommunity
    vpn-target 222:2 import-
    extcommunity
#
mpls lsr-id
2.2.2.9
mpls
#
mpls
ldp
#
interface GigabitEthernet1/0/0
 ip address 202.2.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip binding vpn-instance
vpn1
 ip address 192.168.11.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance
vpn2
 ip address 192.168.12.1 255.255.255.0
#
interface
LoopBack1
 ip address 2.2.2.9
255.255.255.255
#
interface
Tunnel0/0/1
 ip address 172.10.1.2
255.255.255.0
 tunnel-protocol gre
p2mp
 source
GigabitEthernet1/0/0
 ospf network-type
p2mp
```

```
ospf dr-priority
0

mpls

mpls
ldp
nhrp entry 172.10.1.1 202.1.1.2
register
#

bgp
100
peer 1.1.1.9 as-number
100
peer 1.1.1.9 connect-interface
LoopBack1
#

ipv4-family
unicast
undo
synchronization
peer 1.1.1.9
enable
#

ipv4-family
vpn4
policy vpn-
target
peer 1.1.1.9
enable
#

ipv4-family vpn-instance
vpn1
import-route
direct
#

ipv4-family vpn-instance
vpn2
import-route
direct
#

ospf
1
area
0.0.0.0
network 2.2.2.9
0.0.0.0
network 172.10.1.0
0.0.0.255
#

ip route-static 0.0.0.0 0.0.0.0 202.2.1.1
#
return
```

● Spoke-PE3 configuration file

```
#
sysname Spoke-PE3
#
ip vpn-instance
```

```
vpn1
  ipv4-
  family
    route-distinguisher
  300:1
    vpn-target 111:1 export-
  extcommunity
    vpn-target 111:1 import-
  extcommunity
  #

ip vpn-instance
vpn2
  ipv4-
  family
    route-distinguisher
  300:2
    vpn-target 222:2 export-
  extcommunity
    vpn-target 222:2 import-
  extcommunity
  #

mpls lsr-id
3.3.3.9
mpls

#

mpls
ldp
#
interface GigabitEthernet1/0/0
  ip address 202.3.1.2 255.255.255.0
  #
interface GigabitEthernet2/0/0
  ip binding vpn-instance
  vpn1
  ip address 192.168.21.1 255.255.255.0
  #
interface GigabitEthernet3/0/0
  ip binding vpn-instance
  vpn2
  ip address 192.168.22.1 255.255.255.0
  #
interface
  LoopBack1
  ip address 3.3.3.9
  255.255.255.255
  #

interface
  Tunnel1/0/1
  ip address 172.10.1.3
  255.255.255.0
  tunnel-protocol gre
  p2mp
  source
  GigabitEthernet1/0/0
  ospf network-type
  p2mp
  ospf dr-priority
  0

mpls

mpls
ldp
nhrp entry 172.10.1.1 202.1.1.2
```

```
register
#

bgp
100
 peer 1.1.1.9 as-number
100
 peer 1.1.1.9 connect-interface
LoopBack1
#

 ipv4-family
unicast
 undo
synchronization
 peer 1.1.1.9
enable
#

 ipv4-family
vpng4
 policy vpn-
target
 peer 1.1.1.9
enable
#

 ipv4-family vpn-instance
vpn1
 import-route
direct
#

 ipv4-family vpn-instance
vpn2
 import-route
direct
#

ospf
1
 area
0.0.0.0
 network 3.3.3.9
0.0.0.0
 network 172.10.1.0
0.0.0.255
#

ip route-static 0.0.0.0 0.0.0.0 202.3.1.1
#
return
```

8.9.24 Example for Configuring L3VPN with LDP Signals Carried by DSVPN and Protected by IPSec

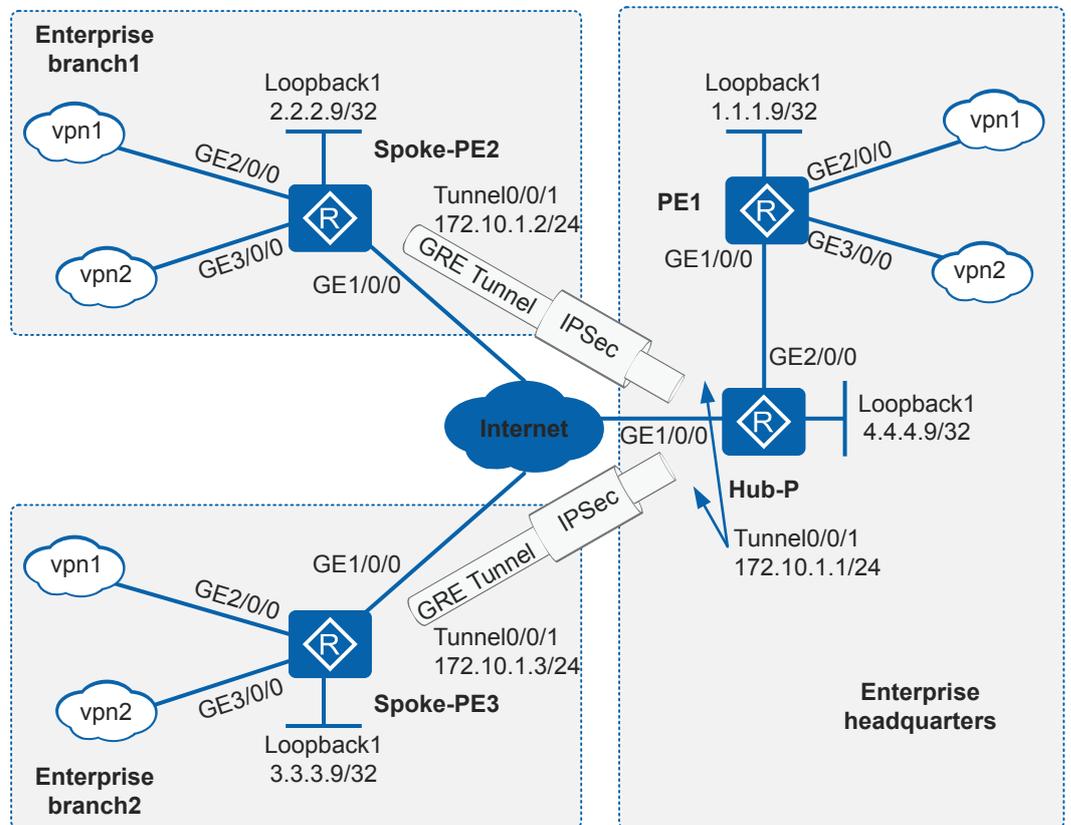
Networking Requirements

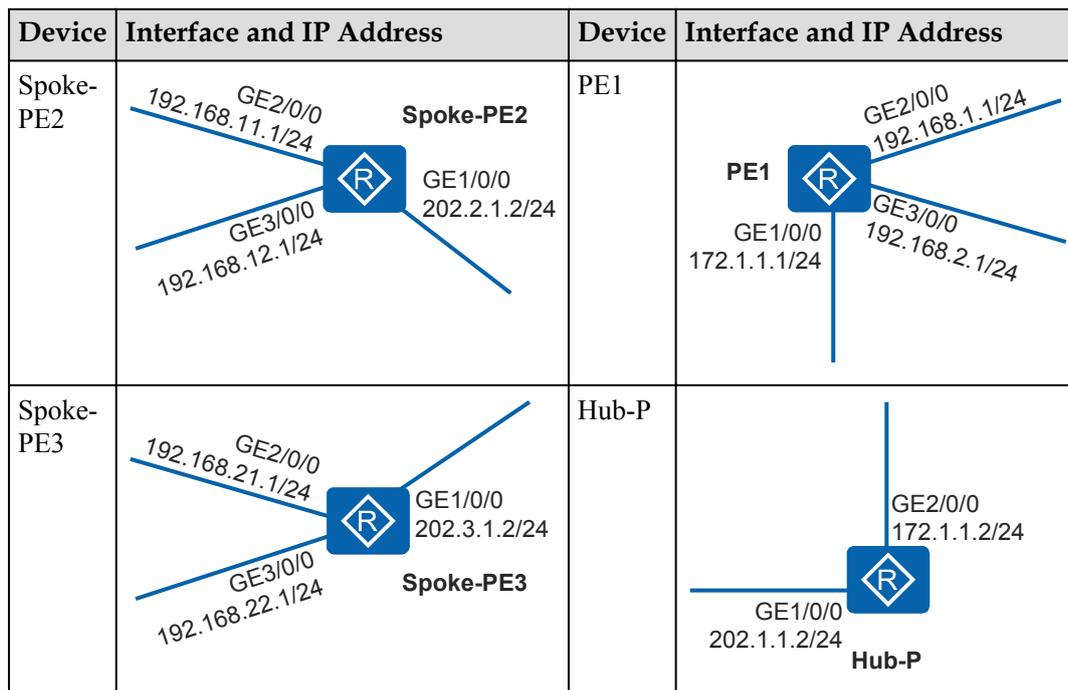
As shown in [Figure 8-65](#), a large-scale enterprise has deployed a production network **vpn1** and an office network **vpn2** in the headquarters and branches respectively. The enterprise establishes an IP/MPLS backbone network in its headquarters, and its branches located in different areas use Spoke-PE to connect to the IP/MPLS backbone network through the

Internet. In this example, the backbone network has only a Hub-P and PE1 and the enterprise has only two branches. Spoke-PE2 and Spoke-PE3 in branches dynamically obtain their public addresses. (Configurations related to dynamic address allocation is omitted in this example and public addresses are manually specified.) Because the Internet cannot provide the MPLS function for the enterprise, the production networks and office networks in branches cannot communicate with those in the headquarters.

The enterprise wants to expand the IP/MPLS backbone network, deploy BGP/MPLS IP VPN in the headquarters and branches, and use LDP LSP to transmit data from **vpn1** and **vpn2** to implement secure interconnection between the headquarters and branches and between branches. VPN data between branches needs to be forwarded by the headquarters and encrypted using IPsec so that the headquarters can monitor and protect data.

Figure 8-65 Networking diagram for configuring L3VPN with LDP signals carried by DSVPN and protected by IPsec





Configuration Roadmap

To expand the IP/MPLS backbone network and deploy BGP/MPLS IP VPN for an enterprise, you need to add the Spoke-PE devices in the branches to the IP/MPLS backbone network in the headquarters. MPLS LDP packets between the headquarters and branches need to be transmitted over GRE tunnels because the Internet cannot provide the MPLS function. As there are a large number of branches and devices in the branches dynamically obtain their public addresses, DSVPN is used to establish GRE tunnels between the headquarters and branches. In addition, IPSec is required to encrypt and protect VPN data transmitted over the Internet. As a result, L3VPN with LDP signals carried by DSVPN and protected by IPSec can meet the requirements of the enterprise.

The configuration roadmap for L3VPN with LDP signals carried by DSVPN and protected by IPSec is as follows:

1. Configure branch devices to save only summarized routes to the headquarters, configure OSPF on Hub-P and Spoke-PEs to advertise routes, and set the OSPF network type to point-to-multipoint (P2MP), so that all VPN data between branches is forwarded by the headquarters.
2. Configure IPSec on Spoke-PE2, Spoke-PE3, and Hub-P and apply IPSec profiles to tunnel interfaces to encrypt and protect VPN data between branches.
3. Enable MPLS LDP on tunnel interfaces of Spoke-PE2, Spoke-PE3, and Hub-P and set up MPLS LSP tunnels to implement LDP over mGRE.
4. Configure L3VPN on Spoke-PE2, Spoke-PE3, and PE1 to implement secure interconnection between the headquarters and branches and between branches. Because there are a large number of branches, a route reflector can be used to reduce the number of MP-IBGP connections between PEs.

NOTE

Do not configure NHRP redirection on the Hub because LDP over mGRE does not need to establish tunnels for direct communication between branches.

Procedure

- Step 1** Configure interface IP addresses and OSPF on Hub-P and PE1 to implement interconnection on the IP/MPLS backbone network.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.1 24
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 32
[PE1-LoopBack1] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

The configuration of Hub-P is similar to that of PE1, and is not mentioned here.

After the configuration is complete, an OSPF neighbor relationship can be set up between Hub-P and PE1. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that Hub-P and PE1 have learnt the routes to Loopback1 of each other.

- Step 2** Configure interface IP addresses and static routes on Hub-P, Spoke-PE2, and Spoke-PE3 to ensure that public routes are reachable.

Because Hub-P, Spoke-PE2, and Spoke-PE3 are directly connected to the Internet, IP addresses and default static routes are manually specified here.

Configure Spoke-PE2.

```
<Huawei> system-view
[Huawei] sysname Spoke-PE2
[Spoke-PE2] interface gigabitethernet 1/0/0
[Spoke-PE2-GigabitEthernet1/0/0] ip address 202.2.1.2 24
[Spoke-PE2-GigabitEthernet1/0/0] quit
[Spoke-PE2] interface loopback 1
[Spoke-PE2-LoopBack1] ip address 2.2.2.9 32
[Spoke-PE2-LoopBack1] quit
[Spoke-PE2] ip route-static 0.0.0.0 0 202.2.1.1
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

After the configuration is complete, devices can ping each other and public routes are reachable.

- Step 3** Create tunnel interfaces and configure DSVPN on Hub-P, Spoke-PE2, and Spoke-PE3.

1. Create an mGRE interface, configure an IP address, and specify a source tunnel interface.

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] ip address 172.10.1.2 24
[Spoke-PE2-Tunnel0/0/1] tunnel-protocol gre p2mp
[Spoke-PE2-Tunnel0/0/1] source gigabitethernet 1/0/0
[Spoke-PE2-Tunnel0/0/1] quit
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

2. Configure OSPF to advertise the MPLS LSR ID as DSVPN subnet information through the tunnel interface.

Configure Spoke-PE2.

```
[Spoke-PE2] ospf 1
[Spoke-PE2-ospf-1] area 0
[Spoke-PE2-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[Spoke-PE2-ospf-1-area-0.0.0.0] network 172.10.1.0 0.0.0.255
[Spoke-PE2-ospf-1-area-0.0.0.0] quit
[Spoke-PE2-ospf-1] quit
```

The configurations of Spoke-PE3 and Hub-P are similar to that of Spoke-PE2, and are not mentioned here.

3. Configure NHRP and set the OSPF network type to P2MP. Do not configure NHRP redirection on the Hub-P.

Configure Hub-P.

```
[Hub-P] interface tunnel 0/0/1
[Hub-P-Tunnel0/0/1] nhrp entry multicast dynamic
[Hub-P-Tunnel0/0/1] ospf network-type p2mp
[Hub-P-Tunnel0/0/1] ospf dr-priority 100
[Hub-P-Tunnel0/0/1] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] nhrp entry 172.10.1.1 202.1.1.2 register
[Spoke-PE2-Tunnel0/0/1] ospf network-type p2mp
[Spoke-PE2-Tunnel0/0/1] ospf dr-priority 0
[Spoke-PE2-Tunnel0/0/1] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] interface tunnel 0/0/1
[Spoke-PE3-Tunnel0/0/1] nhrp entry 172.10.1.1 202.1.1.2 register
[Spoke-PE3-Tunnel0/0/1] ospf network-type p2mp
[Spoke-PE3-Tunnel0/0/1] ospf dr-priority 0
[Spoke-PE3-Tunnel0/0/1] quit
```

After the configuration is complete, run the **display nhrp peer all** command on Hub-P to view registration information about Spoke-PE2 and Spoke-PE3.

```
[Hub] display nhrp peer all
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.10.1.2     32    202.2.1.2      172.10.1.2   dynamic  route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time   : 00:02:36
Expire time    : 01:57:24
-----
Protocol-addr  Mask  NBMA-addr      NextHop-addr  Type      Flag
-----
172.10.1.3     32    202.3.1.2      172.10.1.3   dynamic  route tunnel
-----
Tunnel interface: Tunnel0/0/1
Created time   : 00:00:04
Expire time    : 01:59:56
-----
Number of nhrp peers: 2
```

Run the **display ip routing-table** command on all devices on the IP/MPLS backbone network. You can see that all devices have learnt the routes to Loopback1 of other devices.

Step 4 Configure IPsec on Spoke-PE2, Spoke-PE3, and Hub-P.

Configure IPsec on the devices and bind IPsec profiles to the tunnel interfaces.

Configure Hub-P.

```
[Hub-P] ipsec proposal pro1
[Hub-P-ipsec-proposal-pro1] transform ah-esp
[Hub-P-ipsec-proposal-pro1] ah authentication-algorithm sha2-512
[Hub-P-ipsec-proposal-pro1] esp authentication-algorithm sha2-512
[Hub-P-ipsec-proposal-pro1] esp encryption-algorithm aes-256
[Hub-P-ipsec-proposal-pro1] quit
[Hub-P] ike proposal 1
[Hub-P-ike-proposal-1] dh group5
[Hub-P-ike-proposal-1] authentication-algorithm aes-xcbc-mac-96
[Hub-P-ike-proposal-1] prf aes-xcbc-128
[Hub-P-ike-proposal-1] quit
[Hub-P] ike peer Hub-P v2
[Hub-P-ike-peer-Hub-P] ike-proposal 1
[Hub-P-ike-peer-Hub-P] pre-shared-key cipher huawei
[Hub-P-ike-peer-Hub-P] quit
[Hub-P] ipsec profile profile1
[Hub-P-ipsec-profile-profile1] proposal pro1
[Hub-P-ipsec-profile-profile1] ike-peer Hub-P
[Hub-P-ipsec-profile-profile1] quit
[Hub-P] interface tunnel 0/0/1
[Hub-P-Tunnel0/0/1] ipsec profile profile1
[Hub-P-Tunnel0/0/1] quit
```

The configurations of Spoke-PE2 and Spoke-PE3 are similar to that of Hub-P, and are not mentioned here.

After the configuration is complete, run the **display ipsec sa** command on Spoke-PE2, Spoke-PE3, and Hub-P. You can see that security associations (SAs) have been established.

Step 5 Enable basic MPLS functions and MPLS LDP on Spoke-PE2, Spoke-PE3, Hub-P, and PE1.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
```

The configurations of Spoke-PE2, Spoke-PE3 and Hub-P are similar to that of PE1, and are not mentioned here.

Step 6 Enable MPLS LDP on the interfaces of Spoke-PE2, Spoke-PE3, Hub-P, and PE1.

Enable MPLS LDP on interfaces of Hub-P and PE1 that are directly connected to each other and enable MPLS LDP on tunnel interfaces of Spoke-PE2, Spoke-PE3 and Hub-P to establish MPLS LSP tunnels.

Configure PE1.

```
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure Hub-P.

```
[Hub-P] interface gigabitethernet 2/0/0
[Hub-P-GigabitEthernet2/0/0] mpls
[Hub-P-GigabitEthernet2/0/0] mpls ldp
[Hub-P-GigabitEthernet2/0/0] quit
```

```
[Hub-P] interface tunnel 0/0/1
[Hub-P-Tunnel0/0/1] mpls
[Hub-P-Tunnel0/0/1] mpls ldp
[Hub-P-Tunnel0/0/1] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] interface tunnel 0/0/1
[Spoke-PE2-Tunnel0/0/1] mpls
[Spoke-PE2-Tunnel0/0/1] mpls ldp
[Spoke-PE2-Tunnel0/0/1] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] interface tunnel 0/0/1
[Spoke-PE3-Tunnel0/0/1] mpls
[Spoke-PE3-Tunnel0/0/1] mpls ldp
[Spoke-PE3-Tunnel0/0/1] quit
```

After the configuration is complete, PE1, Spoke-PE2, and Spoke-PE3 can establish LDP sessions with Hub-P. Run the **display mpls ldp session** command. You can see that the MPLS LDP session status is **Operational**.

Step 7 Configure VPN instances on Spoke-PE2, Spoke-PE3, and PE1 and bind VPN instances to interfaces.

Configure PE1.

```
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] ipv4-family
[PE1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[PE1-vpn-instance-vpn1-af-ipv4] quit
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] ipv4-family
[PE1-vpn-instance-vpn2-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[PE1-vpn-instance-vpn2-af-ipv4] quit
[PE1-vpn-instance-vpn2] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[PE1-GigabitEthernet2/0/0] ip address 192.168.1.1 24
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
[PE1-GigabitEthernet3/0/0] ip address 192.168.2.1 24
[PE1-GigabitEthernet3/0/0] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] ip vpn-instance vpn1
[Spoke-PE2-vpn-instance-vpn1] ipv4-family
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] route-distinguisher 200:1
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[Spoke-PE2-vpn-instance-vpn1-af-ipv4] quit
[Spoke-PE2-vpn-instance-vpn1] quit
[Spoke-PE2] ip vpn-instance vpn2
[Spoke-PE2-vpn-instance-vpn2] ipv4-family
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] route-distinguisher 200:2
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[Spoke-PE2-vpn-instance-vpn2-af-ipv4] quit
[Spoke-PE2-vpn-instance-vpn2] quit
[Spoke-PE2] interface gigabitethernet 2/0/0
[Spoke-PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[Spoke-PE2-GigabitEthernet2/0/0] ip address 192.168.11.1 24
[Spoke-PE2-GigabitEthernet2/0/0] quit
[Spoke-PE2] interface gigabitethernet 3/0/0
[Spoke-PE2-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
```

```
[Spoke-PE2-GigabitEthernet3/0/0] ip address 192.168.12.1 24
[Spoke-PE2-GigabitEthernet3/0/0] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] ip vpn-instance vpn1
[Spoke-PE3-vpn-instance-vpn1] ipv4-family
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] route-distinguisher 300:1
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] vpn-target 111:1 both
[Spoke-PE3-vpn-instance-vpn1-af-ipv4] quit
[Spoke-PE3-vpn-instance-vpn1] quit
[Spoke-PE3] ip vpn-instance vpn2
[Spoke-PE3-vpn-instance-vpn2] ipv4-family
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] route-distinguisher 300:2
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] vpn-target 222:2 both
[Spoke-PE3-vpn-instance-vpn2-af-ipv4] quit
[Spoke-PE3-vpn-instance-vpn2] quit
[Spoke-PE3] interface gigabitethernet 2/0/0
[Spoke-PE3-GigabitEthernet2/0/0] ip binding vpn-instance vpn1
[Spoke-PE3-GigabitEthernet2/0/0] ip address 192.168.21.1 24
[Spoke-PE3-GigabitEthernet2/0/0] quit
[Spoke-PE3] interface gigabitethernet 3/0/0
[Spoke-PE3-GigabitEthernet3/0/0] ip binding vpn-instance vpn2
[Spoke-PE3-GigabitEthernet3/0/0] ip address 192.168.22.1 24
[Spoke-PE3-GigabitEthernet3/0/0] quit
```

After the configuration is complete, run the **display ip vpn-instance verbose** command on each device to view the configuration of VPN instances.

Step 8 Set up MP-IBGP peer relationships between Spoke-PE2, Spoke-PE3, and PE1.

Configure PE1 as a route reflector. Spoke-PE2 and Spoke-PE3 can set up MP-IBGP peer relationships with PE1.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] group rr1 internal
[PE1-bgp] peer rr1 connect-interface loopback 1
[PE1-bgp] peer 2.2.2.9 group rr1
[PE1-bgp] peer 3.3.3.9 group rr1
[PE1-bgp] ipv4-family vpv4
[PE1-bgp-af-vpv4] peer rr1 enable
[PE1-bgp-af-vpv4] peer 2.2.2.9 group rr1
[PE1-bgp-af-vpv4] peer 3.3.3.9 group rr1
[PE1-bgp-af-vpv4] reflector cluster-id 100
[PE1-bgp-af-vpv4] peer rr1 reflect-client
[PE1-bgp-af-vpv4] undo policy vpn-target
[PE1-bgp-af-vpv4] quit
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-vpn1] import-route direct
[PE1-bgp-vpn1] quit
[PE1-bgp] ipv4-family vpn-instance vpn2
[PE1-bgp-vpn2] import-route direct
[PE1-bgp-vpn2] quit
[PE1-bgp] quit
```

Configure Spoke-PE2.

```
[Spoke-PE2] bgp 100
[Spoke-PE2-bgp] peer 1.1.1.9 as-number 100
[Spoke-PE2-bgp] peer 1.1.1.9 connect-interface loopback 1
[Spoke-PE2-bgp] ipv4-family vpv4
[Spoke-PE2-bgp-af-vpv4] peer 1.1.1.9 enable
[Spoke-PE2-bgp-af-vpv4] quit
[Spoke-PE2-bgp] ipv4-family vpn-instance vpn1
[Spoke-PE2-bgp-vpn1] import-route direct
[Spoke-PE2-bgp-vpn1] quit
[Spoke-PE2-bgp] ipv4-family vpn-instance vpn2
```

```
[Spoke-PE2-bgp-vpn2] import-route direct
[Spoke-PE2-bgp-vpn2] quit
[Spoke-PE2-bgp] quit
```

Configure Spoke-PE3.

```
[Spoke-PE3] bgp 100
[Spoke-PE3-bgp] peer 1.1.1.9 as-number 100
[Spoke-PE3-bgp] peer 1.1.1.9 connect-interface loopback 1
[Spoke-PE3-bgp] ipv4-family vpnv4
[Spoke-PE3-bgp-af-vpnv4] peer 1.1.1.9 enable
[Spoke-PE3-bgp-af-vpnv4] quit
[Spoke-PE3-bgp] ipv4-family vpn-instance vpn1
[Spoke-PE3-bgp-vpn1] import-route direct
[Spoke-PE3-bgp-vpn1] quit
[Spoke-PE3-bgp] ipv4-family vpn-instance vpn2
[Spoke-PE3-bgp-vpn2] import-route direct
[Spoke-PE3-bgp-vpn2] quit
[Spoke-PE3-bgp] quit
```

After the configuration is complete, run the **display bgp vpnv4 all peer** command on Spoke-PE2, Spoke-PE3, and PE1. You can see that Spoke-PE2, Spoke-PE3, and PE1 have set up BGP peer relationships with PE1 and are in **Established** state.

The display on PE1 is used as an example:

```
[PE1] display bgp vpnv4 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  2.2.2.9       4          100     5        12      0 00:02:00
Established
  3.3.3.9       4          100     5        11      0 00:01:02
Established
```

Step 9 Verify the configuration.

After the configuration is complete, Spoke-PE2, Spoke-PE3, and PE1 can learn the routes to **vpn1** and **vpn2** of each other.

The display on PE1 is used as an example:

```
[PE1] display ip routing-table vpn-instance vpn1
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpn1
  Destinations : 6          Routes : 6

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
  192.168.1.0/24    Direct  0    0       D    192.168.1.1
GigabitEthernet2/0/0
  192.168.1.1/32    Direct  0    0       D    127.0.0.1
GigabitEthernet2/0/0
  192.168.1.255/32  Direct  0    0       D    127.0.0.1
GigabitEthernet2/0/0
  192.168.11.0/24   IBGP    255  0       RD   2.2.2.9
GigabitEthernet1/0/0
  192.168.21.0/24  IBGP    255  0       RD   3.3.3.9
GigabitEthernet1/0/0
  255.255.255.255/32 Direct  0    0       D    127.0.0.1          InLoopBack0

[PE1] display ip routing-table vpn-instance vpn2
Route Flags: R - relay,
```

```
D - download to fib
-----
Routing Tables: vpn2
      Destinations : 6          Routes : 6

Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
 192.168.2.0/24    Direct  0    0        D   192.168.2.1
GigabitEthernet3/0/0
 192.168.2.1/32    Direct  0    0        D   127.0.0.1
GigabitEthernet3/0/0
 192.168.2.255/32  Direct  0    0        D   127.0.0.1
GigabitEthernet3/0/0
 192.168.12.0/24   IBGP    255  0        RD   2.2.2.9
GigabitEthernet1/0/0
 192.168.22.0/24   IBGP    255  0        RD   3.3.3.9
GigabitEthernet1/0/0
255.255.255.255/32 Direct  0    0        D   127.0.0.1          InLoopBack0
```

Devices in the same VPN can successfully ping each other, whereas devices in different VPNs cannot.

The display on Spoke-PE2 is used as an example:

```
[Spoke-PE2] ping -vpn-instance vpn1 -a 192.168.11.1 192.168.1.1
PING 192.168.1.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.1.1: bytes=56 Sequence=1 ttl=254 time=10 ms
  Reply from 192.168.1.1: bytes=56 Sequence=2 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=3 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=4 ttl=254 time=1 ms
  Reply from 192.168.1.1: bytes=56 Sequence=5 ttl=254 time=1 ms

--- 192.168.1.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/2/10 ms

[Spoke-PE2] ping -vpn-instance vpn2 -a 192.168.12.1 192.168.2.1
PING 192.168.2.1: 56 data bytes, press CTRL_C to break
  Reply from 192.168.2.1: bytes=56 Sequence=1 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=2 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=3 ttl=254 time=10 ms
  Reply from 192.168.2.1: bytes=56 Sequence=4 ttl=254 time=1 ms
  Reply from 192.168.2.1: bytes=56 Sequence=5 ttl=254 time=1 ms

--- 192.168.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/2/10 ms
```

----End

Configuration Files

NOTE

This example does not provide configuration files of devices on the Internet.

- PE1 configuration file

```
#
 sysname PE1
#
 ip vpn-instance
 vpn1
  ipv4-
 family
```

```
route-distinguisher
100:1
  vpn-target 111:1 export-
  extcommunity
  vpn-target 111:1 import-
  extcommunity
#

ip vpn-instance
vpn2
  ipv4-
  family
  route-distinguisher
  100:2
  vpn-target 222:2 export-
  extcommunity
  vpn-target 222:2 import-
  extcommunity
#

mpls lsr-id
1.1.1.9
mpls
#

mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 172.1.1.1
  255.255.255.0

mpls

mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance
  vpn1
  ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance
  vpn2
  ip address 192.168.2.1 255.255.255.0
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#

bgp
100
  group rr1
  internal
  peer rr1 connect-interface
  LoopBack1
  peer 2.2.2.9 as-number
  100
  peer 2.2.2.9 group
  rr1
  peer 3.3.3.9 as-number
  100
  peer 3.3.3.9 group
  rr1
#

  ipv4-family
  unicast
  undo
```

```
synchronization
 peer rr1
enable
 peer 2.2.2.9
enable
 peer 2.2.2.9 group
rr1
 peer 3.3.3.9
enable
 peer 3.3.3.9 group
rr1

#

 ipv4-family
vpn4
 reflector cluster-id
100
 undo policy vpn-
target
 peer rr1
enable
 peer rr1 reflect-
client
 peer 2.2.2.9
enable
 peer 2.2.2.9 group
rr1
 peer 3.3.3.9
enable
 peer 3.3.3.9 group
rr1

#

 ipv4-family vpn-instance
vpn1
 import-route
direct

#

 ipv4-family vpn-instance
vpn2
 import-route
direct
#
ospf 1
 area 0.0.0.0
 network 1.1.1.9
0.0.0.0
 network 172.1.1.0 0.0.0.255
#
return
```

● Hub-P configuration file

```
#
sysname Hub-P
#
mpls lsr-id
4.4.4.9
mpls

#

mpls
ldp
#
ipsec proposal
pro1
```

```
transform ah-
esp
 ah authentication-algorithm
 sha2-512
  esp authentication-algorithm
 sha2-512
  esp encryption-algorithm aes-256
#
ike proposal
1
 dh
group5
 authentication-algorithm aes-xcbc-
 mac-96
 prf aes-
 xcbc-128
#
ike peer Hub-P v2
 pre-shared-key cipher %^%#JvZxR2g8c;a9~FPN~n'$7`DEV&=G(=Et02P/%\*!%^%#
 ike-proposal
1
#

ipsec profile
profile1
 ike-peer Hub-
 P
 proposal prol
#
interface GigabitEthernet1/0/0
 ip address 202.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 172.1.1.2
255.255.255.0

mpls

 mpls ldp
#
interface
LoopBack1
 ip address 4.4.4.9
255.255.255.255
#

interface
Tunnel0/0/1
 ip address 172.10.1.1
255.255.255.0
 tunnel-protocol gre
 p2mp
 source
GigabitEthernet1/0/0
 ospf network-type
 p2mp
 ospf dr-priority
100
 ipsec profile profile1

mpls

 mpls
 ldp
 nhrp entry multicast
dynamic
#

ospf
```

```
1
 area
 0.0.0.0
  network 4.4.4.9
 0.0.0.0
  network 172.1.1.0
 0.0.0.255
  network 172.10.1.0
 0.0.0.255
#
ip route-static 0.0.0.0 0.0.0.0 202.1.1.1
#
return
```

● Spoke-PE2 configuration file

```
#
 sysname Spoke-PE2
#
ip vpn-instance
vpn1
 ipv4-
 family
  route-distinguisher
 200:1
  vpn-target 111:1 export-
  extcommunity
  vpn-target 111:1 import-
  extcommunity
#
ip vpn-instance
vpn2
 ipv4-
 family
  route-distinguisher
 200:2
  vpn-target 222:2 export-
  extcommunity
  vpn-target 222:2 import-
  extcommunity
#
mpls lsr-id
2.2.2.9
mpls
#
mpls
 ldp
#
 ipsec proposal
 prol
  transform ah-
  esp
  ah authentication-algorithm
  sha2-512
  esp authentication-algorithm
  sha2-512
  esp encryption-algorithm
  aes-256
#
ike proposal
1
 dh
 group5
 authentication-algorithm aes-xcbc-
 mac-96
```

```
prf aes-
xcbc-128
#

ike peer Spoke-PE2 v2
pre-shared-key cipher %%#K{JG:rWVHPMnf;5\|,GW(Luq'qi8BT4nOj%5W5=)%%#
ike-proposal 1
#

ipsec profile
profile1
ike-peer Spoke-
PE2
proposal
prol
#

interface GigabitEthernet1/0/0
ip address 202.2.1.2 255.255.255.0
#

interface GigabitEthernet2/0/0
ip binding vpn-instance
vpn1
ip address 192.168.11.1 255.255.255.0
#

interface GigabitEthernet3/0/0
ip binding vpn-instance
vpn2
ip address 192.168.12.1 255.255.255.0
#

interface
LoopBack1
ip address 2.2.2.9
255.255.255.255
#

interface
Tunnel0/0/1
ip address 172.10.1.2
255.255.255.0
tunnel-protocol gre
p2mp
source
GigabitEthernet1/0/0
ospf network-type
p2mp
ospf dr-priority
0
ipsec profile profile1

mpls

mpls
ldp
nhrp entry 172.10.1.1 202.1.1.2
register
#

bgp
100
peer 1.1.1.9 as-number
100
peer 1.1.1.9 connect-interface
LoopBack1

#

ipv4-family
unicast
undo
synchronization
```

```
peer 1.1.1.9
enable

#

ipv4-family
vpnv4
policy vpn-
target
peer 1.1.1.9
enable

#

ipv4-family vpn-instance
vpn1
import-route
direct

#

ipv4-family vpn-instance
vpn2
import-route
direct
#

ospf
1
area
0.0.0.0
network 2.2.2.9
0.0.0.0
network 172.10.1.0
0.0.0.255
#

ip route-static 0.0.0.0 0.0.0.0 202.2.1.1
#
return
```

● Spoke-PE3 configuration file

```
#
sysname Spoke-PE3
#
ip vpn-instance
vpn1
ipv4-
family
route-distinguisher
300:1
vpn-target 111:1 export-
extcommunity
vpn-target 111:1 import-
extcommunity
#

ip vpn-instance
vpn2
ipv4-
family
route-distinguisher
300:2
vpn-target 222:2 export-
extcommunity
vpn-target 222:2 import-
extcommunity
#

mpls lsr-id
```

```
3.3.3.9
mpls
#
mpls
ldp
#
ipsec proposal
pro1
  transform ah-
  esp
    ah authentication-algorithm
    sha2-512
    esp authentication-algorithm
    sha2-512
    esp encryption-algorithm
    aes-256
  #
ike proposal
1
  dh
  group5
    authentication-algorithm aes-xcbc-
    mac-96
    prf aes-
    xcbc-128
  #
ike peer Spoke-PE3 v2
  pre-shared-key cipher %%#IRFGEiFPJ1$a'Qy,L*XQL_+*Grq-=yMb}ULZdS6%^%#
  ike-proposal 1
#
ipsec profile
profile1
  ike-peer Spoke-
  PE3
  proposal
  pro1
#
interface GigabitEthernet1/0/0
  ip address 202.3.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance
  vpn1
  ip address 192.168.21.1 255.255.255.0
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance
  vpn2
  ip address 192.168.22.1 255.255.255.0
#
interface
  LoopBack1
  ip address 3.3.3.9
  255.255.255.255
#
interface
  Tunnel0/0/1
  ip address 172.10.1.3
  255.255.255.0
  tunnel-protocol gre
  p2mp
  source
  GigabitEthernet1/0/0
  ospf network-type
```

```
p2mp
  ospf dr-priority
  0
  ipsec profile profile1

mpls

  mpls
  ldp
  nhrp entry 172.10.1.1 202.1.1.2
  register
  #

  bgp
  100
  peer 1.1.1.9 as-number
  100
  peer 1.1.1.9 connect-interface
  LoopBack1
  #

  ipv4-family
  unicast
  undo
  synchronization
  peer 1.1.1.9
  enable
  #

  ipv4-family
  vpnv4
  policy vpn-
  target
  peer 1.1.1.9
  enable
  #

  ipv4-family vpn-instance
  vpn1
  import-route
  direct
  #

  ipv4-family vpn-instance
  vpn2
  import-route
  direct
  #

  ospf
  1
  area
  0.0.0.0
  network 3.3.3.9
  0.0.0.0
  network 172.10.1.0
  0.0.0.255
  #

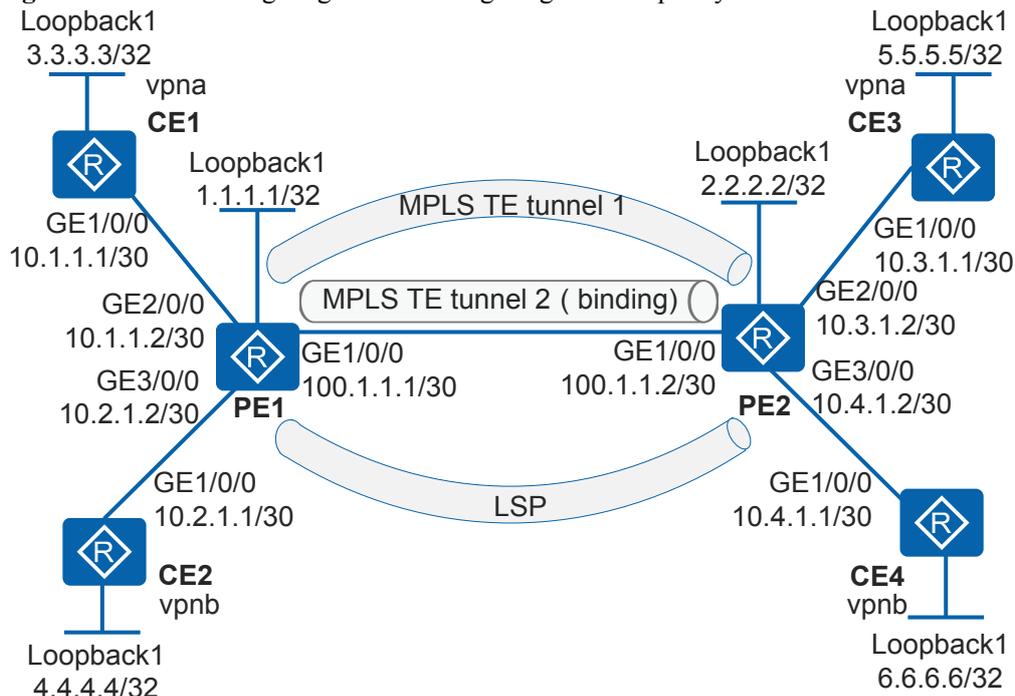
  ip route-static 0.0.0.0 0.0.0.0 202.3.1.1
  #
return
```

8.9.25 Example for Configuring a Tunnel Policy for an L3VPN

Networking Requirements

As shown in [Figure 8-66](#), CE1 and CE3 belong to vpna, and CE2 and CE4 belong to vpnb. Two MPLS TE tunnels and one LSP are set up between PE1 and PE2. To use the tunnels more efficiently, vpnb uses multiple tunnels to share the loads and prefers the TE tunnels for load balancing.

Figure 8-66 Networking diagram for configuring a tunnel policy for an L3VPN



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol so that PEs can communicate with each other.
2. Configure basic MPLS capabilities on the routers on the backbone network and set up an LSP and two MPLS TE tunnels between the PEs.
3. Configure VPN instances on PEs and connect CEs to the PEs.
4. Configure tunnel policies and apply the policies to different VPN instances.
5. Configure MP-IBGP to exchange VPN routing information.

Procedure

Step 1 Configure an IGP on the MPLS backbone network so that PEs can communicate.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
```

```
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.1 32
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet1/0/0
[PE1-GigabitEthernet1/0/0] ip address 100.1.1.1 30
[PE1-GigabitEthernet1/0/0] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 100.1.1.0 0.0.0.3
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Configure PE2.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface loopback 1
[PE2-LoopBack1] ip address 2.2.2.2 32
[PE2-LoopBack1] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip address 100.1.1.2 30
[PE2-GigabitEthernet1/0/0] quit
[PE2] ospf 1
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 100.1.1.0 0.0.0.3
[PE2-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on PEs, and you can view that PEs have learned the routes to Loopback1 interfaces from each other.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table
Route Flags: R - relay, D - download to forwarding
-----
Routing Tables: _public_
      Destinations : 9          Routes : 9
Destination/Mask    Proto Pre  Cost    Flags NextHop         Interface
 1.1.1.1/32         Direct 0     0           D 127.0.0.1         LoopBack1
 2.2.2.2/32         OSPF   10    1           D 100.1.1.2
GigabitEthernet1/0/0
 100.1.1.0/30       Direct 0     0           D 100.1.1.1
GigabitEthernet1/0/0
 100.1.1.1/32       Direct 0     0           D 127.0.0.1
GigabitEthernet1/0/0
 100.1.1.3/32       Direct 0     0           D 127.0.0.1
GigabitEthernet1/0/0
 127.0.0.0/8        Direct 0     0           D 127.0.0.1         InLoopBack0
 127.0.0.1/32       Direct 0     0           D 127.0.0.1         InLoopBack0
127.255.255.255/32 Direct 0     0           D 127.0.0.1         InLoopBack0
255.255.255.255/32 Direct 0     0           D 127.0.0.1         InLoopBack0
```

Step 2 Configure basic MPLS capabilities on the MPLS backbone to set up an LDP LSP between PEs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 2.2.2.2
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, an LDP LSP is set up between PE1 and PE2. Run the **display tunnel-info all** command, and you can find the LSP destined for the address 2.2.2.2. Run the **display mpls ldp lsp** command, and you can view LSP information.

The information displayed on PE1 is used as an example.

```
[PE1] display tunnel-info all
* -> Allocated VC Token
Tunnel ID          Type          Destination      Token
-----
0x15                lsp           2.2.2.2          21
0x16                lsp           2.2.2.2          22
[PE1] display mpls ldp lsp
LDP LSP Information
-----
DestAddress/Mask   In/OutLabel   UpstreamPeer    NextHop          OutInterface
-----
1.1.1.1/32         3/NULL        2.2.2.2          127.0.0.1        InLoop0
*1.1.1.1/32        Liberal/16     -                DS/2.2.2.2
2.2.2.2/32         NULL/3        -                100.1.1.2        GE1/0/0
2.2.2.2/32         16/3          2.2.2.2          100.1.1.2        GE1/0/0
-----
TOTAL: 3 Normal LSP(s) Found.
TOTAL: 1 Liberal LSP(s) Found.
TOTAL: 0 Frr LSP(s) Found.
A '*' before an LSP means the LSP is not established
A '*' before a Label means the USCB or DSCB is stale
A '*' before a UpstreamPeer means the session is stale
A '*' before a DS means the session is stale
A '*' before a NextHop means the LSP is FRR LSP
```

Step 3 Set up MPLS TE tunnels between PEs.

Configure PE1.

```
[PE1] mpls
[PE1-mpls] mpls te
[PE1-mpls] mpls rsvp-te
[PE1-mpls] mpls te cspf
[PE1-mpls] quit
[PE1] interface gigabitethernet1/0/0
[PE1-GigabitEthernet1/0/0] mpls te
[PE1-GigabitEthernet1/0/0] mpls rsvp-te
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls
[PE2-mpls] mpls te
[PE2-mpls] mpls rsvp-te
[PE2-mpls] mpls te cspf
[PE2-mpls] quit
[PE2] interface gigabitethernet1/0/0
[PE2-GigabitEthernet1/0/0] mpls te
[PE2-GigabitEthernet1/0/0] mpls rsvp-te
[PE2-GigabitEthernet1/0/0] quit
```

Enable OSPF on the devices along the TE tunnels to transmit TE attributes.

Configure PE1.

```
[PE1] ospf 1
[PE1-ospf-1] opaque-capability enable
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] mpls-te enable
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Configure PE2.

```
[PE2] ospf 1
[PE2-ospf-1] opaque-capability enable
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] mpls-te enable
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

Configure an MPLS TE tunnel.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE1-Tunnel0/0/1] tunnel-protocol mpls te
[PE1-Tunnel0/0/1] destination 2.2.2.2
[PE1-Tunnel0/0/1] mpls te tunnel-id 11
[PE1-Tunnel0/0/1] mpls te commit
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE2-Tunnel0/0/1] tunnel-protocol mpls te
[PE2-Tunnel0/0/1] destination 1.1.1.1
[PE2-Tunnel0/0/1] mpls te tunnel-id 11
[PE2-Tunnel0/0/1] mpls te commit
[PE2-Tunnel0/0/1] quit
```

Configure an MPLS TE tunnel and bind the tunnel to the VPN.

Configure PE1.

```
[PE1] interface tunnel 0/0/2
[PE1-Tunnel0/0/2] ip address unnumbered interface loopback 1
[PE1-Tunnel0/0/2] tunnel-protocol mpls te
[PE1-Tunnel0/0/2] destination 2.2.2.2
[PE1-Tunnel0/0/2] mpls te tunnel-id 22
[PE1-Tunnel0/0/2] mpls te reserved-for-binding
[PE1-Tunnel0/0/2] mpls te commit
[PE1-Tunnel0/0/2] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/2
[PE2-Tunnel0/0/2] ip address unnumbered interface loopback 1
[PE2-Tunnel0/0/2] tunnel-protocol mpls te
[PE2-Tunnel0/0/2] destination 1.1.1.1
[PE2-Tunnel0/0/2] mpls te tunnel-id 22
[PE2-Tunnel0/0/2] mpls te reserved-for-binding
[PE2-Tunnel0/0/2] mpls te commit
[PE2-Tunnel0/0/2] quit
```

After the configuration is complete, run the **display mpls te tunnel-interface** command on PEs, and you can view that Tunnel0/0/1 and Tunnel0/0/2 are both Up. The information displayed on PE1 is used as an example.

```
[PE1] display mpls te tunnel-interface
-----
                        Tunnel0/0/1
-----
Tunnel State Desc   : UP
Active LSP          : Primary LSP
Session ID          : 11
Ingress LSR ID     : 1.1.1.1           Egress LSR ID: 2.2.2.2
Admin State         : UP               Oper State   : UP
Primary LSP State   : UP
Main LSP State      : READY           LSP ID      : 1
-----
                        Tunnel0/0/2
-----
Tunnel State Desc   : UP
Active LSP          : Primary LSP
Session ID          : 22
Ingress LSR ID     : 1.1.1.1           Egress LSR ID: 2.2.2.2
Admin State         : UP               Oper State   : UP
Primary LSP State   : UP
Main LSP State      : READY           LSP ID      : 2
```

Step 4 Configure VPN instances on PEs and bind the instances to the interfaces connected to CEs.

Configure PE1.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] route-distinguisher 100:2
[PE1-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface gigabitethernet2/0/0
[PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.2 30
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] ip binding vpn-instance vpb
[PE1-GigabitEthernet3/0/0] ip address 10.2.1.2 30
[PE1-GigabitEthernet3/0/0] quit
```

Configure PE2.

```
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 100:3
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv4-family
[PE2-vpn-instance-vpb-af-ipv4] route-distinguisher 100:4
[PE2-vpn-instance-vpb-af-ipv4] vpn-target 222:2 both
[PE2-vpn-instance-vpb-af-ipv4] quit
[PE2-vpn-instance-vpb] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet2/0/0] ip address 10.3.1.2 30
[PE2-GigabitEthernet2/0/0] quit
[PE2] interface gigabitethernet 3/0/0
[PE2-GigabitEthernet3/0/0] ip binding vpn-instance vpb
[PE2-GigabitEthernet3/0/0] ip address 10.4.1.2 30
[PE2-GigabitEthernet3/0/0] quit
```

Assign IP addresses to the interfaces on the CEs according to [Figure 8-66](#). The configuration procedure is not provided here.

After the configuration is complete, run the **display ip vpn-instance verbose** command on PEs, and you can view configuration of the VPN instances.

 **NOTE**

If a PE has multiple interfaces bound to the same VPN, when you run the ping command to ping the CE connected to the remote PE, specify the source IP address; that is, specify **-a source-ip-address** in the **ping -a source-ip-address -vpn-instance vpn-instance-name destination-address** command. Otherwise, the ping fails.

Step 5 Configure and apply a tunnel policy on PEs.

Configure the tunnel policy for binding primary tunnel and apply the tunnel policy to vpna.

Configure PE1.

```
[PE1] tunnel-policy policy1
[PE1-tunnel-policy-policy1] tunnel binding destination 2.2.2.2 te tunnel 0/0/2
[PE1-tunnel-policy-policy1] quit
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] tnl-policy policy1
[PE1-vpn-instance-vpna-af-ipv4] quit
[PE1-vpn-instance-vpna] quit
```

Configure PE2.

```
[PE2] tunnel-policy policy1
[PE2-tunnel-policy-policy1] tunnel binding destination 1.1.1.1 te tunnel 0/0/2
[PE2-tunnel-policy-policy1] quit
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] tnl-policy policy1
[PE2-vpn-instance-vpna-af-ipv4] quit
[PE2-vpn-instance-vpna] quit
```

Configure a tunnel type prioritizing policy and apply the policy to vpb.

Configure PE1.

```
[PE1] tunnel-policy policy2
[PE1-tunnel-policy-policy2] tunnel select-seq cr-lsp lsp load-balance-number 2
[PE1-tunnel-policy-policy2] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv4-family
[PE1-vpn-instance-vpb-af-ipv4] tnl-policy policy2
[PE1-vpn-instance-vpb-af-ipv4] quit
[PE1-vpn-instance-vpb] quit
```

Configure PE2.

```
[PE2] tunnel-policy policy2
[PE2-tunnel-policy-policy2] tunnel select-seq cr-lsp lsp load-balance-number 2
[PE2-tunnel-policy-policy2] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv4-family
[PE2-vpn-instance-vpb-af-ipv4] tnl-policy policy2
[PE2-vpn-instance-vpb-af-ipv4] quit
[PE2-vpn-instance-vpb] quit
```

Step 6 Set up an MP-IBGP peer relationship between the PEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.2 as-number 100
```

```
[PE1-bgp] peer 2.2.2.2 connect-interface loopback 1
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 2.2.2.2 enable
[PE1-bgp-af-vpnv4] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] peer 1.1.1.1 as-number 100
[PE2-bgp] peer 1.1.1.1 connect-interface loopback 1
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 1.1.1.1 enable
[PE2-bgp-af-vpnv4] quit
```

After the configuration is complete, run the **display bgp peer** or **display bgp vpnv4 all peer** command on the PEs. The command output shows that a BGP peer relationship is set up between PEs and the BGP peer relationship is in Established state.

Step 7 Set up EBGP peer relationships between PEs and CEs.

Configure PE1.

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-af-vpna] peer 10.1.1.1 as-number 65410
[PE1-bgp-af-vpna] quit
[PE1-bgp] ipv4-family vpn-instance vpnb
[PE1-bgp-af-vpnb] peer 10.2.1.1 as-number 65410
[PE1-bgp-af-vpnb] quit
[PE1-bgp] quit
```

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 10.1.1.2 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] quit
```

Configure CE2.

```
[CE2] bgp 65410
[CE2-bgp] peer 10.2.1.2 as-number 100
[CE2-bgp] import-route direct
[CE2-bgp] quit
```

Configure PE2.

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-af-vpna] peer 10.3.1.1 as-number 65420
[PE2-bgp-af-vpna] quit
[PE2-bgp] ipv4-family vpn-instance vpnb
[PE2-bgp-af-vpnb] peer 10.4.1.1 as-number 65420
[PE2-bgp-af-vpnb] quit
[PE2-bgp] quit
```

Configure CE3.

```
[CE3] bgp 65420
[CE3-bgp] peer 10.3.1.2 as-number 100
[CE3-bgp] import-route direct
[CE3-bgp] quit
```

Configure CE4.

```
[CE4] bgp 65420
[CE4-bgp] peer 10.4.1.2 as-number 100
[CE4-bgp] import-route direct
[CE4-bgp] quit
```

Step 8 Verify the configuration.

Run the **display bgp routing-table** command on CEs, and you can find the routes to the remote CEs.

The information displayed on CE1 is used as an example.

```
[CE1] display bgp routing-table

BGP Local router ID is 3.3.3.3
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 5
  Network          NextHop          MED           LocPrf        PrefVal Path/Ogn
* > 3.3.3.3/32      0.0.0.0          0              0              0      ?
* > 5.5.5.5/32      10.1.1.2         0              0              0      100 65420?
* > 10.4.1.0/24     0.0.0.0          0              0              0      ?
              10.4.1.1         0              0              0      100?
* > 10.1.1.2/32    0.0.0.0          0              0              0      ?
* > 10.3.1.0/30    10.1.1.2         0              0              0      100?
* > 127.0.0.0      0.0.0.0          0              0              0      ?
* > 127.0.0.1/32   0.0.0.0          0              0              0      ?
```

Run the **display ip routing-table vpn-instance verbose** command on PEs, and you can find the tunnels used by the VPN routes.

The information displayed on PE1 is used as an example.

```
[PE1] display ip routing-table vpn-instance vpna 5.5.5.5 verbose
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: vpna
Summary Count : 1

Destination: 5.5.5.5/32
  Protocol: IBGP          Process ID: 0
  Preference: 255        Cost: 0
  NextHop: 2.2.2.2       Neighbour: 2.2.2.2
  State: Active Adv Relied Age: 00h00m08s
  Tag: 0                 Priority: low
  Label: 0x13            QoSInfo: 0x0
  IndirectID: 0xb9
RelayNextHop: 0.0.0.0    Interface: Tunnel0/0/2
  TunnelID: 0x3d         Flags: RD
[PE1] display ip routing-table vpn-instance vpnb 6.6.6.6 verbose
Route Flags: R - relay, D - download for forwarding
-----
Routing Table : vpnb
Summary Count : 1

Destination: 6.6.6.6/32
  Protocol: IBGP          Process ID: 0
  Preference: 255        Cost: 0
  NextHop: 2.2.2.2       Neighbour: 2.2.2.2
  State: Active Adv Relied Age: 00h04m37s
  Tag: 0                 Priority: low
  Label: 0x15            QoSInfo: 0x0
  IndirectID: 0xb8
RelayNextHop: 0.0.0.0    Interface: Tunnel0/0/1
  TunnelID: 0x3b         Flags: RD
RelayNextHop: 0.0.0.0    Interface: LDP LSP
  TunnelID: 0x1c         Flags: RD
```

CEs in the same VPN can ping each other, whereas CEs in different VPNs cannot.

----End

Configuration Files

- PE1 configuration file

```
#
sysname PE1
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:1
    tnl-policy policy1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 100:2
    tnl-policy policy2
    vpn-target 222:2 export-extcommunity
    vpn-target 222:2 import-extcommunity
#
mpls lsr-id 1.1.1.1
mpls
  mpls te
  mpls rsvp-te
  mpls te cspf
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 100.1.1.1 255.255.255.252
  mpls
  mpls te
  mpls rsvp-te
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpna
  ip address 10.1.1.2 255.255.255.252
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance vpnb
  ip address 10.2.1.2 255.255.255.252
#
interface LoopBack1
  ip address 1.1.1.1 255.255.255.255
#
interface Tunnel0/0/1
  ip address unnumbered interface LoopBack1
  tunnel-protocol mpls te
  destination 2.2.2.2
  mpls te tunnel-id 11
  mpls te commit
#
interface Tunnel0/0/2
  ip address unnumbered interface LoopBack1
  tunnel-protocol mpls te
  destination 2.2.2.2
  mpls te tunnel-id 22
  mpls te reserved-for-binding
  mpls te commit
#
bgp 100
  peer 2.2.2.2 as-number 100
  peer 2.2.2.2 connect-interface LoopBack1
```

```
#
ipv4-family unicast
  undo synchronization
  peer 2.2.2.2 enable
#
ipv4-family vpnv4
  policy vpn-target
  peer 2.2.2.2 enable
#
ipv4-family vpn-instance vpna
  peer 10.1.1.1 as-number 65410
#
ipv4-family vpn-instance vpnb
  peer 10.2.1.1 as-number 65410
#
ospf 1
  opaque-capability enable
  area 0.0.0.0
  mpls-te enable
  network 100.1.1.0 0.0.0.3
  network 1.1.1.1 0.0.0.0
#
tunnel-policy policy1
  tunnel binding destination 2.2.2.2 te Tunnel0/0/2
#
tunnel-policy policy2
  tunnel select-seq cr-lsp lsp load-balance-number 2
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 100:3
    tnl-policy policy1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
  ipv4-family
    route-distinguisher 100:4
    tnl-policy policy2
    vpn-target 222:2 export-extcommunity
    vpn-target 222:2 import-extcommunity
#
mpls lsr-id 2.2.2.2
mpls
  mpls te
  mpls rsvp-te
  mpls te cspf
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 100.1.1.2 255.255.255.252
  mpls
    mpls te
    mpls rsvp-te
    mpls ldp
#
interface GigabitEthernet2/0/0
  ip binding vpn-instance vpna
  ip address 10.3.1.2 255.255.255.252
#
interface GigabitEthernet3/0/0
  ip binding vpn-instance vpnb
  ip address 10.4.1.2 255.255.255.252
```

```
#
interface LoopBack1
 ip address 2.2.2.2 255.255.255.255
#
interface Tunnel0/0/1
 ip address unnumbered interface LoopBack1
 tunnel-protocol mpls te
 destination 1.1.1.1
 mpls te tunnel-id 11
 mpls te commit
#
interface Tunnel0/0/2
 ip address unnumbered interface LoopBack1
 tunnel-protocol mpls te
 destination 1.1.1.1
 mpls te tunnel-id 22
 mpls te reserved-for-binding
 mpls te commit
#
bgp 100
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack1
#
ipv4-family unicast
 undo synchronization
 peer 1.1.1.1 enable
#
ipv4-family vpnv4
 policy vpn-target
 peer 1.1.1.1 enable
#
ipv4-family vpn-instance vpna
 peer 10.3.1.1 as-number 65420
#
ipv4-family vpn-instance vpnb
 peer 10.4.1.1 as-number 65420
#
ospf 1
 opaque-capability enable
 area 0.0.0.0
 mpls-te enable
 network 100.1.1.0 0.0.0.3
 network 2.2.2.2 0.0.0.0
#
tunnel-policy policy1
 tunnel binding destination 1.1.1.1 te Tunnel0/0/2
#
tunnel-policy policy2
 tunnel select-seq cr-lsp lsp load-balance-number 2
#
return
```

● CE1 configuration file

```
#
 sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.252
#
interface LoopBack1
 ip address 3.3.3.3 255.255.255.255
#
bgp 65410
 peer 10.1.1.2 as-number 100
#
ipv4-family unicast
 undo synchronization
 import-route direct
 peer 10.1.1.2 enable
```

```
#  
return
```

- CE2 configuration file

```
#  
sysname CE2  
#  
interface GigabitEthernet1/0/0  
ip address 10.2.1.1 255.255.255.252  
#  
interface LoopBack1  
ip address 4.4.4.4 255.255.255.255  
#  
bgp 65410  
peer 10.2.1.2 as-number 100  
#  
ipv4-family unicast  
undo synchronization  
import-route direct  
peer 10.2.1.2 enable  
#  
return
```

- CE3 configuration file

```
#  
sysname CE3  
#  
interface GigabitEthernet1/0/0  
ip address 10.3.1.1 255.255.255.252  
#  
interface LoopBack1  
ip address 5.5.5.5 255.255.255.255  
#  
bgp 65420  
peer 10.3.1.2 as-number 100  
#  
ipv4-family unicast  
undo synchronization  
import-route direct  
peer 10.3.1.2 enable  
#  
return
```

- CE4 configuration file

```
#  
sysname CE4  
#  
interface GigabitEthernet1/0/0  
ip address 10.4.1.1 255.255.255.252  
#  
interface LoopBack1  
ip address 6.6.6.6 255.255.255.255  
#  
bgp 65420  
peer 10.4.1.2 as-number 100  
#  
ipv4-family unicast  
undo synchronization  
import-route direct  
peer 10.4.1.2 enable  
#  
return
```

8.10 FAQ About BGP/MPLS IP VPN

This section describes the FAQ about BGP/MPLS IP VPN.

8.10.1 Why Routes Cannot Be Imported When AS Numbers on the BGP/MPLS IP VPN Are the Same?

When the AS number in the Update message to be received by the EBGP-enabled device is the same as the AS number on the device, the device does not receive the Update message. This prevents routing loops. In some scenarios, the device needs to receive a Update message that carries the same AS number as the AS number on the device. In Hub and Spoke networking, when the Hub-PE and Hub-CE use EBGP, the Update message received by the Hub-CE contains the AS number of the Hub-PE. To prevent the Hub-PE from discarding such Update message, run the **peer allow-as-loop** command to set the number of times for the repeated AS number.

8.11 References for BGP/MPLS IP VPN

This section lists references for BGP/MPLS IP VPN.

The following table lists the references for BGP/MPLS IP VPN.

Docu ment	Description	Remarks
RFC1772	Application of the Border Gateway Protocol in the Internet	-
RFC 4760	Multiprotocol Extensions for BGP-4	-
RFC 4364	BGP/MPLS IP Virtual Private Networks (VPNs)	-
RFC 2764	A Framework for IP Based Virtual Private Networks	-
RFC 5492	Capabilities Advertisement with BGP-4	-
RFC 2917	A Core MPLS IP VPN Architecture	-
RFC 3107	Carrying Label Information in BGP-4	-
RFC 4026	Provider Provisioned Virtual Private Network (VPN) Terminology	-
RFC 4577	OSPF as the Provider/Customer Edge Protocol for BGP/ MPLS IP Virtual Private Networks (VPNs)	-
RFC3478	Graceful Restart Mechanism for Label Distribution Protocol	-

9 MCE IPv6 Configuration

About This Chapter

In the IPv6 network, one CE device connects to only one VPN. If multiple VPNs are deployed on a customer network, multiple CE devices are required. A multi-VPN-instance CE (MCE) device can connect to multiple VPNs. The MCE solution isolates services of different VPNs while reducing cost of network devices.

[9.1 Overview of MCE IPv6](#)

A multi-VPN-instance CE (MCE) device uses routing multi-instance to isolate services or users of IPv6.

[9.2 Licensing Requirements and Limitations for MCE IPv6](#)

[9.3 Configuring an MCE Device](#)

An MCE device can connect to multiple VPNs. The MCE solution isolates services of different VPNs while reducing cost of network devices.

[9.4 Configuration Examples for MCE IPv6](#)

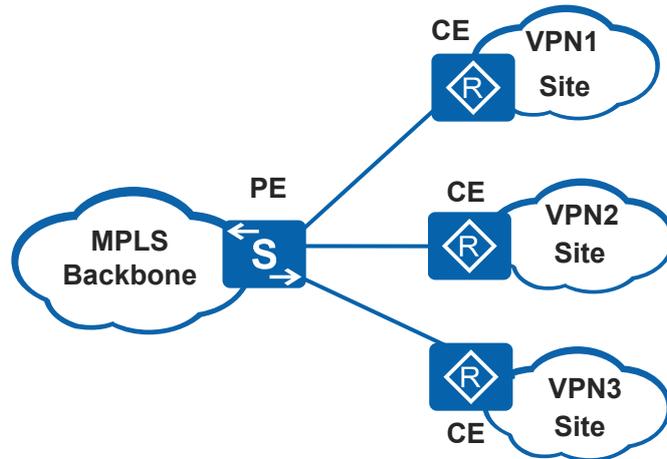
This section provides an example for configuring an MCE IPv6 device.

9.1 Overview of MCE IPv6

A multi-VPN-instance CE (MCE) device uses routing multi-instance to isolate services or users of IPv6.

BGP/MPLS IP VPN uses tunnels to transmit data of private networks on a public network. In the traditional BGP/MPLS IP VPN architecture, each VPN instance must use a CE device to connect to a PE device, as shown in [Figure 9-1](#).

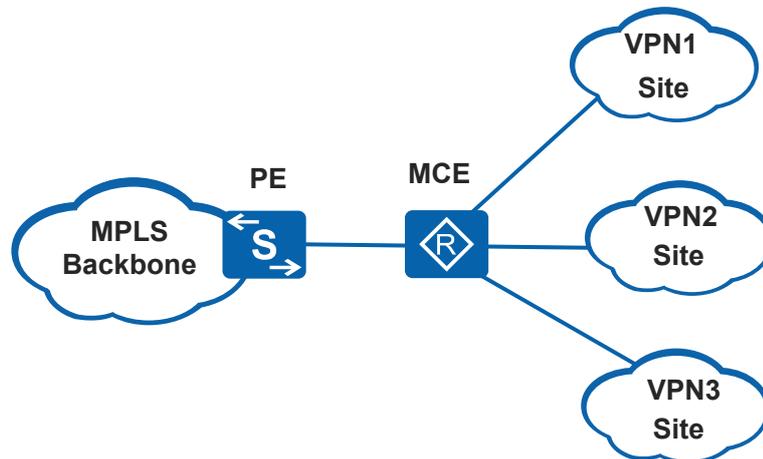
Figure 9-1 Networking without an MCE device



In many cases, a private network must be divided into multiple VPNs to realize fine-grained service management and enhance security. Services of users in different VPNs must be completely isolated. Deploying a CE device for each VPN increases the cost of device procurement and maintenance. If multiple VPNs share one CE device, data security cannot be ensured because all the VPNs use the same routing and forwarding table.

The MCE technology ensures data security between different VPNs while reducing network construction and maintenance costs. Figure 9-2 shows the MCE deployment.

Figure 9-2 Networking with an MCE device



An MCE device has some PE functions. By binding each VPN instance to a different interface, an MCE device creates and maintains an independent VRF for each VPN. This application is also called multi-VRF application. The MCE device isolates forwarding paths of different VPNs on a private network and advertises routes of each VPN to the peer PE device, ensuring that VPN packets are correctly transmitted on the public network.

9.2 Licensing Requirements and Limitations for MCE IPv6.

Involved Network Elements

None

License Requirements

MCE IPv6 is a basic feature of the device and is not under license control.

Feature Limitations

The AR120-S series do not support MCE IPv6.

9.3 Configuring an MCE Device

An MCE device can connect to multiple VPNs. The MCE solution isolates services of different VPNs while reducing cost of network devices.

Pre-configuration Tasks

Before configuring an MCE device, complete the following task:

- Configuring the link layer protocol and network layer protocol for LAN interfaces and connecting the LAN to the MCE device (reserve one interface for each service)

Configuration Procedure

The first three tasks are mandatory.

9.3.1 Configuring a VPN Instance

Context

The following configurations are performed on the MCE device.

Similar configurations must be performed on the PE devices. The PE configuration procedure and commands used vary in devices from different vendors and different product models. For detailed configuration, see the documentation of the PE devices.

Procedure

Step 1 Enable IPv6 globally

1. Run **system-view**

The system view is displayed.

2. Run **ipv6**

IPv6 is enabled globally.

Step 2 Create VPN instance

1. Run **ip vpn-instance** *vpn-instance-name*

A VPN instance is created, and its view is displayed.

NOTE

A VPN instance name is case sensitive. For example, vpn1 and VPN1 are different VPN instances.

No default VPN instance is defined on an MCE device, and you can create multiple VPN instances on the MCE device.

2. (Optional) Run **description** *description-information*

The description is configured for the VPN instance.

The description is similar to that of the host name and interface, which can be used to record information about the relationship between a VPN instance and a VPN.

3. (Optional) Run **service-id** *service-id*

A service ID is created for the VPN instance.

A service ID is unique on a device. It distinguishes a VPN service from other VPN services on the network.

4. Run **ipv6-family**

The IPv6 address family is enabled for the VPN instance, and the VPN instance IPv6 address family view is displayed.

VPN instances support both the IPv4 and IPv6 address families. Configurations in a VPN instance can be performed only after an address family is enabled for the VPN instance based on the advertised route and forwarding data type.

5. Run **route-distinguisher** *route-distinguisher*

An RD is configured for the VPN instance IPv6 address family.

A VPN instance IPv6 address family takes effect only after being configured with an RD. The RDs of different VPN instances on a PE must be different.

NOTE

- An RD can be modified or deleted only after the VPN instance is deleted or the VPN instance IPv6 address family is disabled.

Step 3 Bind the VPN instance to an interface.

1. Run **system-view**

The system view is displayed.

2. Run **interface** *interface-type interface-number*

The interface view is displayed.

3. Run **ip binding vpn-instance** *vpn-instance-name*

The VPN instance is bound to the interface.

By default, no VPN instance is bound to a interface.

 **NOTE**

When you run the **ip binding vpn-instance** command on an interface, all configurations of Layer 3 features on the interface, such as the IP address and routing protocol, are deleted. To use these features, reconfigure them.

4. Run **ipv6 enable**

IPv6 is enabled on the interface.

5. Run **ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }**

An IPv6 address is configured for the interface.

----End

9.3.2 Configure Route Exchange Between an MCE Device and VPN Sites

Context

Routing protocols that can be used between an MCE device and VPN sites are IPv6 static routing, RIPng, OSPFv3, IS-IS IPv6. Choose one of the following configurations as needed:

- [Configure IPv6 Static Routes Between an MCE Device and a Site](#)
- [Configure RIPng Between an MCE Device and a Site](#)
- [Configure OSPFv3 Between an MCE Device and a Site](#)
- [Configure IS-IS IPv6 Between an MCE Device and a Site](#)

The following configurations are performed on the MCE device. On the devices in the site, you only need to configure the corresponding routing protocol.

Configure IPv6 Static Routes Between an MCE Device and a Site

Perform the following configurations on the MCE device. You only need to configure a IPv6 static route to the MCE device in the site. The site configuration is not provided here.

 **NOTE**

For detailed configuration of static routes, see *Configuring IPv6 Static Routes in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide – IP Routing*.

Table 9-1 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-

Action	Command	Description
Configure an ipv6 static route to the site.	ipv6 route-static vpn-instance <i>vpn-instance-name</i> <i>dest-ipv6-address</i> <i>prefix-length</i> { [<i>interface-type</i> <i>interface-number</i>] <i>nexthop-ipv6-address</i> <i>nexthop-ipv6-address</i> [public] vpn-instance <i>vpn-destination-name</i> <i>nexthop-ipv6-address</i> } [preference <i>preference</i> tag tag] * [description <i>text</i>]	You must specify the next hop address on the MCE device.

Configure RIPng Between an MCE Device and a Site

Perform the following configurations on the MCE device.

NOTE

For detailed RIPng configuration, see RIPng Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 9-2 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create a RIPng process running between the MCE device and the site and enter the RIPng view.	ripng <i>process-id</i> vpn-instance <i>vpn-instance-name</i>	A RIPng process can be bound to only one VPN instance. If a RIPng process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
(Optional) Import the routes to the remote sites advertised by the PE device in to the RIPng routing table.	import-route { { ripng isis ospfv3 } [<i>process-id</i>] unr direct static } [cost <i>cost</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.
Return to system view.	quit	-
Enter the interface view.	interface <i>interface-type</i> <i>interface-number</i>	-
Enable RIPng on the interface.	ripng <i>process-id</i> enable	-

Configure OSPFv3 Between an MCE Device and a Site

Perform the following configurations on the MCE device. Configure OSPFv3 in the site. The site configuration is not provided here.

 **NOTE**

For detailed OSPFv3 configuration, see OSPFv3 Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 9-3 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an OSPFv3 process running between the MCE device and the site and enter the OSPFv3 view.	ospfv3 [<i>process-id</i>] [vpn-instance <i>vpn-instance-name</i>]	-
(Optional) Import the routes to the remote sites advertised by the PE device into the OSPFv3 routing table.	import-route { unr direct ripng <i>help-process-id</i> static isis <i>help-process-id</i> ospfv3 <i>help-process-id</i> } [cost <i>cost</i> type <i>type</i> tag <i>tag</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.
Return to system view.	quit	-
Enter the interface view.	interface <i>interface-type</i> <i>interface-number</i>	-
Enable OSPFv3 on the interface.	ospfv3 <i>process-id</i> area <i>area-id</i> [instance <i>instance-id</i>]	-

Configure IS-IS IPv6 Between an MCE Device and a Site

Perform the following configurations on the MCE device. You only need to configure IS-IS IPv6 in the site. The site configuration is not provided here.

 **NOTE**

For detailed IS-IS configuration, see IS-IS IPv6 Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - IP Routing*.

Table 9-4 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an IS-IS process running between the MCE device and the site and enter the IS-IS IPv6 view.	isis process-id vpn-instance vpn-instance-name	An IS-IS process can be bound to only one VPN instance. If an IS-IS IPv6 process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Set a network entity title (NET) for the IS-IS process.	network-entity net	A NET specifies the current IS-IS area address and the system ID of the router. A maximum of three NETs can be configured for one process on each router.
Enable IS-IS IPv6 on the process.	ipv6 enable [topology { compatible [enable-mt-spf] ipv6 standard }]	-
(Optional) Import the routes to the remote sites advertised by the PE device into the IS-IS IPv6 routing table.	Use either of the following commands: <ul style="list-style-type: none"> ● ipv6 import-route { direct unr { ospfv3 ripng isis } [process-id] } inherit-cost [tag tag route-policy route-policy-name { level-1 level-2 level-1-2 }] * ● ipv6 import-route { static direct unr { ospfv3 ripng isis } [process-id] } [cost cost tag tag route-policy route-policy-name { level-1 level-2 level-1-2 }] * 	Perform this step if another routing protocol is running between the MCE and PE devices in the VPN instance.
Return to system view.	quit	-
Enter the view of the interface to which the VPN instance is bound.	interface interface-type interface-number	-
Enable IS-IS IPv6 on the interface.	isis ipv6 enable [process-id]	-

9.3.3 Configure Route Exchange Between an MCE Device and a PE Device

Context

Routing protocols that can be used between an MCE device and a PE device are IPv6 static routing, RIPng, OSPFv3, IS-IS IPv6. Choose one of the following configurations as needed:

- [Configure IPv6 Static Routes Between an MCE Device and a PE Device](#)
- [Configure RIPng Between an MCE Device and a PE Device](#)
- [Configure OSPFv3 Between an MCE Device and a PE Device](#)
- [Configure IS-IS IPv6 Between an MCE Device and a PE Device](#)

Configure IPv6 Static Routes Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 9-5 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Configure a IPv6 static route to the PE device.	ipv6 route-static vpn-instance <i>vpn-instance-name</i> <i>dest-ipv6-address</i> <i>prefix-length</i> { [<i>interface-type</i> <i>interface-number</i>] <i>nexthop-ipv6-address</i> <i>nexthop-ipv6-address</i> [public] vpn-instance <i>vpn-destination-name</i> <i>nexthop-ipv6-address</i> } [preference <i>preference</i> tag <i>tag</i>] * [description <i>text</i>]	You must specify the next hop address on the MCE device.

Configure RIPng Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 9-6 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-

Action	Command	Description
Create a RIPng process running between the MCE and PE devices and enter the RIPng view.	ripng <i>process-id</i> vpn-instance <i>vpn-instance-name</i>	A RIPng process can be bound to only one VPN instance. If a RIPng process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
(Optional) Import VPN routes of the site into the RIP routing table.	import-route { { ripng isis ospfv3 } [<i>process-id</i>] unr direct static } [cost <i>cost</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.
Return to system view.	quit	-
Enter an interface view.	interface <i>interface-type</i> <i>interface-number</i>	-
Enable RIPng on the interface.	ripng <i>process-id</i> enable	-

Configure OSPFv3 Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 9-7 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an OSPFv3 process running between the MCE and PE devices and enter the OSPFv3 view.	ospfv3 [<i>process-id</i>] vpn-instance <i>vpn-instance-name</i>	-
(Optional) Import VPN routes of the site into the OSPF routing table.	import-route { unr direct ripng <i>help-process-id</i> static isis <i>help-process-id</i> ospfv3 <i>help-process-id</i> } [cost <i>cost</i> type <i>type</i> tag <i>tag</i> route-policy <i>route-policy-name</i>] *	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.
Return to system view.	quit	-
Enter an interface view.	interface <i>interface-type</i> <i>interface-number</i>	-

Action	Command	Description
Enable OSPFv3 on the interface which the VPN instance is bound.	ospfv3 <i>process-id</i> area <i>area-id</i> [instance <i>instance-id</i>]	-

Configure IS-IS IPv6 Between an MCE Device and a PE Device

Perform the following configurations on the MCE device.

Table 9-8 MCE configuration

Action	Command	Description
Enter the system view.	system-view	-
Create an IS-IS process running between the MCE and PE devices and enter the IS-IS view.	isis <i>process-id</i> vpn-instance <i>vpn-instance-name</i>	An IS-IS process can be bound to only one VPN instance. If an IS-IS process is not bound to any VPN instance before it is started, this process becomes a public network process and can no longer be bound to a VPN instance.
Set a network entity title (NET) for the IS-IS process.	network-entity <i>net</i>	A NET specifies the current IS-IS area address and the system ID of the router. A maximum of three NETs can be configured for one process on each router.
Enable IPv6 for the IS-IS process.	ipv6 enable [topology { compatible [enable-mt-spf] ipv6 standard }]	-
(Optional) Import VPN routes of the site into the IS-IS routing table.	Use either of the following commands: <ul style="list-style-type: none"> ● ipv6 import-route { direct unr { ospfv3 ripng isis } [<i>process-id</i>] } inherit-cost [tag <i>tag</i> route-policy <i>route-policy-name</i> { level-1 level-2 level-1-2 }] * ● ipv6 import-route { static direct unr { ospfv3 ripng isis } [<i>process-id</i>] } [cost <i>cost</i> tag <i>tag</i> route-policy <i>route-policy-name</i> { level-1 level-2 level-1-2 }] * 	Perform this step if another routing protocol is running between the MCE device and VPN sites in the VPN instance.

Action	Command	Description
Return to system view.	quit	-
Enter the view of the interface to which the VPN instance is bound.	interface <i>interface-type interface-number</i>	-
Enable IS-IS IPv6 on the interface.	isis ipv6 enable [<i>process-id</i>]	-

9.3.4 Verifying the MCE Configuration

Prerequisites

The MCE configuration is complete.

Procedure

- Run the **display ip vpn-instance** *vpn-instance-name* command to check brief information about a specified VPN instance.
- Run the **display ip vpn-instance verbose** *vpn-instance-name* command to check detailed information about a specified VPN instance.
- Run the **display ip vpn-instance** [*vpn-instance-name*] **interface** command to check brief information about the interface to which a specified VPN instance is bound.
- Run the **display ipv6 routing-table vpn-instance** *vpn-instance-name* [**verbose**] command to check the IPv6 routing table on the MCE device. The routing table contains routes to the LAN and remote sites for each service.

----End

9.4 Configuration Examples for MCE IPv6

This section provides an example for configuring an MCE IPv6 device.

9.4.1 Example for Configuring an MCE IPv6 Device

Networking Requirements

The headquarters and branch of a company need to communicate through MPLS VPN, and two IPv6 services of the company must be isolated. To reduce hardware costs, the company wants the branch to connect to the PE device through one CE device with high capability.

As shown in [Figure 9-3](#), the networking requirements are as follows:

- CE1 and CE2 connect to the headquarters. CE1 belongs to vpna, and CE2 belongs to vpb.

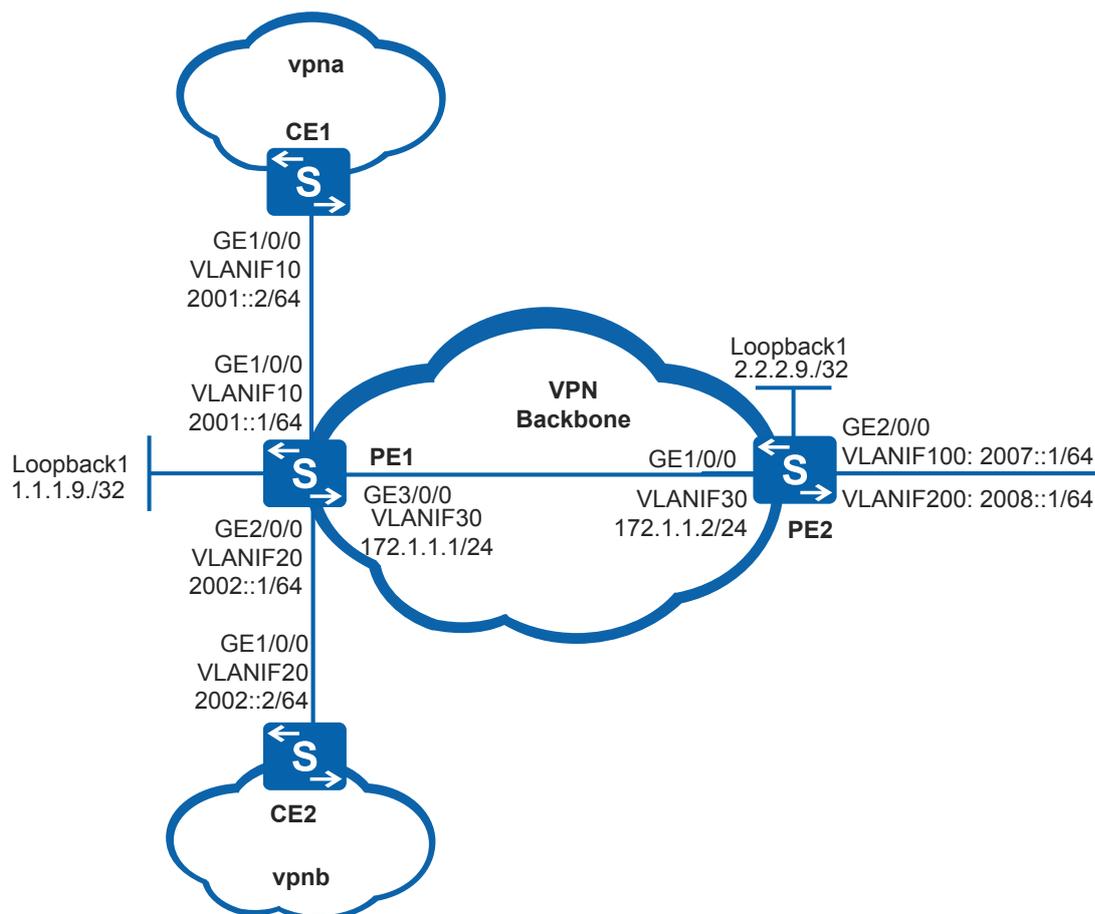
- The MCE device connects to vpna and vpnb of the branch through CE3 and CE4. This implements communication within the same VPN and service isolation between different VPNs.

NOTE

CE1, CE2, CE3 and CE4 are huawei Ethernet Switches S5700.

PE1 and PE2 are huawei Ethernet Switches S9700.

Figure 9-3 MCE IPv6 networking



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF between PE devices to implement interworking between them and configure MP-IBGP to exchange VPN routing information.
2. Enable basic MPLS capabilities and MPLS LDP on the PE devices to set up LDP LSPs.
3. Create VPN instances vpna and vpnb on the MCE and PE devices to isolate services.
4. Set up EBGP peer relationships between PE1 and local CE devices to exchange VPN routes.
5. Configure routing between MCE and sites and between MCE and PE2 to exchange VPN routes.

Procedure

Step 1 Configure OSPF on PE1 and PE2 to implement interworking between them.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Take the display on PE2 as an example:

```
[PE2] display ip routing-table
Route Flags: R - relay,
D - download to fib
-----
Routing Tables: Public
      Destinations : 9          Routes : 9

Destination/Mask    Proto    Pre  Cost           Flags    NextHop         Interface
-----
      1.1.1.9/32     OSPF     10   2              D        172.1.1.1       Vlanif30
      2.2.2.9/32     Direct   0     0              D        127.0.0.1       LoopBack1
      127.0.0.0/8    Direct   0     0              D        127.0.0.1       InLoopBack0
      127.0.0.1/32   Direct   0     0              D        127.0.0.1       InLoopBack0
127.255.255.255/32  Direct   0     0              D        127.0.0.1       InLoopBack0
      172.1.1.0/24   Direct   0     0              D        172.1.1.2       Vlanif30
      172.1.1.2/32   Direct   0     0              D        172.1.1.1       Vlanif30
      172.1.1.255/32 Direct   0     0              D        127.0.0.1       Vlanif30
255.255.255.255/32 Direct   0     0              D        127.0.0.1       InLoopBack0
```

Step 2 Enable basic MPLS capabilities and MPLS LDP on the PE devices to set up LDP LSPs between them.

Configure PE1. The configuration on PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mps] quit
[PE1] mpls ldp
[PE1-mps-ldp] quit
[PE1] interface vlanif 30
[PE1-Vlanif30] mpls
[PE1-Vlanif30] mpls ldp
[PE1-Vlanif30] quit
```

After the configuration is complete, run the **display mpls ldp session** command on the PE devices. You can see that the MPLS LDP session between the PE devices is in Operational state.

Take the display on PE2 as an example:

```
[PE2] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
Peer-ID              Status      LAM  SsnRole  SsnAge      KA-Sent/Rcv
-----
```

```
1.1.1.9:0      Operational DU   Active   0000:00:04   17/17
-----
TOTAL: 1 session(s) Found.
```

Step 3 Enable IPv6. Configure VPN instances on the PE devices. On PE1, bind the VPN instances to the interfaces connected to CE1 and CE2 respectively. On PE2, bind the VPN instances to the interfaces connected to the MCE device.

Configure PE1.

```
[PE1] ipv6
[PE1] vlan batch 10 20
[PE1] interface gigabitEthernet 1/0/0
[PE1-GigabitEthernet1/0/0] port hybrid pvid vlan 10
[PE1-GigabitEthernet1/0/0] port hybrid untagged vlan 10
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitEthernet 2/0/0
[PE1-GigabitEthernet2/0/0] port hybrid pvid vlan 20
[PE1-GigabitEthernet2/0/0] port hybrid untagged vlan 20
[PE1-GigabitEthernet2/0/0] quit
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv6-family
[PE1-vpn-instance-vpna-af-ipv6] route-distinguisher 100:1
[PE1-vpn-instance-vpna-af-ipv6] vpn-target 111:1 both
[PE1-vpn-instance-vpna-af-ipv6] quit
[PE1-vpn-instance-vpna] quit
[PE1] ip vpn-instance vpb
[PE1-vpn-instance-vpb] ipv6-family
[PE1-vpn-instance-vpb-af-ipv6] route-distinguisher 100:2
[PE1-vpn-instance-vpb-af-ipv6] vpn-target 222:2 both
[PE1-vpn-instance-vpb-af-ipv6] quit
[PE1-vpn-instance-vpb] quit
[PE1] interface vlanif 10
[PE1-Vlanif10] ip binding vpn-instance vpna
[PE1-Vlanif10] ipv6 enable
[PE1-Vlanif10] ipv6 address 2001::1 64
[PE1-Vlanif10] quit
[PE1] interface vlanif 20
[PE1-Vlanif20] ip binding vpn-instance vpb
[PE1-Vlanif20] ipv6 enable
[PE1-Vlanif20] ipv6 address 2002::1 64
[PE1-Vlanif20] quit
```

Configure PE2.

```
[PE2] ipv6
[PE2] vlan batch 100 200
[PE2] interface gigabitEthernet 2/0/0
[PE2-GigabitEthernet2/0/0] port link-type trunk
[PE2-GigabitEthernet2/0/0] port trunk allow-pass vlan 100 200
[PE2-GigabitEthernet2/0/0] quit
[PE2] ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv6-family
[PE2-vpn-instance-vpna-af-ipv6] route-distinguisher 200:1
[PE2-vpn-instance-vpna-af-ipv6] vpn-target 111:1 both
[PE2-vpn-instance-vpna-af-ipv6] quit
[PE2-vpn-instance-vpna] quit
[PE2] ip vpn-instance vpb
[PE2-vpn-instance-vpb] ipv6-family
[PE2-vpn-instance-vpb-af-ipv6] route-distinguisher 200:2
[PE2-vpn-instance-vpb-af-ipv6] vpn-target 222:2 both
[PE2-vpn-instance-vpb-af-ipv6] quit
[PE2-vpn-instance-vpb] quit
[PE2] interface vlanif 60
[PE2-Vlanif60] ip binding vpn-instance vpna
[PE2-Vlanif60] ipv6 enable
[PE2-Vlanif60] ipv6 address 2007::1 64
[PE2-Vlanif60] quit
[PE2] interface vlanif 70
```

```
[PE2-Vlnaif70] ip binding vpn-instance vpnb
[PE2-Vlnaif70] ipv6 enable
[PE2-Vlnaif70] ipv6 address 2008::1 64
[PE2-Vlnaif70] quit
```

Step 4 Enable IPv6 and configure VPN instances on the MCE device. Bind the vpn instances to the interfaces connected to CE3 and CE4 respectively.

```
<Huawei> system-view
[Huawei] sysname MCE
[MCE] ipv6
[MCE] ip vpn-instance vpna
[MCE-vpn-instance-vpna] ipv6-family
[MCE-vpn-instance-vpna-af-ipv6] route-distinguisher 100:1
[MCE-vpn-instance-vpna-af-ipv6] quit
[MCE-vpn-instance-vpna] quit
[MCE] ip vpn-instance vpnb
[MCE-vpn-instance-vpnb] ipv6-family
[MCE-vpn-instance-vpnb-af-ipv6] route-distinguisher 200:2
[MCE-vpn-instance-vpnb-af-ipv6] quit
[MCE-vpn-instance-vpnb] quit
[MCE] interface gigabitethernet 1/0/0.1
[MCE-GigabitEthernet1/0/0.1] ip binding vpn-instance vpna
[MCE-GigabitEthernet1/0/0.1] dot1q termination vid 100
[MCE-GigabitEthernet1/0/0.1] ipv6 enable
[MCE-GigabitEthernet1/0/0.1] ipv6 address 2007::2 64
[MCE-GigabitEthernet1/0/0.1] ipv6 nd ns multicast-enable
[MCE-GigabitEthernet1/0/0.1] quit
[MCE] interface gigabitethernet 1/0/0.2
[MCE-GigabitEthernet1/0/0.2] ip binding vpn-instance vpnb
[MCE-GigabitEthernet1/0/0.2] dot1q termination vid 200
[MCE-GigabitEthernet1/0/0.2] ipv6 enable
[MCE-GigabitEthernet1/0/0.2] ipv6 address 2008::2 64
[MCE-GigabitEthernet1/0/0.2] ipv6 nd ns multicast-enable
[MCE-GigabitEthernet1/0/0.2] quit
[MCE] interface gigabitethernet 3/0/0
[MCE-GigabitEthernet3/0/0] ip binding vpn-instance vpna
[MCE-GigabitEthernet3/0/0] ipv6 enable
[MCE-GigabitEthernet3/0/0] ipv6 address FC03::2 64
[MCE-GigabitEthernet3/0/0] quit
[MCE] interface gigabitethernet 4/0/0
[MCE-GigabitEthernet4/0/0] ip binding vpn-instance vpnb
[MCE-GigabitEthernet4/0/0] ipv6 enable
[MCE-GigabitEthernet4/0/0] ipv6 address FC04::1 64
[MCE-GigabitEthernet4/0/0] quit
```

Step 5 Set up an MP-IBGP peer relationship between PE1 and PE2. Set up an EBGP peer relationship between PE1 and CE1, and between PE1 and CE2.

Configure PE1 to set up an MP-IBGP peer relationship between PE1 and PE2.

```
[PE1] bgp 100
[PE1-bgp] peer 2.2.2.9 as-number 100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 1
[PE1-bgp] ipv6-family vpnv6
[PE1-bgp-af-vpnv6] peer 2.2.2.9 enable
[PE1-bgp-af-vpnv6] quit
```

The configuration of PE2 is the same as the configuration of PE1.

Configure PE1 to set up an EBGP peer relationship between PE1 and CE1, and between PE1 and CE2.

```
[PE1] bgp 100
[PE1-bgp] ipv6-family vpn-instance vpna
[PE1-bgp6-vpna] peer 2001::2 as-number 65410
[PE1-bgp6-vpna] quit
[PE1-bgp] ipv6-family vpn-instance vpnb
[PE1-bgp6-vpnb] peer 2002::2 as-number 65420
```

```
[PE1-bgp6-vpnb] quit
[PE1-bgp] quit
```

Configure CE1.

```
[CE1] bgp 65410
[CE1-bgp] peer 2001::1 as-number 100
[CE1-bgp] ipv6-family unicast
[CE1-bgp-af-ipv6] peer 2001::1 enable
[CE1-bgp-af-ipv6] import-route direct
[CE1-bgp-af-ipv6] quit
[CE1-bgp] quit
```

The configuration of CE2 is the same as the configuration of CE1.

After the configuration is complete, run the **display bgp vpnv6 all peer** command on PE1. The command output shows that the PE1 has set up an IBGP peer relationship with PE2 and EBGP peer relationships with CE1 and CE2. All the peer relationships are in Established state.

```
[PE1] display bgp vpnv6 all peer

BGP local router ID : 1.1.1.9
Local AS number : 100
Total number of peers : 1                Peers in established state : 1

Peer          V    AS  MsgRcvd  MsgSent  OutQ  Up/Down      State PrefRcv
-----
2.2.2.9       4   100     288     287     0 01:19:16  Established    6

Peer of vpn instance :
vpn instance vpna :
2001::2       4 65410      9      11     0 00:04:14  Established    2
vpn instance vpnb :
2002::2       4 65420      9      12     0 00:04:09  Established    2
```

Step 6 Configure routing between the MCE device and VPN sites.

The MCE device directly connects to vpna, and no routing protocol is used in vpna. Configure IPv6 static routes to implement communication between the MCE device and vpna.

- # Configure CE3.

Assign IPv6 address FC05::1/64 to the interface connected to vpna.

```
<Huawei> system-view
[Huawei] sysname CE3
[CE3] ipv6
[CE3] vlan batch 60 80
[CE3] interface gigabitethernet 1/0/0
[CE3-GigabitEthernet1/0/0] port link-type trunk
[CE3-GigabitEthernet1/0/0] port trunk allow-pass vlan 60
[CE3-GigabitEthernet1/0/0] port trunk pvid vlan 60
[CE3-GigabitEthernet1/0/0] quit
[CE3] interface vlanif 60
[CE3-Vlanif60] ipv6 enable
[CE3-Vlanif60] ipv6 address FC03::3 64
[CE3-Vlanif60] quit
[CE3] interface gigabitethernet 2/0/0
[CE3-GigabitEthernet2/0/0] port link-type trunk
[CE3-GigabitEthernet2/0/0] port trunk allow-pass vlan 80
[CE3-GigabitEthernet2/0/0] quit
[CE3] interface vlanif 80
[CE3-Vlanif80] ipv6 enable
[CE3-Vlanif80] ipv6 address FC05::1 64
[CE3-Vlanif80] quit
[CE3] ipv6 route-static 0::0 0 FC03::2
```

- # Configure the MCE device.

```
[MCE] ipv6 route-static vpn-instance vpna FC05:: 64 FC03::3
```

- # Check the IPv6 routes of vpna on the MCE device.

```
[MCE]display ipv6 routing-table vpn-instance vpna
Routing Table :
vpna
Destinations : 6          Routes :
6
Destination : FC03::          PrefixLength :
64
NextHop      : FC03::2        Preference   :
0
Cost         : 0              Protocol     :
Direct
RelayNextHop : ::            TunnelID    :
0x0
Interface    : GigabitEthernet3/0/0  Flags       :
D

Destination : FC03::2        PrefixLength :
128
NextHop      : ::1           Preference   :
0
Cost         : 0              Protocol     :
Direct
RelayNextHop : ::            TunnelID    :
0x0
Interface    : GigabitEthernet3/0/0  Flags       :
D

Destination : FC05::          PrefixLength :
64
NextHop      : FC03::3        Preference   :
60
Cost         : 0              Protocol     :
Static
RelayNextHop : ::            TunnelID    :
0x0
Interface    : GigabitEthernet3/0/0  Flags       :
RD

Destination : 2007::          PrefixLength :
64
NextHop      : 2007::2        Preference   :
0
Cost         : 0              Protocol     :
Direct
RelayNextHop : ::            TunnelID    :
0x0
Interface    : GigabitEthernet1/0/0.1  Flags       :
D

Destination : 2007::2        PrefixLength :
128
NextHop      : ::1           Preference   :
0
Cost         : 0              Protocol     :
Direct
RelayNextHop : ::            TunnelID    :
0x0
Interface    : GigabitEthernet1/0/0.1  Flags       :
D

Destination : FE80::          PrefixLength :
```

```

10
  NextHop      : ::                               Preference  :
0
  Cost        : 0                               Protocol    :
Direct
  RelayNextHop : ::                               TunnelID    :
0x0
  Interface    : NULL0                           Flags       :
D
  
```

The preceding information shows that the MCE device has a static route to vpnb.

The RIPng protocol runs in vpnb. Configure RIPng process 200 on the MCE device and bind it to vpnb so that routes learned by RIPng are added to the IPv6 routing table of vpnb.

- # Configure the MCE device.

```

[MCE] ripng 200 vpn-instance vpnb
[MCE-ripng-200] import-route ospfv3 200
[MCE-ripng-200] quit
[MCE] interface gigabitethernet 4/0/0
[MCE-GigabitEthernet4/0/0] ripng 200 enable
[MCE-GigabitEthernet4/0/0] quit
  
```

- # Configure CE4.

Assign IP address FC06::1/64 to the interface connected to vpnb.

```

<Huawei> system-view
[Huawei] sysname CE4
[CE4] ipv6
[CE4] vlan batch 70 80
[CE4] interface gigabitethernet 1/0/0
[CE4-GigabitEthernet1/0/0] port link-type trunk
[CE4-GigabitEthernet1/0/0] port trunk allow-pass vlan 70
[CE4-GigabitEthernet1/0/0] port trunk pvid vlan 70
[CE4-GigabitEthernet1/0/0] quit
[CE4] interface vlanif 70
[CE4-Vlanif70] ipv6 enable
[CE4-Vlanif70] ipv6 address FC04::3 64
[CE4-Vlanif70] quit
[CE4] interface gigabitethernet 2/0/0
[CE4-GigabitEthernet2/0/0] port link-type trunk
[CE4-GigabitEthernet2/0/0] port trunk allow-pass vlan 80
[CE4-GigabitEthernet2/0/0] quit
[CE4] interface vlanif 80
[CE4-Vlanif80] ipv6 enable
[CE4-Vlanif80] ipv6 address FC06::1 64
[CE4-Vlanif80] quit
[CE4] ripng 200
[CE4-ripng-200] quit
[CE4] interface vlanif 70
[CE4-Vlanif70] ripng 200 enable
[CE4-Vlanif70] quit
[CE4] interface vlanif 80
[CE4-Vlanif80] ripng 200 enable
[CE4-Vlanif80] quit
  
```

- # Check the routes of vpnb on the MCE device.

```

[MCE] display ipv6 routing-table vpn-instance vpnb
Routing Table :
vpnb
Destinations : 6          Routes :
6
Destination : FC04::          PrefixLength :
64
NextHop      : FC04::2        Preference   :
  
```

```

0
Cost          : 0                               Protocol    :
Direct
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : GigabitEthernet4/0/0           Flags       :
D

Destination   : FC04::2                         PrefixLength :
128
NextHop       : ::1                             Preference  :
0
Cost          : 0                               Protocol    :
Direct
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : GigabitEthernet4/0/0           Flags       :
D

Destination   : FC06::                          PrefixLength :
64
NextHop       : FE80::5689:98FF:FE28:8442       Preference  :
100
Cost          : 0                               Protocol    :
RIPng
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : GigabitEthernet4/0/0           Flags       :
D

Destination   : 2008::                          PrefixLength :
64
NextHop       : 2008::2                         Preference  :
0
Cost          : 0                               Protocol    :
Direct
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : GigabitEthernet1/0/0.2         Flags       :
D

Destination   : 2008::2                         PrefixLength :
128
NextHop       : ::1                             Preference  :
0
Cost          : 0                               Protocol    :
Direct
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : GigabitEthernet1/0/0.2         Flags       :
D

Destination   : FE80::                          PrefixLength :
10
NextHop       : ::                              Preference  :
0
Cost          : 0                               Protocol    :
Direct
RelayNextHop  : ::                               TunnelID    :
0x0
Interface     : NULL0                           Flags       :
D
  
```

The preceding information shows that the MCE device has learned the route to `vpnb` through RIPng. The route to `vpnb` and the route to `vpna` (FC05::/64) are maintained in different VPN routing tables so that users in the two VPNs are isolated from each other.

Step 7 Configure OSPFv3 multi-instance between the MCE device and PE2.

Configure PE2.

NOTE

To configure OSPFv3 multi-instance between the MCE device and PE2, complete the following tasks on PE2:

- In the OSPFv3 view, import BGP routes and advertise VPN routes of PE1 to the MCE device.
- In the BGP view, import routes of the OSPFv3 processes and advertise the VPN routes of the MCE device to PE1.

```
[PE2] ospfv3 100 vpn-instance vpna
[PE2-ospfv3-100] import-route bgp
[PE2-ospf-100] quit
[PE2] ospfv3 200 vpn-instance vpb
[PE2-ospfv3-200] import-route bgp
[PE2-ospf-200] quit
[PE2] bgp 100
[PE2-bgp] ipv6-family vpn-instance vpna
[PE2-bgp-vpna] import-route ospfv3 100
[PE2-bgp-vpna] quit
[PE2-bgp] ipv6-family vpn-instance vpb
[PE2-bgp-vpb] import-route ospfv3 200
[PE2-bgp-vpb] quit
[PE2] interface vlanif 100
[PE2-Vlanif100] ospfv3 100 area 0
[PE2-Vlanif100] quit
[PE2] interface vlanif 200
[PE2-Vlanif200] ospfv3 200 area 0
[PE2-Vlanif200] quit
```

Configure the MCE device.

NOTE

Import VPN routes to the OSPFv3 processes.

```
[MCE] ospfv3 100 vpn-instance vpna
[MCE-ospfv3-100] router-id 1.1.1.1
[MCE-ospfv3-100] import-route static
[MCE-ospfv3-100] vpn-instance-capability simple
[MCE-ospfv3-100] quit
[MCE] ospfv3 200 vpn-instance vpb
[MCE-ospfv3-200] router-id 2.2.2.2
[MCE-ospfv3-200] import-route ripng 200
[MCE-ospfv3-200] vpn-instance-capability simple
[MCE-ospfv3-200] quit
[MCE] interface gigabitethernet 1/0/0.1
[MCE-GigabitEthernet1/0/0.1] ospfv3 100 area 0
[MCE-GigabitEthernet1/0/0.1] quit
[MCE] interface gigabitethernet 1/0/0.2
[MCE-GigabitEthernet1/0/0.2] ospfv3 200 area 0
[MCE-GigabitEthernet1/0/0.2] quit
```

Step 8 Verify the configuration.

After the configuration is complete, run the **display ipv6 routing-table vpn-instance** command on the MCE device to view the routes to the remote CE devices.

Take the routing table of `vpna` as an example:

```
[MCE] display ipv6 routing-table vpn-instance vpna
Routing Table : vpna
Destinations : 7          Routes : 7

Destination : 2001::          PrefixLength : 64
NextHop     : FE80::5689:98FF:FE28:8472 Preference : 150
Cost       : 0                Protocol   : OSPFv3ASE
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet1/0/0.1 Flags     : D

Destination : FC03::          PrefixLength : 64
NextHop     : FC03::2        Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet3/0/0  Flags     : D

Destination : FC03::2        PrefixLength : 128
NextHop     : ::1            Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet3/0/0  Flags     : D

Destination : FC05::          PrefixLength : 64
NextHop     : FC03::3        Preference : 60
Cost       : 0                Protocol   : Static
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet3/0/0  Flags     : RD

Destination : 2007::          PrefixLength : 64
NextHop     : 2007::2        Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet1/0/0.1 Flags     : D

Destination : 2007::2        PrefixLength : 128
NextHop     : ::1            Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : GigabitEthernet1/0/0.1 Flags     : D

Destination : FE80::          PrefixLength : 10
NextHop     : ::            Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : NULL0           Flags     : D
```

Run the **display ipv6 routing-table vpn-instance** command on the PE devices to view the routes to the remote CE devices.

Take the VPN routing table of vpna on PE as an example:

```
[PE1] display ipv6 routing-table vpn-instance vpna
Routing Table : vpna
Destinations : 6          Routes : 6

Destination : 2001::          PrefixLength : 64
NextHop     : 2001::1        Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : Vlanif30       Flags     : D

Destination : 2001::1        PrefixLength : 128
NextHop     : ::1            Preference : 0
Cost       : 0                Protocol   : Direct
RelayNextHop : ::            TunnelID  : 0x0
Interface  : Vlanif30       Flags     : D

Destination : FC03::          PrefixLength : 64
NextHop     : ::FFFF:2.2.2.9 Preference : 255
```

```

Cost          : 0                               Protocol     : BGP
RelayNextHop  : ::                               TunnelID     : 0x0
Interface     : Vlanif30                         Flags        : RD

Destination   : FC05::                          PrefixLength : 64
NextHop       : ::FFFF:2.2.2.9                   Preference   : 255
Cost          : 0                               Protocol     : BGP
RelayNextHop  : ::                               TunnelID     : 0x0
Interface     : Vlanif30                         Flags        : RD

Destination   : 2007::                          PrefixLength : 64
NextHop       : ::FFFF:2.2.2.9                   Preference   : 255
Cost          : 0                               Protocol     : BGP
RelayNextHop  : ::                               TunnelID     : 0x0
Interface     : Vlanif30                         Flags        : RD

Destination   : FE80::                          PrefixLength : 10
NextHop       : ::                               Preference   : 0
Cost          : 0                               Protocol     : Direct
RelayNextHop  : ::                               TunnelID     : 0x0
Interface     : NULL0                            Flags        : D
  
```

CE1 and CE3 can communicate with each other. CE2 and CE4 can communicate with each other.

CE1 cannot ping CE2 or CE4. CE3 cannot ping CE2 or CE4.

----End

Configuration Files

- CE1 configuration file

```

#
sysname CE1
#
vlan batch 10
#
ipv6
#
interface Vlanif10
  ipv6 enable
  ipv6 address 2001::2/64
#
interface GigabitEthernet1/0/0
  port hybrid pvid vlan 10
  port hybrid untagged vlan 10
#
bgp 65410
  peer 2001::1 as-number 100
#
  ipv6-family unicast
    undo synchronization
    import-route direct
    peer 2001::1 enable
#
return
  
```

- CE2 configuration file

```

#
sysname CE2
#
ipv6
#
vlan batch 20
#
interface Vlanif20
  ipv6 enable
  
```

```
ipv6 address 2002::2/64
#
interface GigabitEthernet1/0/0
port hybrid pvid vlan 20
port hybrid untagged vlan 20
#
bgp 65420
peer 2002::1 as-number 100
#
ipv6-family unicast
undo synchronization
import-route direct
peer 2002::1 enable
#
return
```

● PE1 configuration file

```
#
sysname PE1
#
ipv6
#
vlan batch 10 20 30
#
ip vpn-instance vpna
ipv6-family
route-distinguisher 100:1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
ipv6-family
route-distinguisher 100:2
vpn-target 222:2 export-extcommunity
vpn-target 222:2 import-extcommunity
#
mpls lsr-id 1.1.1.9
mpls
#
mpls ldp
#
interface Vlanif10
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001::1/64
#
interface Vlanif20
ip binding vpn-instance vpnb
ipv6 enable
ipv6 address 2002::1/64
#
interface Vlanif30
ip address 172.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet1/0/0
port hybrid pvid vlan 10
port hybrid untagged vlan 10
#
interface GigabitEthernet2/0/0
port hybrid pvid vlan 20
port hybrid untagged vlan 20
#
interface GigabitEthernet3/0/0
port hybrid pvid vlan 30
port hybrid untagged vlan 30
#
interface LoopBack1
```

```
ip address 1.1.1.9 255.255.255.255
#
bgp 100
peer 2.2.2.9 as-number 100
peer 2.2.2.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 2.2.2.9 enable
#
ipv6-family vpnv6
policy vpn-target
peer 2.2.2.9 enable
#
ipv6-family vpn-instance vpna
peer 2001::2 as-number 65410
import-route direct
#
ipv6-family vpn-instance vpnb
peer 2002::2 as-number 65420
import-route direct
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● PE2 configuration file

```
#
sysname PE2
#
ipv6
#
vlan batch 30 100 200
#
ip vpn-instance vpna
ipv6-family
route-distinguisher 200:1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
ip vpn-instance vpnb
ipv6-family
route-distinguisher 200:2
vpn-target 222:2 export-extcommunity
vpn-target 222:2 import-extcommunity
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
ospfv3 100 vpn-instance vpna
router-id 3.3.3.3
import-route bgp
#
ospfv3 200 vpn-instance vpnb
router-id 4.4.4.4
import-route bgp
#
interface vlanif30
ip address 172.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface Vlanif100
ip binding vpn-instance vpna
ipv6 enable
```

```
ipv6 address 2007::1/64
ospfv3 100 area 0.0.0.0
#
interface Vlanif200
ip binding vpn-instance vpnb
ipv6 enable
ipv6 address 2008::1/64
ospfv3 200 area 0.0.0.0
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
interface GigabitEthernet1/0/0
port hybrid pvid vlan 30
port hybrid untagged vlan 30
#
interface GigabitEthernet2/0/0
port link-type trunk
port trunk allow-pass vlan 100 200
#
bgp 100
peer 1.1.1.9 as-number 100
peer 1.1.1.9 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
peer 1.1.1.9 enable
#
ipv6-family vpnv6
policy vpn-target
peer 1.1.1.9 enable
#
ipv6-family vpn-instance vpna
import-route ospfv3 100
#
ipv6-family vpn-instance vpnb
import-route ospfv3 200
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

● MCE configuration file

```
#
sysname MCE
#
ipv6
#
ip vpn-instance vpna
ipv6-family
route-distinguisher 100:1
#
ip vpn-instance vpnb
ipv6-family
route-distinguisher 200:2
#
ospfv3 100 vpn-instance vpna
router-id 1.1.1.1
import-route static
vpn-instance-capability simple
#
ospfv3 200 vpn-instance vpnb
router-id 2.2.2.2
import-route ripng 200
vpn-instance-capability simple
#
ripng 200 vpn-instance vpnb
```

```
import-route ospfv3 200
#
interface GigabitEthernet1/0/0.1
 dot1q termination vid 100
 ip binding vpn-instance vpna
 ipv6 enable
 ipv6 address 2007::2/64
 ipv6 nd ns multicast-enable
 ospfv3 100 area 0.0.0.0
#
interface GigabitEthernet1/0/0.2
 dot1q termination vid 200
 ip binding vpn-instance vpb
 ipv6 enable
 ipv6 address 2008::2/64
 ipv6 nd ns multicast-enable
 ospfv3 200 area 0.0.0.0
#
interface GigabitEthernet3/0/0
 ip binding vpn-instance vpna
 ipv6 enable
 ipv6 address FC03::2/64
#
interface GigabitEthernet4/0/0
 ip binding vpn-instance vpb
 ipv6 enable
 ipv6 address FC04::1/64
 ripng 200 enable
#
ipv6 route-static vpn-instance vpna FC05:: 64 FC03::3
#
return
```

- CE3 configuration file

```
#
sysname CE3
#
ipv6
#
vlan batch 60 80
#
interface vlanif60
 ipv6 enable
 ipv6 address FC03::3/64
#
interface vlanif80
 ipv6 enable
 ipv6 address FC05::1/64
#
interface GigabitEthernet1/0/0
 port link-type trunk
 port trunk allow-pass vlan 60
 port trunk pvid vlan 60
#
interface GigabitEthernet2/0/0
 port link-type trunk
 port trunk allow-pass vlan 80
#
ipv6 route-static :: 0 FC03::2
#
return
```

- CE4 configuration file

```
#
sysname CE4
#
ipv6
#
vlan batch 70 80
#
```

```
ripng 200
#
interface vlanif70
  ipv6 enable
  ipv6 address FC04::2/64
  ripng 200 enable
#
interface vlanif80
  ipv6 enable
  ipv6 address FC06::1/64
  ripng 200 enable
#
interface GigabitEthernet1/0/0
  port link-type trunk
  port trunk allow-pass vlan 70
  port trunk pvid vlan 70
#
interface GigabitEthernet2/0/0
  port link-type trunk
  port trunk allow-pass vlan 80
#
return
```

10 EVPN Configuration

About This Chapter

This chapter provides an overview of Ethernet virtual private network (EVPN) and describes its basic configurations.

[10.1 Overview of EVPN](#)

[10.2 Understanding EVPN](#)

[10.3 Application Scenarios for EVPN](#)

[10.4 Licensing Requirements and Limitations for EVPN](#)

[10.5 Configuring EVPN Functions](#)

Configuring EVPN functions involves configuring VPN instances, binding interfaces to VPN instances, and configuring EVPN BGP peer relationships.

[10.6 Maintaining EVPN](#)

[10.7 Configuration Examples for EVPN](#)

[10.8 References for EVPN](#)

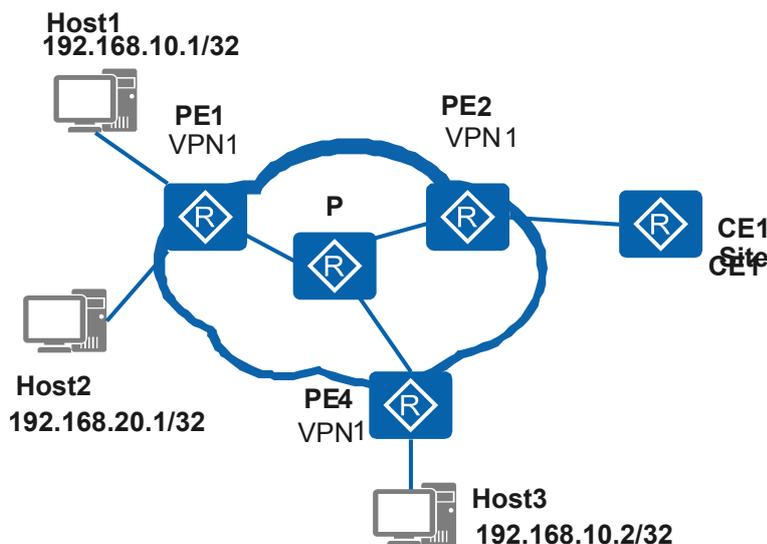
10.1 Overview of EVPN

Definition

An Ethernet virtual private network (EVPN) is a VPN used for Layer 2 interworking. EVPN is similar to Border Gateway Protocol (BGP)/Multiprotocol Label Switching (MPLS) IP VPN. Using extended reachability information, EVPN implements MAC address learning and advertisement between Layer 2 networks at different sites on the control plane rather than on the data plane.

Figure 10-1 shows the basic EVPN model.

Figure 10-1 Basic EVPN model



Purpose

Originally, the Virtual Extensible LAN (VXLAN) solution did not provide the control plane. VXLAN used traffic flooding on the data plane to implement VXLAN tunnel endpoint (VTEP) discovery and host information learning, including IP and MAC addresses, VXLAN Network Identifiers (VNIs), and gateway VTEP IP addresses. This way resulted in high traffic volumes on data center networks. To resolve this problem, VXLAN uses EVPN as the control plane. EVPN allows VTEPs to exchange EVPN routes so that VTEPs can be automatically discovered and host information can be mutually advertised. Therefore, EVPN prevents unnecessary data traffic flooding.

EVPN is similar to BGP/MPLS IP VPN and communicates EVPN routes over a public network, improving security of customers' private data.

Additionally, manual configuration of the original VXLAN solution is time-consuming on a large-scale network. Using EVPN helps reduce the manual configuration workload.

10.2 Understanding EVPN

10.2.1 Implementation

Overview

An EVPN is a VPN used for Layer 2 interworking. EVPN is similar to BGP/MPLS IP VPN. Using extended reachability information, EVPN implements MAC address learning and advertisement between Layer 2 networks at different sites on the control plane rather than on the data plane.

EVPN Routes

EVPN defines a new type of BGP network layer reachability information (NLRI), called the EVPN NLRI. The EVPN NLRI defines new types of EVPN routes for IP address learning and advertisement between Layer 3 networks at different sites.

During dynamic establishment of a VXLAN tunnel, EVPN functions as the VXLAN control plane and uses IP prefix routes defined by the EVPN NLRI to communicate VTEP addresses and host information. Therefore, EVPN implements VTEP discovery and host information learning on the control plane instead of the data plane.

An IP prefix route is type 5 route. The format of EVPN NLRI specific to IP prefix routes is shown in [Figure 10-2](#).

Figure 10-2 Format of EVPN NLRI specific to IP prefix routes

Route Distinguisher (8 bytes)
Ethernet Segment Identifier (10 bytes)
Ethernet Tag ID (4 bytes)
IP Prefix Length (1 byte)
IP Prefix (4 or 16 bytes)
GW IP Address (4 or 16 bytes)
MPLS Label (3 bytes)

The description of each field is as follows:

Field	Description
Route Distinguisher	Indicates the RD value of a VPN instance.
Ethernet Segment Identifier	Uniquely identifies a connection between the current PE and a peer CE.
Ethernet Tag ID	Indicates the actual VLAN ID configured on the current PE.
IP Prefix Length	Indicates the mask length carried in an IP prefix route.
IP Prefix	Indicates the IP prefix address carried in an IP prefix route.
GW IP Address	Indicates the default gateway address. This field is not used in a VXLAN scenario.
MPLS Label	Indicates the Layer 3 VNI carried in an IP prefix route.

The IP Prefix Length and IP Prefix fields carry information of a host IP address or network segment address to identify a host or network segment.

- If the fields carry information of a host IP address, the information is used to advertise host IP routes on the control plane.
- If the fields carry information of a network segment address, the information is used for hosts in a VXLAN to access an external network.

Related Concepts

- **VXLAN tunnel endpoint (VTEP)**
A VTEP encapsulates and decapsulates VXLAN packets. It is represented by an NVE. A VTEP connects to a physical network and is assigned a physical network IP address. This IP address is irrelevant to virtual networks.
In VXLAN packets, the source IP address is the local node's VTEP address, and the destination IP address is the peer node's VTEP address. This pair of VTEP addresses corresponds to a VXLAN tunnel.
- **Network Virtualization Edge (NVE)**
An NVE is a network entity that is deployed at the network edge and implements network virtualization functions. NVEs encapsulate and convert VXLAN packets and then establish a Layer 2 overlay virtual network over the Layer 3 infrastructure.
- **VXLAN Network Identifier (VNI)**
A VNI is similar to a VLAN ID and is used to identify a VXLAN segment.
A VNI identifies only one tenant. Even if multiple terminal users belong to the same VNI, they are considered one tenant.
A VNI is associated with a VPN instance to allow VXLAN packets to be forwarded between sub-networks.
- **EVPN-VPN target**
A VPN instance is associated with one or more EVPN-VPN targets. EVPN-VPN targets are classified into the following types:
 - Export EVPN-VPN targets are carried in the EVPN routes to be advertised to remote EVPN peers.
 - Import EVPN-VPN targets are compared with the export EVPN-VPN targets carried in EVPN routes to determine which EVPN routes can be imported to the routing table of the local VPN instance IPv4 address family.EVPN-VPN targets control the sending and receiving of EVPN routes. During EVPN route cross, if one of the export EVPN-VPN targets carried in EVPN routes is the same as the import EVPN-VPN target configured in the local VPN instance IPv4 address family, the EVPN routes can be imported to the local VPN instance IPv4 address family.

Deploying a VXLAN Tunnel Using EVPN

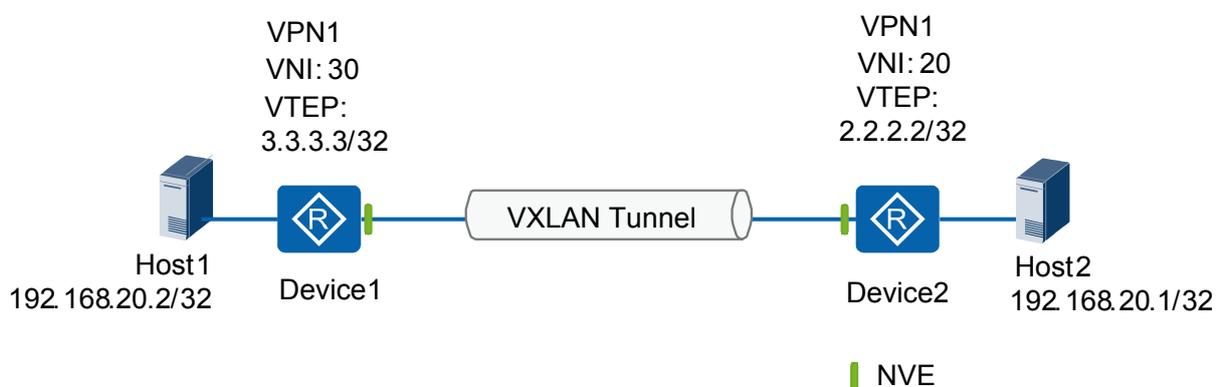
As shown in [Figure 10-3](#), a VXLAN tunnel is established in the following process:

1. Device1 and Device2 each have a VPN instance created, and an EVPN peer relationship is established between them.
2. Device1 and Device2 send IP prefix routes to each other. Device1 imports to its VPN instance the address of Host1 or address of the network segment to which Host1 belongs.

Then Device1 sends an IP prefix route carrying the address to Device2. Device2 also performs the same operation.

3. After Device1 and Device2 receive the IP prefix routes from each other, they each check the export EVPN-VPN target in the IP prefix routes. If the export EVPN-VPN target is the same as the import EVPN-VPN target configured in the local VPN instance IPv4 address family, they each accept the IP prefix route sent by the peer device. If the export and import EVPN-VPN targets are different, they each discard the IP prefix route. After accepting the IP prefix routes, Device1 and Device2 each save the peer VTEP IP address and VNI carried in the IP prefix routes. During packet forwarding, the saved information is encapsulated in the outer layer of the data packets, and then they are transmitted through the VXLAN tunnel.

Figure 10-3 Deploying a VXLAN tunnel using EVPN



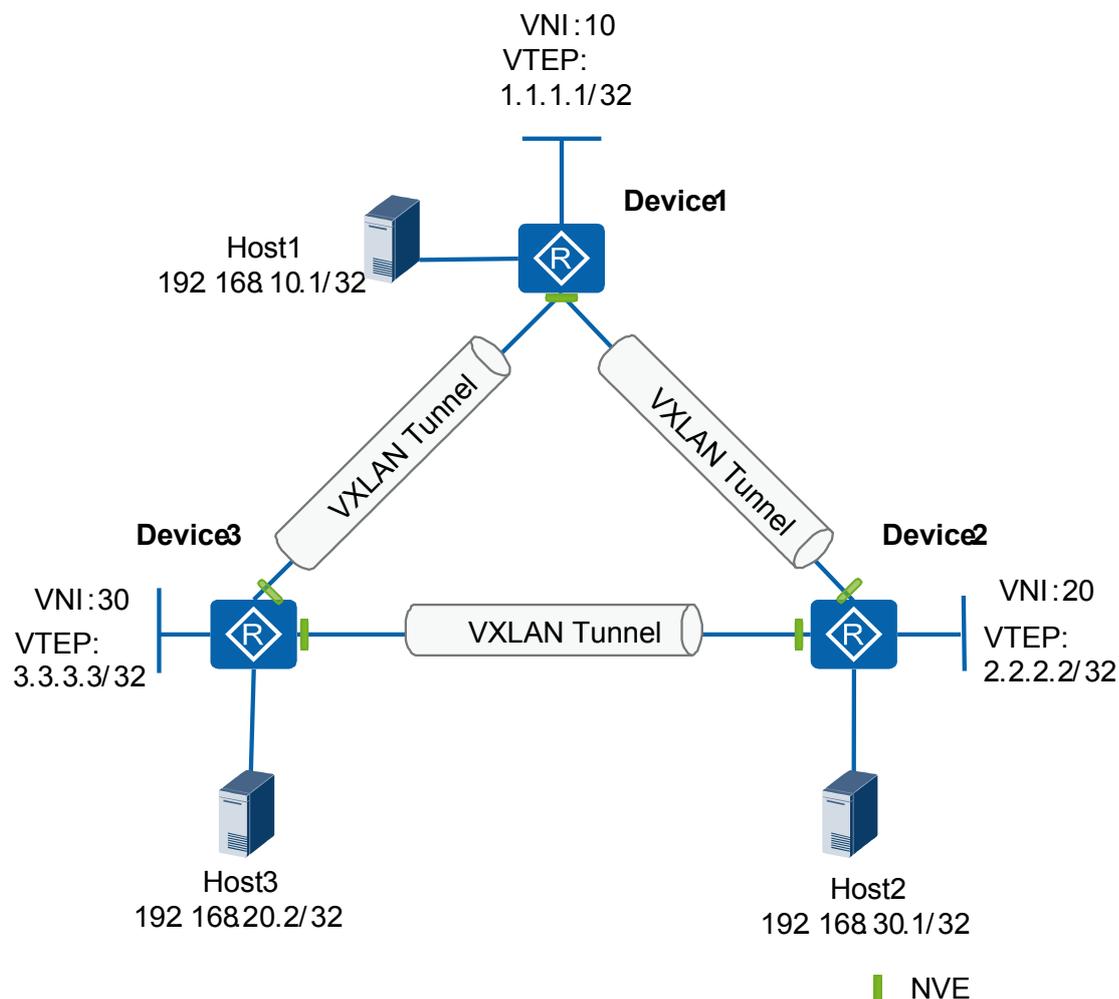
10.3 Application Scenarios for EVPN

10.3.1 EVPN Applications

Service Overview

Figure 10-4 shows the basic networking in which EVPN functions as the VXLAN control plane. A VXLAN tunnel is established using EVPN. Specifically, an EVPN peer relationship is established between two VTEPs. The VTEPs exchange EVPN routes carrying the VNIs and IP addresses of the VTEPs to establish a VXLAN tunnel.

Figure 10-4 EVPN networking application



Networking Description

Device1, Device2, and Device3 each have a VPN instance created. An EVPN peer relationship is configured between every two peers to exchange EVPN routes so that a corresponding VXLAN tunnel is established between every two peers. As a result, Host1, Host2, and Host3 communicate with each other through the VXLAN tunnels.

10.4 Licensing Requirements and Limitations for EVPN

Involved Network Elements

None.

License Requirements

EVPN is a basic feature of the device and is not under license control.

Feature Limitations

When configuring Static EVPN on the router, pay attention to the following points:

Only the AR100-S, AR120-S, AR150-S, AR160-S, AR200-S, AR1200-S series and AR2220E-S support EVPN.

10.5 Configuring EVPN Functions

Configuring EVPN functions involves configuring VPN instances, binding interfaces to VPN instances, and configuring EVPN BGP peer relationships.

10.5.1 Before You Start

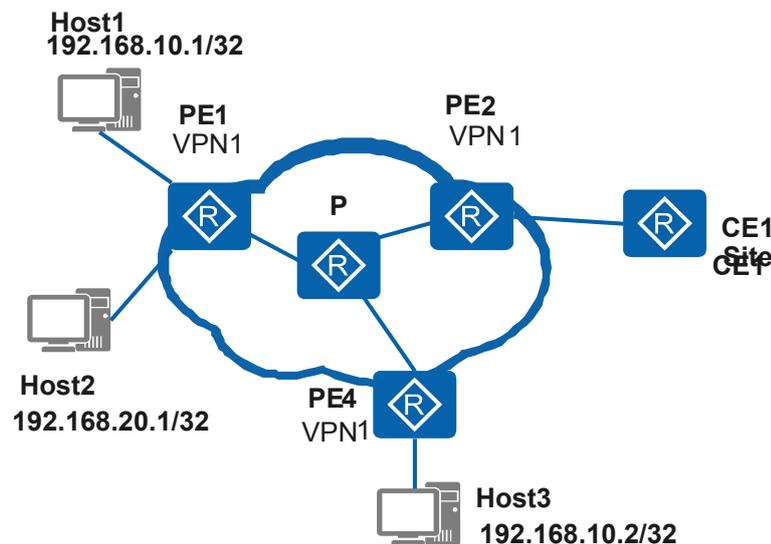
Before configuring Ethernet virtual private network (EVPN), familiarize yourself with the usage scenario, complete the pre-configuration tasks, and obtain the required data. This can help you complete the configuration task quickly and accurately.

Usage Scenario

Functioning as the control plane of Virtual Extensible LAN (VXLAN), EVPN allows EVPN routes to be exchanged so that VXLAN tunnel endpoints (VTEPs) can be automatically discovered and host information can be mutually advertised.

This section describes the EVPN networking configuration. As shown in [Figure 10-5](#), the hosts need to communicate with the Layer 3 network between sites. EVPN can be configured to meet this requirement.

Figure 10-5 EVPN networking model



Pre-configuration Tasks

Before configuring EVPN functions, complete the following tasks:

- Configure an IGP on the PEs and Ps of a backbone network to ensure IP connectivity.
- Configure IP addresses for the interfaces that connect CEs to PEs.

Data Preparation

To configure EVPN functions, you need the following data.

No.	Data
1	VPN instance data, including: <ul style="list-style-type: none">● VPN instance name● RD and EVPN-VPN target of the VPN instance IPv4 address family
2	IP addresses of interfaces that connect PEs to CEs
3	IP addresses of interfaces that connect CEs to PEs
4	AS numbers of PEs, and interfaces and IP addresses used by PEs to establish Border Gateway Protocol (BGP) peer relationships
5	Type of the routing protocol running between PEs and CEs, such as static route, RIP, OSPF, IS-IS, or BGP

10.5.2 Configuring a VPN Instance

Configure VPN instances on PEs to manage EVPN routes.

Context

VPN instances isolate EVPN routes from public routes. The routes of VPN instances are also isolated from each other. VPN instances are required in all EVPN networking solutions. Perform the following steps on each PE:

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **ip vpn-instance** *vpn-instance-name*

A VPN instance is created, and its view is displayed.

Step 3 Run **ipv4-family**

The IPv4 address family is enabled for the VPN instance, and the VPN instance IPv4 address family view is displayed.

Step 4 Run **route-distinguisher** *route-distinguisher*

A route distinguisher (RD) is configured for the VPN instance.

A VPN instance takes effect only after the RD is configured. The RDs of different VPN instances on a PE must be different.

Step 5 Run **vpn-target** *vpn-target* &<1-8> [**both** | **export-extcommunity** | **import-extcommunity**] **evpn**

EVPN-VPN targets are configured for the VPN instance.

An EVPN-VPN target is a BGP extended community attribute used to control the receiving and advertisement of EVPN routes. A maximum of eight EVPN-VPN targets can be configured using the **vpn-target evpn** command at a time. To configure more EVPN-VPN targets for an EVPN instance address family, run the **vpn-target evpn** command several times.

Step 6 (Optional) Run **export route-policy** *policy-name* **evpn**

An export route-policy is associated with the VPN instance IPv4 address family to filter the EVPN routes that the VPN instance IPv4 address family advertises to the EVPN address family.

An export route-policy must be configured to precisely control EVPN routes. An export route-policy filters routes before they are advertised to the EVPN address family.

Step 7 (Optional) Run **import route-policy** *policy-name* **evpn**

An import route-policy is associated with the VPN instance IPv4 address family to filter the EVPN routes imported from the EVPN address family.

An import route-policy must also be configured to precisely control EVPN routes. An import route-policy filters routes received from the EVPN address family.

Step 8 Run **quit**

Return to the system view.

Step 9 Run **bgp**

The BGP view is displayed.

Step 10 Run **ipv4-family vpn-instance** *vpn-instance-name*

The BGP-VPN instance IPv4 address family view is displayed.

Step 11 Run **advertise l2vpn evpn**

The VPN instance is enabled to advertise IP routes to the BGP EVPN address family.

----End

10.5.3 Binding an Interface to a VPN Instance

After an interface is bound to a VPN instance, the interface becomes a part of the VPN. Packets entering the interface will be forwarded based on the virtual routing and forwarding (VRF) table of the VPN.

Context

After a VPN instance is configured on a PE, an interface that belongs to the VPN must be bound to the VPN instance. Otherwise, the interface functions as a public network interface and cannot forward VPN data.

Perform the following steps on the PEs that are connected to CEs:

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The view of the interface to be bound to a VPN instance is displayed.

Step 3 Run **ip binding vpn-instance** *vpn-instance-name*

The interface is bound to the VPN instance.

NOTE

- Running the **ip binding vpn-instance** command deletes Layer 3 IPv4 and IPv6 configurations, such as the IP address and routing protocol on the interface. Reconfigure them if needed.
- An interface cannot be bound to a VPN instance that has no address family enabled.

Step 4 Run **ip address** *ip-address { mask | mask-length }*

The interface IP address is configured.

---End

10.5.4 Configuring an EVPN BGP Peer Relationship

After two PEs establish an EVPN BGP peer relationship, they can exchange EVPN routes.

Context

In EVPN networking, PEs need to have EVPN BGP peer relationships established before they can exchange EVPN route information and implement communication between EVPN instances.

Perform the following steps on each PE:

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** *as-number*

The BGP view is displayed.

Step 3 Run **peer** *ipv4-address as-number { as-number-plain | as-number-dot }*

An EVPN BGP peer IP address is specified.

Step 4 Run **peer** *ipv4-address connect-interface loopback interface-number*

The interface on which a TCP connection to the specified peer is to be established is specified.

Step 5 Run **l2vpn-family evpn**

The BGP-EVPN address family is enabled, and its view is displayed.

Step 6 Run **peer** { *ipv4-address* | *group-name* } **enable**

The capability to exchange EVPN routes with the specified peer or peer group is enabled.

Step 7 (Optional) Run **peer** *ipv4-address* **group** *group-name*

The EVPN BGP peer is added to a peer group.

Adding EVPN BGP peers to peer groups simplifies BGP network configuration and management.

Step 8 (Optional) Run **peer** { *group-name* | *ipv4-address* } **route-policy** *route-policy-name* **export**

A route-policy is specified for an EVPN BGP peer or peer group to advertise only specified routes.

An export route-policy must be configured to precisely control EVPN routes. An export route-policy filters routes before they are advertised to other EVPN BGP peers or peer groups.

Step 9 (Optional) Run **peer** { *ipv4-address* | *group-name* } **route-policy** *route-policy-name* **import**

A route-policy is specified for an EVPN BGP peer or peer group to receive only specified routes.

An import route-policy must also be configured to precisely control EVPN routes. An import route-policy filters routes that are received from other EVPN BGP peers or peer groups.

Step 10 (Optional) Run **undo policy vpn-target**

EVPN-VPN target-based filtering for received EVPN routes is disabled.

Step 11 (Optional) Run **peer** { *group-name* | *ipv4-address* } **mac-limit** *mac-limit* [**idle-forever** | **idle-timeout** *times*]

The maximum number of MAC advertisement routes that can be received from each peer is configured.

If an EVPN instance imports many invalid MAC advertisement routes from some peers and these routes occupy a large proportion of the total MAC advertisement routes, run this command to configure the maximum number of MAC advertisement routes that can be received from each peer.

----End

10.5.5 Verifying the EVPN Configuration

Prerequisites

EVPN functions have been configured.

Procedure

- Run the **display bgp evpn group** [*group-name*] command to check information about an EVPN BGP peer group.
- Run the **display bgp evpn peer** [[*ipv4-address*] **verbose**] command to check information about peers in the EVPN address family.
- Run the **display bgp evpn routing-table peer statistics** command to check statistics about routes advertised and received by the peers in the EVPN address family.

- Run the **display bgp evpn routing-table** command to check information about EVPN routes.

----End

Example

Run the **display bgp evpn group** [*group-name*] command on a PE to view information about an EVPN BGP peer group.

Display information about an EVPN BGP peer group named **aa**.

```
<Huawei> display bgp evpn group aa

Group in
EVPN:

BGP peer-group: aa Remote AS: 100
Authentication type configured: None
Type : internal
Configured hold timer value: 180
Keepalive timer value: 60
Connect-retry timer value: 32
Minimum route advertisement interval is 0 seconds
PeerSession Members:
  2.2.2.2

Status codes: * - Dynamic
Peer Preferred Value: 0
No routing policy is configured
Peer Members:
  Peer          V      AS      MsgRcvd      MsgSent      OutQ      Up/
  Down         State  PrefRcv
  2.2.2.2      4      100      0            0            0
  00:02:28    Idle  0
```

Run the **display bgp evpn peer** [[*ipv4-address*] **verbose**] command to view information about peers in the EVPN address family.

Display detailed information about the peer 2.2.2.2 in the EVPN address family.

```
<Huawei> display bgp evpn peer 2.2.2.2 verbose

Peer is 2.2.2.2, remote AS 100
Type: IBGP link
BGP version 4, Remote router ID 13.0.0.2
Update-group ID: 0
BGP current state: Established, Up for 00h03m58s
BGP current event: KATimerExpired
BGP last state: OpenConfirm
BGP Peer Up count: 5
Received total routes: 0
Received active routes total: 0
Received mac routes: 0
Advertised total routes: 0
Port: Local - 179 Remote - 65163
Configured: Connect-retry Time: 32 sec
Configured: Min Hold Time: 0 sec
Configured: Active Hold Time: 180 sec
Keepalive Time:60 sec
Received : Active Hold Time: 180 sec
Negotiated: Active Hold Time: 180 sec Keepalive Time:60 sec
Peer optional capabilities:
Peer supports bgp multi-protocol extension
```

```
Peer supports bgp route refresh capability
Peer supports bgp 4-byte-as capability
Address family IPv4 Unicast: advertised and received
Address family VPNv4: advertised
Address family L2VPN EVPN: advertised and received
Received: Total 5 messages
  Update messages          0
  Open messages           1
  KeepAlive messages      4
  Notification messages   0
  Refresh messages        0
Sent: Total 7 messages
  Update messages          0
  Open messages           2
  KeepAlive messages      5
  Notification messages   0
  Refresh messages        0
Authentication type configured:
None
Last keepalive received: 2017-01-03
19:13:02-08:00
Last keepalive sent      : 2017-01-03
19:13:02-08:00
Last update received: 2017-01-03
19:06:15-08:00
Last update sent        : 2017-01-03
19:06:15-08:00
Minimum route advertisement interval is 15
seconds
Optional
capabilities:

Route refresh capability has been
enabled
4-byte-as capability has been
enabled
Connect-interface has been
configured
Peer Preferred Value:
0
Routing policy
configured:
No routing policy is configured
```

Run the **display bgp evpn routing-table peer statistics** command to view statistics about routes advertised and received by the peers in the EVPN address family.

Display statistics about routes advertised and received by the peers in the BGP EVPN address family.

```
<Huawei> display bgp evpn routing-table peer statistics
BGP local router ID : 34.1.1.2
Local AS number : 200 Total number of peers : 7
Number of Peers in established state : 4
Peer          Received routes      Advertised routes
3.3.3.3       3                       5
5.5.5.5       20132                   5
10.1.1.2      0                       0
11.11.11.11   0                       0
16.1.1.1      0                       0
24.1.1.2      312                     5
24.24.24.2    312                     5
```

Run the **display bgp evpn routing-table** command to view information about EVPN routes.

Display IP prefix information about all EVPN routes.

```
<Huawei> display bgp evpn all routing-table prefix-route
Local AS number :100 BGP Local router ID is 189.35.99.225
```

```
Status codes: * - valid, > - best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
EVPN address family:
Number of Ip Prefix Routes : 1
Route Distinguisher: 1:1

      Network (EthTagId/IpPrefix/IpPrefixLen)          NextHop
*>      0:9.9.9.0:24                                   9.9.9.9
VPN-Instance vpn1, Router ID 189.35.99.225:
Total Number of Routes: 1
Network      NextHop      MED      LocPrf      PrefVal      Path/Ogn
i  9.9.9.0/24      9.9.9.9      0          100          0            ?
```

10.6 Maintaining EVPN

10.6.1 Configuring EVPN BGP Soft Reset

Context

EVPN BGP soft reset allows a device to receive EVPN routes from EVPN BGP peers again.

EVPN BGP soft reset performs a soft reset on EVPN BGP connections, which triggers EVPN BGP peers to send EVPN routes to a local device without tearing down the EVPN BGP connections and allows the local device to apply a new filtering policy and refresh the EVPN BGP routing table.

Procedure

- Run the **refresh bgp evpn** { **all** | *peer-address* | **group group-name** } { **export** | **import** } command in the user view to configure EVPN BGP soft reset.

----End

10.6.2 Resetting EVPN BGP Connections

Context

This section describes how to use the **reset bgp** command to reset EVPN BGP connections. Resetting EVPN BGP connections causes BGP peer relationships to be interrupted.



NOTICE

A BGP peer relationship between routers is interrupted if you reset an EVPN BGP connection using the **reset bgp** command. Exercise caution when you reset an EVPN BGP connection.

Procedure

- Run the **reset bgp evpn all** command in the user view to reset all EVPN BGP connections.
- Run the **reset bgp evpn as-number-plain** command in the user view to reset the EVPN BGP connections with a specified AS.

- Run the **reset bgp evpn ipv4-address** command in the user view to reset the EVPN BGP connections with specified BGP peers.
- Run the **reset bgp evpn group group-name** command in the user view to reset the EVPN BGP connections with specified BGP peer groups.

----End

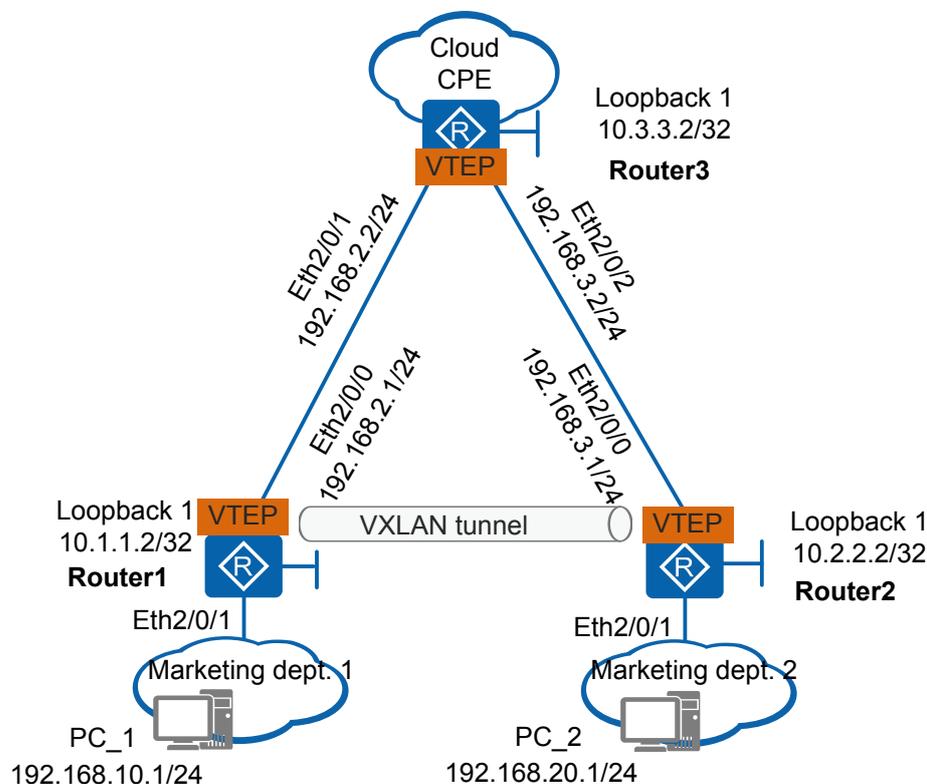
10.7 Configuration Examples for EVPN

10.7.1 Example for Dynamically Establishing a VXLAN Tunnel in BGP EVPN Mode to Implement Communication Between Users in Different Network Segments

Networking Requirements

In **Figure 10-6**, Router1 and Router2 are the branch and headquarters gateways of an enterprise. As users in the headquarters and branch have different service requirements, they are planned in different network segments. PC_1 in the branch and PC_2 in the headquarters belong to VLAN 10 and VLAN 20, respectively. The enterprise requires that users in the headquarters and branch can communicate over a VXLAN tunnel dynamically established using BGP EVPN.

Figure 10-6 Configuring communication between different network segments through a Layer 3 VXLAN gateway



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Router1, Router2, and Router3 to ensure Layer 3 network connectivity.
2. Configure a deployment mode for the VXLAN access service on Router1 and Router2.
3. Establish a BGP EVPN peer relationship.
4. Configure an IP address for the source VTEP on Router1 and Router2.
5. Configure a VPN instance on Router1 and Router2.
6. Configure a Layer 3 gateway on Router1 and Router2.
7. Configure Router1, Router2, and Router3 to advertise IP prefix routes to the BGP peer.

Procedure

Step 1 Configure a routing protocol.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here. When OSPF is used, the 32-bit loopback address of each router must be advertised.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface loopback 1
[Router1-LoopBack1] ip address 10.1.1.2 32
[Router1-LoopBack1] quit
[Router1] interface ethernet 2/0/0
[Router1-Ethernet2/0/0] undo portswitch
[Router1-Ethernet2/0/0] ip address 192.168.2.1 24
[Router1-Ethernet2/0/0] quit
[Router1] ospf
[Router1-ospf-1] area 0
[Router1-ospf-1-area-0.0.0.0] network 10.1.1.2 0.0.0.0
[Router1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Router1-ospf-1-area-0.0.0.0] quit
[Router1-ospf-1] quit
```

After OSPF is configured, the routers can learn the loopback interface address of each other and successfully ping each other. The following shows the ping result from Router1 to Router2.

```
[Router1] ping 10.2.2.2
PING 10.2.2.2: 56 data bytes, press CTRL_C to break
  Reply from 10.2.2.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.2.2.2: bytes=56 Sequence=2 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.2.2.2: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.2.2.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/3/5 ms
```

Step 2 Configure a service access point on Router1 and Router2, respectively.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] bridge-domain 10
[Router1-bd10] quit
[Router1] interface ethernet 2/0/1.1 mode l2
[Router1-Ethernet2/0/1.1] encapsulation dot1q vid 10
[Router1-Ethernet2/0/1.1] bridge-domain 10
[Router1-Ethernet2/0/1.1] quit
```

Step 3 Establish a BGP EVPN peer relationship.

Establish a BGP EVPN peer relationship on Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] bgp 100
[Router1-bgp] peer 10.3.3.2 as-number 100
[Router1-bgp] peer 10.3.3.2 connect-interface LoopBack1
[Router1-bgp] l2vpn-family evpn
[Router1-bgp-af-evpn] peer 10.3.3.2 enable
[Router1-bgp-af-evpn] quit
[Router1-bgp] quit
[Router1] interface nve 1
[Router1-Nve1] source 10.1.1.2
[Router1-Nve1] quit
```

On Router3, establish a BGP EVPN peer relationship with Router1 and Router2.

```
[Router3] bgp 100
[Router3-bgp] peer 10.1.1.2 as-number 100
[Router3-bgp] peer 10.1.1.2 connect-interface LoopBack1
[Router3-bgp] peer 10.2.2.2 as-number 100
[Router3-bgp] peer 10.2.2.2 connect-interface LoopBack1
[Router3-bgp] l2vpn-family evpn
[Router3-bgp-af-evpn] peer 10.1.1.2 enable
[Router3-bgp-af-evpn] peer 10.2.2.2 enable
[Router3-bgp-af-evpn] quit
[Router3-bgp] quit
```

Step 4 Configure a VPN instance on Router1 and Router2.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] ip vpn-instance vpn1
[Router1-vpn-instance-vpn1] ipv4-family
[Router1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1
[Router1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 evpn
[Router1-vpn-instance-vpn1-af-ipv4] quit
[Router1-vpn-instance-vpn1] vxlan vni 5010
[Router1-vpn-instance-vpn1] quit
[Router1] bridge-domain 10
[Router1-bd10] vxlan vni 2010
[Router1-bd10] quit
```

Step 5 Configure a Layer 3 VXLAN gateway on Router1 and Router2 and bind the VPN instance to the gateway.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] interface vbdif 10
[Router1-Vbdif10] ip binding vpn-instance vpn1
[Router1-Vbdif10] ip address 192.168.10.10 24
[Router1-Vbdif10] quit
```

Step 6 Configure Router1, Router2, and Router3 to advertise IP prefix routes to the BGP peer.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here.

```
[Router1] bgp 100
[Router1-bgp] ipv4-family vpn-instance vpn1
[Router1-bgp-vpn1] import-route direct
[Router1-bgp-vpn1] advertise l2vpn evpn
[Router1-bgp-vpn1] quit
[Router1-bgp] quit
```

Step 7 Verify the configuration.

After the configuration is complete, run the **display vxlan tunnel** command on Router1, Router2, and Router3. You can view VXLAN tunnel information. The command output on Router3 is used as an example.

```
[Router3] display vxlan tunnel
```

Tunnel ID	Source	Destination	State	Type
4026531842	10.3.3.2	10.1.1.2	up	dynamic
4026531841	10.3.3.2	10.2.2.2	up	dynamic

```
Number of vxlan tunnel : 2
```

----End

Configuration Files

- Router1 configuration file

```
#
sysname Router1
#

ip vpn-instance
vpn1
  ipv4-
  family
    route-distinguisher
    100:1
  vpn-target 1:1 export-extcommunity
evpn
  vpn-target 1:1 import-extcommunity
evpn
  vxlan vni
  5010
#
bridge-domain
10
  vxlan vni 2010
#

interface
Ethernet2/0/0
  undo
  portswitch
  ip address 192.168.2.1
  255.255.255.0
#

interface Ethernet2/0/1.1 mode
12
  encapsulation dot1q vid
10
  bridge-domain 10
#

interface
LoopBack1
  ip address 10.1.1.2 255.255.255.255
#
```

```
interface
Vbdif10
 ip binding vpn-instance
vpn1
 ip address 192.168.10.10
255.255.255.0
#

interface
Nve1
 source
10.1.1.2
#

bgp
100
 peer 10.3.3.2 as-number
100
 peer 10.3.3.2 connect-interface
LoopBack1
#

 ipv4-family
unicast
 undo
synchronization
 peer 10.3.3.2
enable
#

 l2vpn-family
evpn
 policy vpn-
target
 peer 10.3.3.2
enable
#

 ipv4-family vpn-instance
vpn1
 import-route
direct
 advertise l2vpn
evpn
#
ospf
1
 area
0.0.0.0
 network 10.1.1.2
0.0.0.0
 network 192.168.2.0 0.0.0.255
#

return
```

● Router2 configuration file

```
#
sysname Router2
#

ip vpn-instance
vpn1
 ipv4-
family
 route-distinguisher
100:1
```

```
    vpn-target 1:1 export-extcommunity
evpn
    vpn-target 1:1 import-extcommunity
evpn
    vxlan vni
5020
#
bridge-domain
20
    vxlan vni 2020
#

interface
Ethernet2/0/0
    undo
portswitch
    ip address 192.168.3.1
255.255.255.0
#

interface Ethernet2/0/1.1 mode
l2
encapsulation dot1q vid
20
    bridge-domain 20
#

interface
LoopBack1
    ip address 10.2.2.2 255.255.255.255
#

interface
Vbdif20
    ip binding vpn-instance
vpn1
    ip address 192.168.20.10
255.255.255.0
#

interface
Nve1
    source
10.2.2.2
#

bgp
100
    peer 10.3.3.2 as-number
100
    peer 10.3.3.2 connect-interface
LoopBack1
#

    ipv4-family
unicast
    undo
synchronization
    peer 10.3.3.2
enable
#

    l2vpn-family
evpn
    policy vpn-
target
    peer 10.3.3.2
```

```
enable
#
ipv4-family vpn-instance
vpn1
import-route
direct
advertise l2vpn
evpn
#
ospf
1
area
0.0.0.0
network 10.2.2.2
0.0.0.0
network 192.168.3.0 0.0.0.255
#
return
```

● Router3 configuration file

```
#
sysname Router3
#
interface Ethernet2/0/1
undo
portswitch
ip address 192.168.2.2
255.255.255.0
#
interface Ethernet2/0/2
undo
portswitch
ip address 192.168.3.2
255.255.255.0
#
interface
LoopBack1
ip address 10.3.3.2 255.255.255.255
#
bgp
100
peer 10.1.1.2 as-number
100
peer 10.1.1.2 connect-interface
LoopBack1
peer 10.2.2.2 as-number
100
peer 10.2.2.2 connect-interface
LoopBack1
#
ipv4-family
unicast
undo
synchronization
peer 10.1.1.2
enable
peer 10.2.2.2
enable
#
```

```
l2vpn-family
evpn
  policy vpn-
  target
  peer 10.1.1.2
enable
  peer 10.2.2.2
enable

#

  ipv4-family vpn-instance
  vpn1
  import-route
  direct
  advertise l2vpn
evpn
#
ospf
1
  area
0.0.0.0
  network 10.3.3.2
0.0.0.0
  network 192.168.2.0 0.0.0.255
  network 192.168.3.0 0.0.0.255
#
return
```

10.8 References for EVPN

The following table lists the references for EVPN.

Document No.	Document Name	Protocol Compliance
draft-ietf-bess-evpn-prefix-advertisement-01	IP Prefix Advertisement in EVPN	Partially compliant Compliant only with IP prefix routes.

11 VLL Configuration

About This Chapter

This chapter describes principles, applications, and configurations of the Virtual Leased Line (VLL).

[11.1 Overview of VLL](#)

[11.2 Understanding VLL](#)

This section describes the implementation of VLL.

[11.3 Application Scenarios for VLL](#)

This section describes application scenarios for application scenarios for VLL.

[11.4 Summary of VLL Configuration Tasks](#)

VLLs are classified into three modes: CCC, Martini, SVC. If a network spans multiple ASs, you need to configure inter-AS VLL. You can also configure VLL FRR to provide link-layer reliability for VLL networks.

[11.5 Licensing Requirements and Limitations for VLL](#)

This section describes licensing requirements and limitations for VLL.

[11.6 Default Settings for VLL](#)

This section provides the default settings for VLL.

[11.7 Configuring VLL](#)

This section describes how to configure VLL functions in details.

[11.8 Maintaining VLL](#)

This section describes how to maintain VLL, including collecting, querying, and clearing VLL statistics, checking VLL network connectivity, and resetting BGP TCP connections.

[11.9 Configuration Examples for VLL](#)

This section describes VLL configuration examples including the networking requirements, configuration notes, and configuration roadmap.

[11.10 Troubleshooting VLL](#)

This section describes the common configuration errors and troubleshooting methods.

[11.11 References for VLL](#)

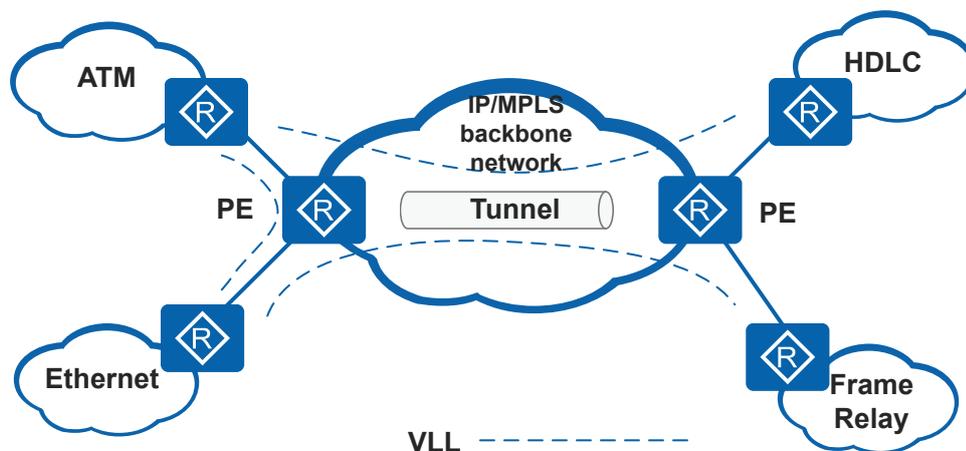
This section lists the references for VLL.

11.1 Overview of VLL

Definition

Virtual leased line (VLL) technology, also called Virtual Private Wire Service (VPWS), emulates leased lines on an IP network to provide a low-cost asymmetrical digital data network (DDN) service. It is a point-to-point (P2P) Layer 2 tunneling technology based on Multiprotocol Label Switching (MPLS), which allows the exchange of data packets between networks using different media. [Figure 11-1](#) shows an example of VLL networking.

Figure 11-1 VLL networking



Purpose

- In network development, various Layer 2 networks have been deployed and are still in use, for example, Ethernet, Asynchronous Transfer Mode (ATM), Frame Relay (FR), and High-level Data Link Control (HDLC) networks. These Layer 2 networks are isolated from one another because they use different Layer 2 protocols. However, there is an increasing demand for direct communication between different Layer 2 networks.
- As Ethernet develops rapidly, more Ethernet connections are required between large cities or remote areas. Traditionally, service providers deploy direct leased lines or set up Layer 2 tunnels between two areas. (The Layer 2 tunnels are set up between Layer 2 switching devices.) This solution has a high cost and is difficult to maintain and extend in some areas.

Service providers need a solution to these problems. VLL technology is introduced to establish a compatible Layer 2 switching network that is cost effective and easy to maintain and extend. Based on MPLS technology, VLL allows multiple customers to share one leased line and creates an exclusive virtual channel for each customer on the shared line. On an Ethernet network, sites in different cities can communicate over a P2P VLL connection on an MPLS network, just like communicating within a virtual local area network (VLAN).

Since Ethernet has replaced most types of Layer 2 networks, VLL is rarely used for communication between different networks, but is widely used for Layer 2 transparent transmission.

VLL is an MPLS-based Layer 2 virtual private network (L2VPN) technology to directly transmit Layer 2 data. VLL establishes P2P VPN tunnels for P2P communication.

Benefits

VLL brings the following benefits:

- Extended network functions and service capabilities for carriers
Carriers can use VLL and enhanced MPLS technologies, such as traffic engineering (TE) and quality of service (QoS), to provide users with differentiated services, meeting diversified user requirements.
- Interconnection between networks using different Layer 2 protocols
VLL allows an Internet service provider (ISP) network to provide connections and switching services using multiple Layer 2 protocols.
- Higher scalability
VLL uses label stacks to identify multiple virtual circuits (VCs) in one label switched path (LSP). Therefore, the P device only needs to maintain information about one LSP, improving system scalability.
- Smaller maintenance workload
VLL provides a method to establish VPNs in large-scale enterprises with large sites and many routes. P devices on the ISP networks only need to forward packets over tunnels on the public network according to MPLS labels and do not need to maintain any Layer 2 information.
- Smooth network upgrade
VLL is transparent to users; therefore, carriers can smoothly upgrade their traditional L2VPN networks, such as ATM and FR networks, to MPLS L2VPN networks, without affecting configurations on customer networks. The network upgrade does not affect user services, except for a brief period of data loss during network migration.

11.2 Understanding VLL

This section describes the implementation of VLL.

11.2.1 Implementation

Basic VLL Architecture

VLL transparently transmits Layer 2 data packets from CE devices over tunnels and provides P2P L2VPN service for users.

The basic VLL architecture consists of three components: attachment circuit (AC), virtual circuit (VC), and tunnel. Pseudo wire (PW) is also a common term used in the VLL service. [Figure 11-2](#) shows the basic VLL architecture.

Figure 11-2 Basic VLL architecture

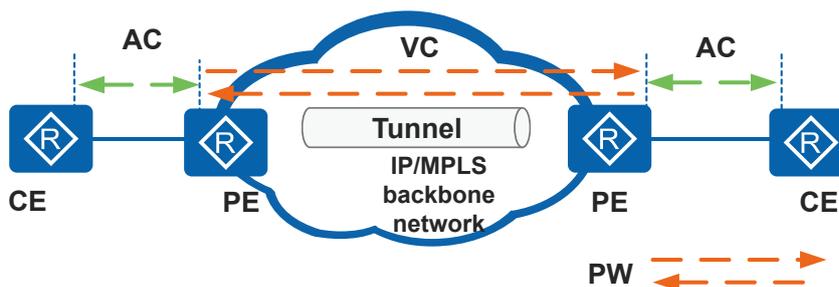


Table 11-1 Description of VLL components

Component	Full Name	Description
AC	Attachment circuit	A connection between a customer and an ISP, namely, a CE-PE link.
VC	Virtual circuit	A unidirectional logical connection between two PE devices.
PW	Pseudo wire	A bidirectional logical connection between two PE devices. A PW, also called a simulated circuit, consists of two unidirectional VCs in forward and reverse directions.
Tunnel	Tunnel	A logical channel that carries one or multiple PWs. A tunnel is a direct channel that transparently transmits data between the local and remote PE devices. It can be an LSP, an MPLS TE tunnel, or a GRE tunnel.

VLL Implementation

VLL implementation involves VLL establishment and VLL packet forwarding.

VLL Establishment

To establish a VLL network, you need to establish a PW and bind the AC with the PW.

1. Establishing a PW: You can configure a static PW between two PE devices, or configure a signaling protocol to enable two PE devices to set up a PW by exchanging VC information. After a PW is established, it is used as a dedicated channel on the public network.
2. Binding the AC to the PW: After a PW is established, bind the AC-side interfaces on the PE devices to the PW to associate the AC with the PW.

VLL Packet Forwarding

After a VLL network is established, packets transmitted on the network undergo encapsulation, transparent transmission, and decapsulation processes.

1. Encapsulation

Before a PE device sends a packet from an AC-side interface to a PW, it processes the packet based on the outer tag type and PW encapsulation mode.

The outer tag of a packet may be a U-Tag or P-Tag.

- A U-Tag is inserted into the packet by a customer device and is irrelevant to services of the service provider (SP). When a GE interface, Ethernet interface, or Eth-Trunk interface is used as the AC-side interface, packets sent from the AC-side interface to a PW carry a U-Tag by default.
- A P-Tag is inserted into the packet by an SP device to distinguish traffic from different users. When a sub-interface or VLANIF interface is used as the AC-side interface, packets sent from the AC-side interface to a PW carry a P-Tag by default.

Two PW encapsulation modes are available: Ethernet encapsulation (raw mode) and VLAN encapsulation (tagged mode).

Table 11-2 describes how a PE device processes a packet sent from an AC-side interface to a PW.

Table 11-2 Processing a packet sent from an AC-side interface to a PW

Packet from an AC-side Interface to a PW	PW Encapsulation Mode	Packet Processing on the PE
Packet with a P-Tag	Ethernet	Removes the P-Tag from the packet and adds two MPLS labels (an inner VC label and an outer tunnel label) before forwarding the packet.
	VLAN	Retains the P-Tag and adds two MPLS labels (an inner VC label and an outer tunnel label) before forwarding the packet.
Packet without a P-Tag	Ethernet/VLAN	Does not process the tag and only adds two MPLS labels (an inner VC label and an outer tunnel label) before forwarding the packet, regardless of which encapsulation mode is used.

2. Transparent transmission

VLL uses an MPLS tunnel to transmit packets. Encapsulated packets are transparently transmitted to the remote PE device over the MPLS tunnel, with their inner VC labels unchanged.

3. Decapsulation

After the remote PE device receives a packet, it decapsulates the packet and forwards the packet to the AC-side interface based on the VC label carried in the packet.

After the packet is decapsulated, the packet is transmitted through the PW to the AC. The remote PE device processes the packet based on the outer tag type and AC-side interface type. **Table 11-3** describes how the PE device processes a packet sent from a PW to an AC-side interface.

Table 11-3 Processing a packet sent from a PW to an AC-side interface

Packet from a PW to an AC-side Interface	VLAN Tag Processing on the PE
Packet with a P-Tag	<p>The PE device processes the packet differently depending on the type of the AC-side interface.</p> <ul style="list-style-type: none"> ● Main interface (Ethernet, GE, or Eth-Trunk interface): does not process the packet. ● VLANIF interface: replaces the P-Tag in the packet. ● Dot1q termination sub-interface: does not process the packet. When the Ethernet encapsulation mode is used, the Dot1q termination sub-interface allows packets from only one VLAN to pass through. ● QinQ termination sub-interface: replaces the P-Tag in the packet.
Packet without a P-Tag	<p>The PE device processes the packet differently depending on the type of the AC-side interface.</p> <ul style="list-style-type: none"> ● Main interface: does not process the packet. ● VLANIF interface: adds a P-Tag to the packet. ● Dot1q termination sub-interface: adds a P-Tag to the packet. When the Ethernet encapsulation mode is used, the Dot1q termination sub-interface allows packets from only one VLAN to pass through. ● QinQ termination sub-interface: adds a P-Tag to the packet.

11.2.2 VLL Modes

11.2.2.1 VLL in CCC Mode

Introduction

A VLL connection in Circuit Cross Connect (CCC) mode is set up through the static configuration.

A CCC connection does not require signaling negotiation or exchange of control packets; therefore, it consumes few resources and is easy to configure. This mode applies to small MPLS networks with simple topologies.

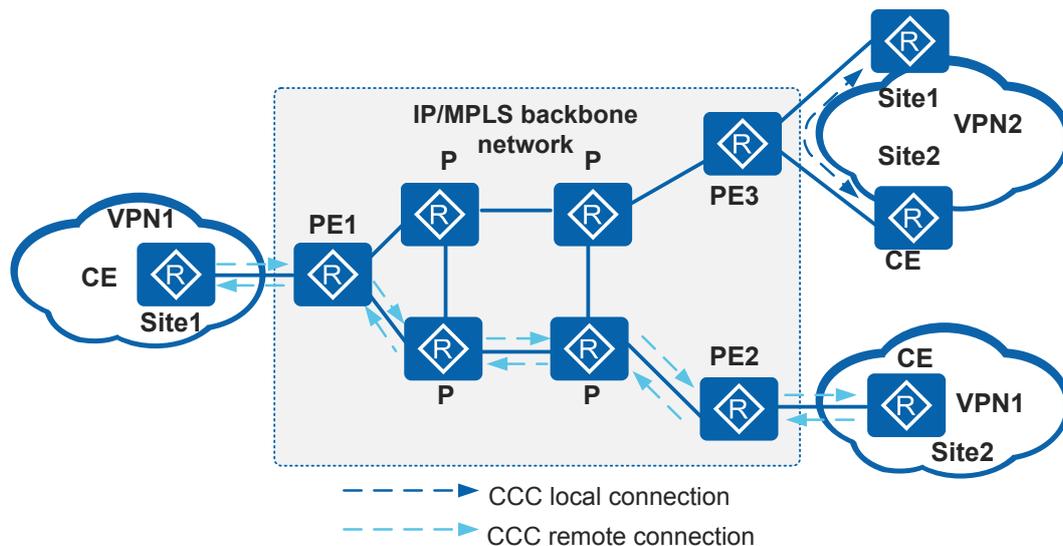
Topology

The CCC mode supports both local and remote connections. [Figure 11-3](#) shows the topology in CCC mode.

NOTE

Currently, the device does not support remote CCC connection.

Figure 11-3 CCC connections



Local connection: Site1 and Site2 of VPN2 are connected through a CCC local connection (the black dashed line). PE3 acts as a Layer 2 switch for Site1 and Site2, and no LSP is required between the CE devices connected to PE3.

Remote connection: Site1 and Site2 of VPN1 are connected through a CCC remote connection (the blue dashed line). Site1 and Site2 require two static LSPs: one from PE1 to PE2 and one from PE2 to PE1. The two blue dashed lines represent a bidirectional PW, or CCC remote connection. This CCC remote connection is similar to a traditional L2VPN connection.

A CCC remote connection uses static VCs and maps L2PDUs received on one end of a VC to a static LSP. The L2PDUs are forwarded along the static LSP hop by hop based on the MPLS configuration and finally reach the other end of the VC. Unlike other VLL modes, the CCC mode uses a single label to transmit data. This label is swapped on each label switching router (LSR). Therefore, each LSP is used exclusively, and two LSPs in forward and reverse directions must be configured for each CCC connection. The LSPs associated with a CCC connection can transmit only the data of this connection and cannot be used for other MPLS L2VPN connections. In addition, the LSPs cannot be used to set up a BGP/MPLS IP VPN connection or transmit common IP packets.

11.2.2.2 VLL in Martini Mode

Introduction

VLL in Martini mode uses the Label Distribution Protocol (LDP) as the signaling protocol to transmit VC information. It complies with RFC4906 and extends LDP by adding a forwarding equivalence class (FEC), VC FEC, for VC label switching. A PE device assigns a VC label to each connection between CE devices. VC labels are carried in L2VPN information and forwarded to a remote PE device through an LDP LSP over the public network. As VLL connections are identified using VC labels, and multiple VC LSPs can be created on an LSP on the public network. The mappings between VC labels and LSPs are saved only on PE devices, while the P devices do not need to maintain any L2VPN information. Therefore,

Martini mode is highly scalable. Additionally, it allows multiple VLL connections to use the same public tunnel, which is not supported by the CCC mode.

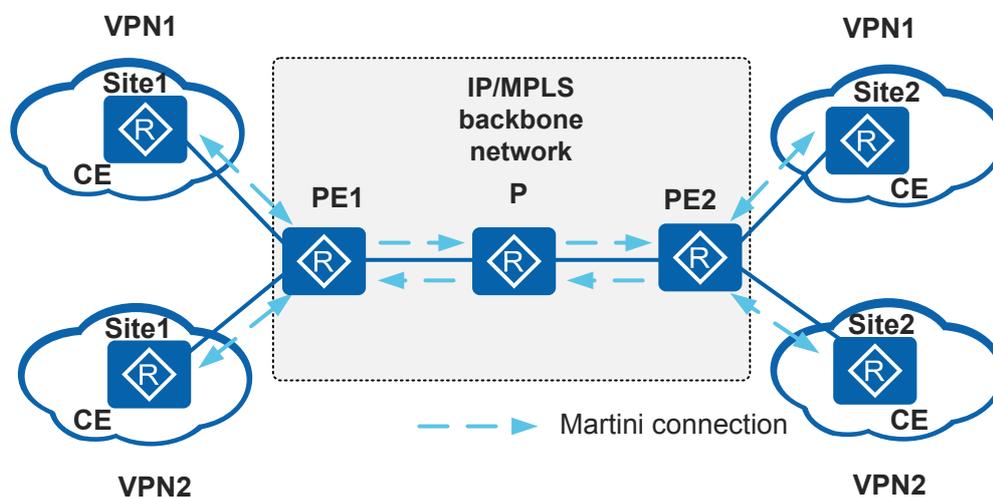
In Martini mode, a VC is identified by its VC type and VC ID.

- VC type: indicates the encapsulation type of a VC, VLAN encapsulation or Ethernet encapsulation.
- VC ID: identifies a VC. VCs of the same type must have different VC IDs on a PE device.

Topology

VLL in Martini mode supports only remote connections. [Figure 11-4](#) shows the topology in Martini mode.

Figure 11-4 Topology in Martini mode



Implementation

Martini implementation involves VLL establishment and VLL packet forwarding. PW establishment is key to VLL establishment. As long as a PW is established, packets can be forwarded.

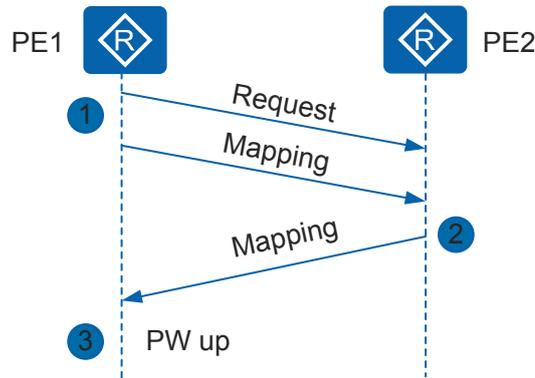
1. **PW Establishment and Teardown**
2. **Packet Forwarding**

The Martini mode uses extended LDP to exchange VC labels. For details about LDP, see [VC Information Exchange](#).

PW Establishment and Teardown

- Establishing a PW
The downstream unsolicited (DU) label distribution mode and liberal label retention mode are used to establish a PW. For details, see LDP LSP Establishment in the *MPLS Configuration*.

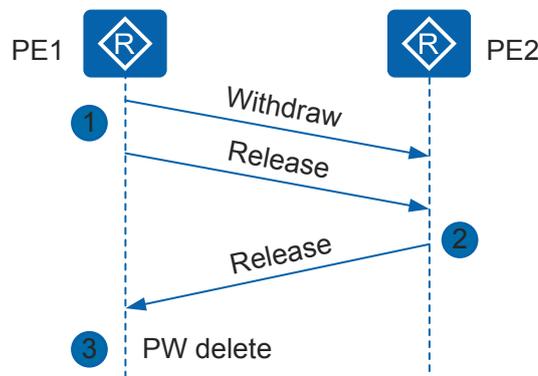
Figure 11-5 Establishing a PW using LDP



After an LDP session is established between the PE devices, a PW is established in the process shown in **Figure 11-5**.

- a. PE1 sends a Request packet to PE2 and sends a Label Mapping message to PE2 in DU mode. The Label Mapping message carries information including the VC label, VC type, VC ID, and interface parameters.
 - b. After PE2 receives the Request packet, it sends a Label Mapping message to PE1. After PE2 receives the Label Mapping message, it compares VC information carried in the message with its own VC information. If they are the same, PE1 and PE2 are in the same VLL. PE2 then accepts the Label Mapping message, and a unidirectional VC1 is established. PE2 knows the inner VC label that it needs to add to packets to sent the packets to PE1.
 - c. After PE1 receives the Label Mapping message from PE2, it processes the message in the same way to establish VC2 in the reverse direction. The two unidirectional VCs constitute a PW.
- Tearing down a PW

Figure 11-6 Tearing down a PW using LDP



When the AC or tunnel goes Down or a VC is deleted, the PW is torn down. **Figure 11-6** shows the process of tearing down a PW.

- a. When PE1 detects that the AC or tunnel has gone Down or a VC has been deleted, it sends a Withdraw message to PE2 to instruct PE2 to delete the VC label. To tear down the PW more quickly, PE1 sends a Withdraw message and a Release message consecutively. The Release message notifies PE2 that PE1 has deleted the VC label.
- b. After receiving the Withdraw and Release messages, PE2 deletes the VC1 label and tears down VC1. PE2 then sends a Release message to PE1 to instruct PE1 to delete the VC2 label.
- c. After receiving the Release message, PE1 deletes the VC2 label and tears down VC2. Then the PW is torn down.

Packet Forwarding

A VLL is established after VC information exchange and PW establishment. The following describes the packet forwarding process in Martini mode. (This figure shows two VLL networks: VPN1 and VPN2.)

Figure 11-7 Packet forwarding process in Martini mode

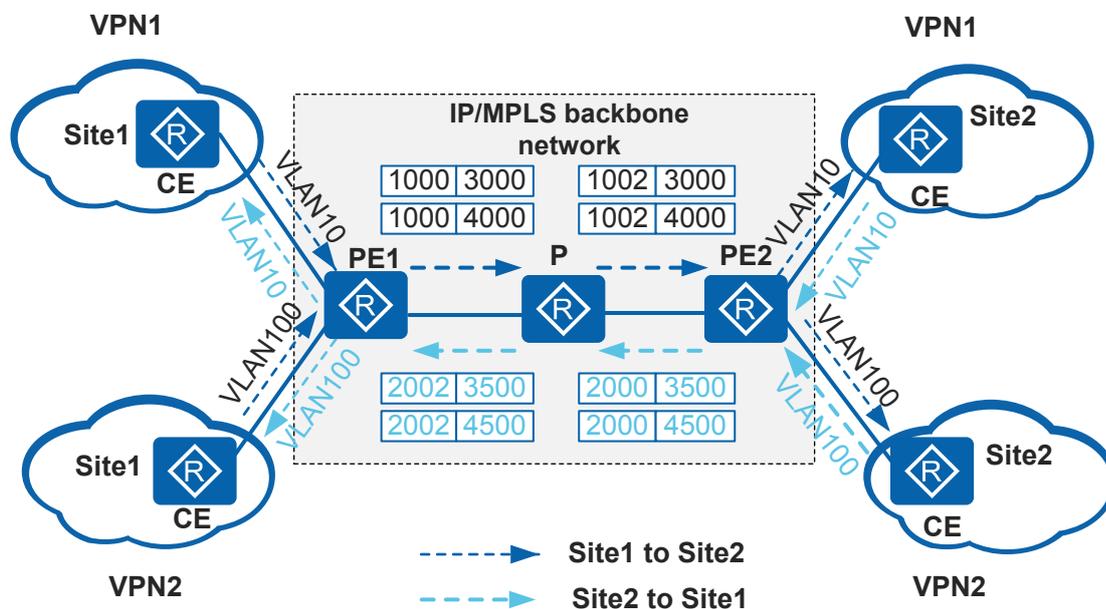


Figure 11-7 shows packet forwarding in two directions: from Site1 to Site2 and from Site2 to Site1.

- From Site1 to Site2

When a packet of VLAN 10 is sent from Site1 of VPN1 to PE1, PE1 adds a VC label 3000 and an outbound label 1000 of LSP1 to the packet. Then the packet enters LSP1 (the black dashed line). When a packet of VLAN 100 is sent from Site1 of VPN2 to PE1, PE1 adds a VC label 4000 and an outbound label 1000 of LSP1 to the packet. Then the packet enters LSP1 (the black dashed line).

When packets sent from Site1 reach PE2, PE2 removes the inbound label 1002 of LSP1 and selects the outbound interface according to the inner VC label. If the inner VC label is 3000, PE2 forwards the packets to the outbound interface connected to Site2 of VPN1. If the inner VC label is 4000, PE2 forwards the packets to the outbound interface connected to Site2 of VPN2. PE2 transmits VC labels 3000 and 4000 to PE1 using LDP when they set up the VCs.

- From Site2 to Site1

When a packet of VLAN 10 is sent from Site2 of VPN1 to PE2, PE2 adds a VC label 3500 and an outbound label 2000 of LSP2 to the packet. Then the packet enters LSP2 (the blue dashed line). When a packet of VLAN 100 is sent from Site2 of VPN2 to PE2, PE2 adds a VC label 4500 and an outbound label 2000 of LSP2 to the packet. Then the packet enters LSP2 (the blue dashed line).

When packets sent from Site2 reach PE1, PE1 removes the inbound label 2002 of LSP2 and selects the outbound interface according to the inner VC label. If the inner VC label is 3500, PE1 forwards the packets to the outbound interface connected to Site1 of VPN1. If the inner VC label is 4500, PE1 forwards the packets to the outbound interface connected to Site1 of VPN2. PE1 transmits VC labels 3500 and 4500 to PE2 using LDP when they set up the VCs.

In the transmission process, the outer labels specify the LSP for data transmission on the ISP network, and the inner VC labels identify data from different users. Data from multiple VCs can be transmitted over the same LSP.

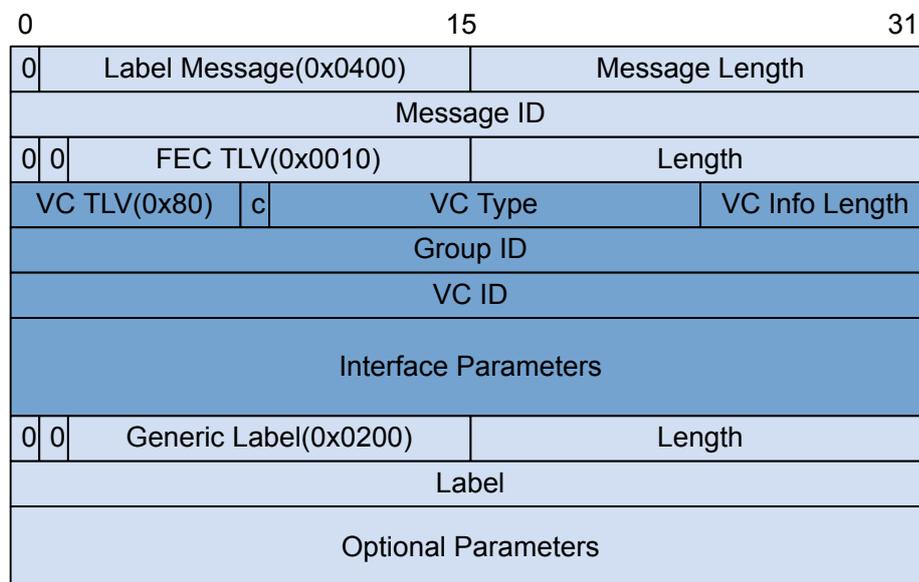
To deploy VLL in Martini mode, the ISP network must be able to automatically set up LSPs. Therefore, the ISP network must support MPLS forwarding and MPLS LDP. If the ISP network does not support LDP, GRE tunnels can be used on the ISP network.

VC Information Exchange

The Martini VLL extends the standard LDP by adding a VC FEC (type 128) to a Label Mapping message to carry VC information during PW establishment.

Figure 11-8 shows the format of a Label Mapping message. You can see the VC FEC in the Label Mapping message.

Figure 11-8 LDP Label Mapping message



VC FEC

VC FEC contains the inner VC label and interface parameters.

Table 11-4 Description of fields in the VC FEC (Type 128)

Field	Description	Bits	Remarks
VC TLV	Type, Length, and Value (TLV) of a VC	8	The value is 0x80, or 128 in decimal notation.
C	Control word	1	If the value is 1, control word is supported. If the value is 0, control word is not supported.
VC Type	Type of a VC	15	The value can be Ethernet or VLAN.
VC Info Length	Length of VC information	8	The value is the total length of the VC ID and the Interface Parameters field.
Group ID	ID of a VC group	32	Multiple VCs can constitute a VC group and information about all VCs in the group can be deleted together.
VC ID	ID of a VC	32	-
Interface Parameters	Interface parameters	Variable, smaller than the value of VC Info Length	The frequently used interface parameters include MTU and interface description.

11.2.2.3 VLL in SVC Mode

Introduction

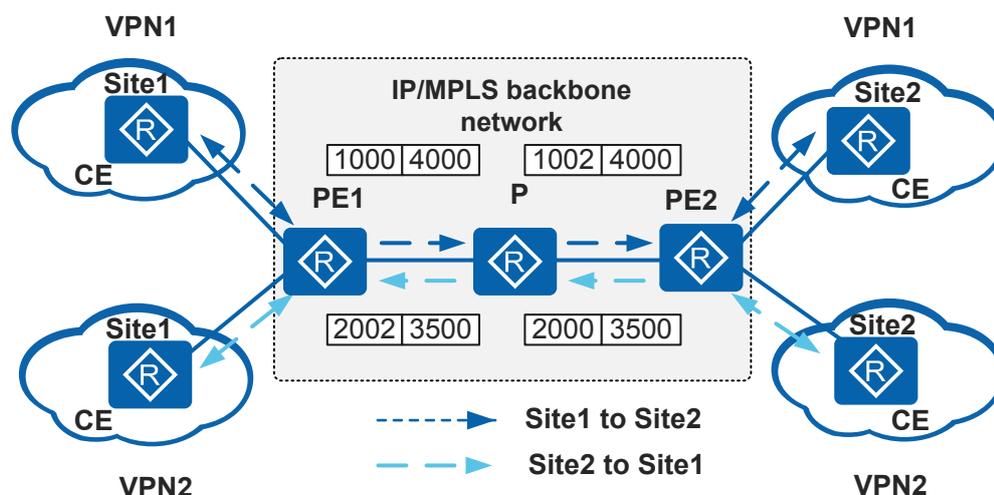
The static virtual circuit (SVC) mode is a simplified Martini mode. Unlike the Martini mode that uses LDP to exchange VC labels, the SVC mode uses VC labels that are manually configured on PE devices.

An SVC VLL uses static VC labels and does not need VC label mapping. Therefore, LDP is not required for transmitting VC labels.

Topology

The SVC mode sets up a public tunnel (outer label) in the same way as the Martini mode. The inner label is manually configured during VC setup, and PE devices do not need to exchange VC labels using any signaling protocol. The SVC mode does not support local connections. The network topology and packet exchange process in SVC mode are the same as those in Martini mode.

Figure 11-9 Packet exchange in SVC mode



As shown in [Figure 11-9](#), an SVC VLL is established between two sites of VPN1. On PE1, the label for sent packets is set to 4000 and the label for received packets is set to 3500. On PE2, the label for sent packets is set to 3500 and the label for received packets is set to 4000. When a packet is sent from Site1 to Site2 of VPN1, PE1 adds the inner VC label 4000 to the packet. After PE2 receives the packet with the inner VC label 4000, it sends the packet to the CE device through the AC mapping the inner VC label.

11.2.2.4 Comparison of VLL Modes

[Table 11-5](#) compares three VLL modes.

Table 11-5 Comparison of VLL modes

Implementation	VC Label Distribution Mode	PW Signaling Protocol	Characteristics
CCC	Manually specified	None	This mode establishes one-layer static LSP tunnels for VC information transmission.
Martini	Randomly distributed by the system	LDP	This mode establishes two layers of tunnels. The outer tunnel is a public network tunnel used to transparently transmit data, and the inner tunnels are identified by VC labels distributed by the system.

Implementation	VC Label Distribution Mode	PW Signaling Protocol	Characteristics
SVC	Manually specified	None	This mode establishes two layers of tunnels. The outer tunnel is a public network tunnel used to transparently transmit data, and the inner tunnels are identified by VC labels that are manually specified.

11.2.3 Inter-AS VLL

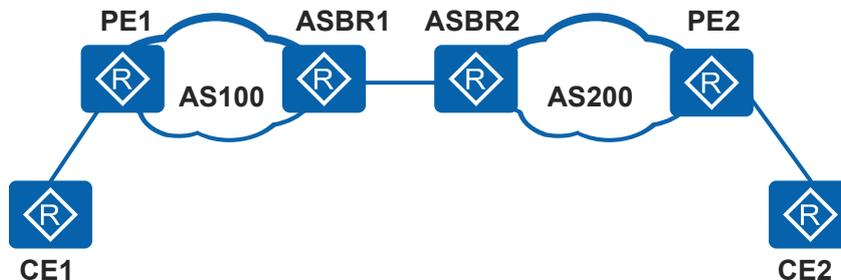
The MPLS VPN solution is widely used, serving an increasing number of users in a large number of applications. As additional sites are developed in an enterprise, sites in different geographical locations are often connected to different ISP networks. Consider, for example, the inter-AS issue facing operators who manage different metropolitan area networks (MANs) or backbone networks that span different autonomous systems (AS). This scenario requires a different interworking model than the MPLS VPN architecture. This interworking model, called the inter-AS VLL, features a VPN that spans multiple ASs. In inter-AS networking, the devices located at the edge of ASs are AS boundary routers (ASBRs).

Inter-AS VLL implementation depends on the VLL mode.

- The CCC mode uses a single label. Therefore, the inter-AS VLL can be set up when static LSPs are set up between ASBRs.
- The SVC and Martini modes can implement inter-AS VLL Option A (VRF-to-VRF).

In inter-AS Option A, a sub-interface must be reserved for each inter-AS VC on an ASBR. The ASBRs do not require MPLS or other additional inter-AS configuration. With Option A, ASBRs of the two ASs are directly connected and function as PE devices in their respective ASs. The two ASBRs regard each other as CE devices. [Figure 11-10](#) shows the inter-AS option A networking.

Figure 11-10 Inter-AS Option A



For details about inter-AS VPN, see [8.2.4 Inter-AS VPN](#) in the *BGP/MPLS IP VPN Configuration*.

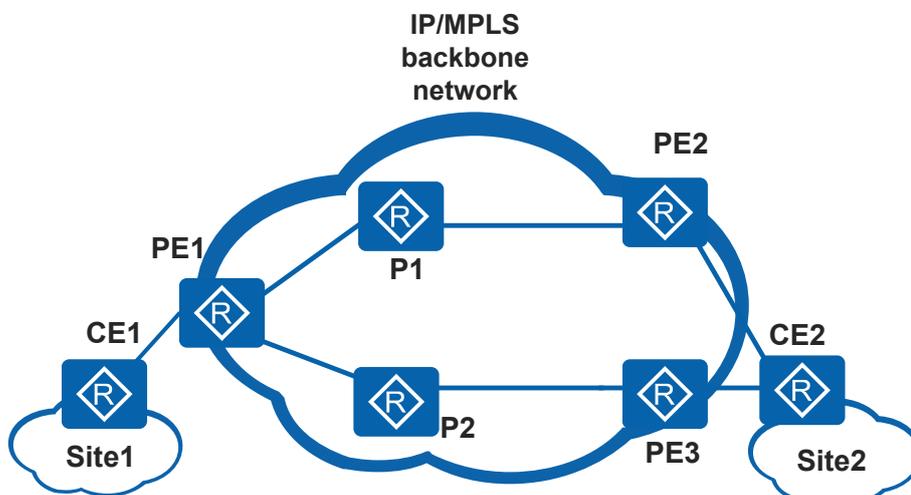
11.2.4 VLL FRR

Widespread adoption of MPLS L2VPN technologies has raised high reliability requirements for L2VPNs, especially for L2VPNs that carry real-time services such as VoIP and IPTV.

Virtual Lease Line Fast Reroute (VLL FRR) uses redundant networking to improve MPLS L2VPN reliability. When a PW or PE device fails, VLL FRR fast switches traffic to a backup link. This mechanism implements end-to-end fault detection on PWs and provides PW protection, greatly improving link-layer reliability for MPLS L2VPNs.

Figure 11-11 shows the application of VLL FRR in asymmetrical CE deployment: the CE device is connected to one PE device at one end while the CE device is dual-homed to two PE devices at the other end.

Figure 11-11 Asymmetrical CE deployment



In asymmetrical networking, PE1 or CE2 is the failover point and the last step in the fault notification process. When the primary link fails, PE1 in the single-homed site detects the fault, triggers traffic switching, and does not send fault notification to CE1. When the primary link in the dual-homed site fails, CE2 receives the fault notification from the PE device on the primary link and switches traffic to the backup link.

VLL FRR Implementation

- **Fault Detection**

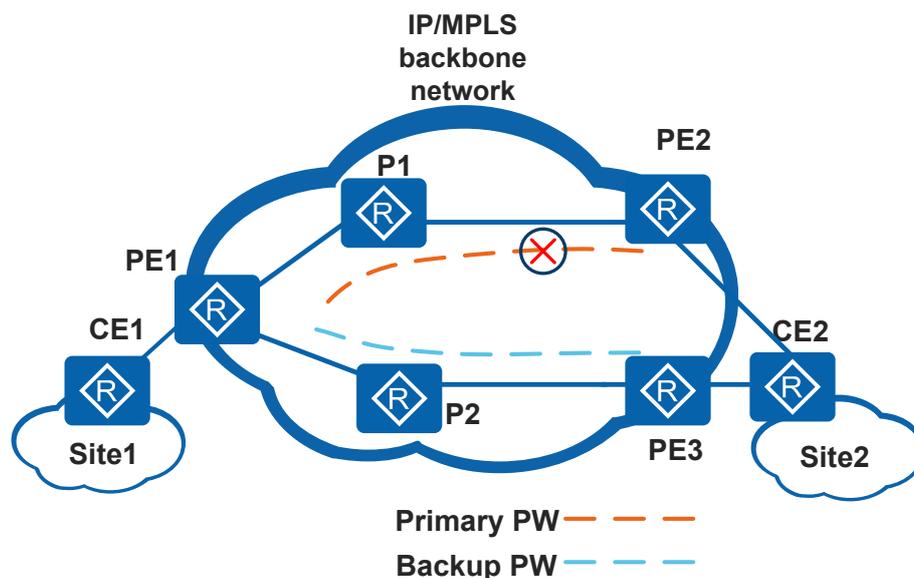
When a fault occurs on the VLL network, devices can detect link failures through route convergence. However, the detection speed is slow and cannot meet the requirements of delay-sensitive services, such as VoIP services. To improve the fault detection speed, you can deploy Bidirectional Forwarding Detection (BFD) on the PE device to rapidly detect PW failures. BFD has a low cost and can implement millisecond-level fault detection.

- **Switching Between Primary and Backup PWs**

You can set up a backup PW on the VLL network. When the primary PW is working properly, the backup PW does not transmit data. When the primary PW is faulty, data is switched to the backup PW. The backup PW remains in Up state to ensure fast traffic switching when the primary PW fails, preventing traffic loss.

As shown in [Figure 11-12](#), PE1-P1-PE2 is the primary link and PE1-P2-PE3 is the backup link. When the system detects that the primary PW or PE2 is faulty, PE1 performs switching between primary and backup PWs to import traffic to the backup PW, ensuring traffic transmission from CE1 to CE2.

Figure 11-12 Switching between primary and backup PWs



- **Fast Fault Notification**

As shown in [Figure 11-12](#), switching between primary and backup PWs ensures normal traffic transmission from CE1 to CE2, while switching of traffic from CE2 to CE1 is ensured by fast fault notification.

The OAM mapping between a PW and an AC interface can be created. In this manner, when a PW or a PE is faulty, a CE can take measures in time to fast switch traffic to a secondary path. The OAM messages are transparently transmitted on a PW. This enables the PW with the end-to-end fault detection function.

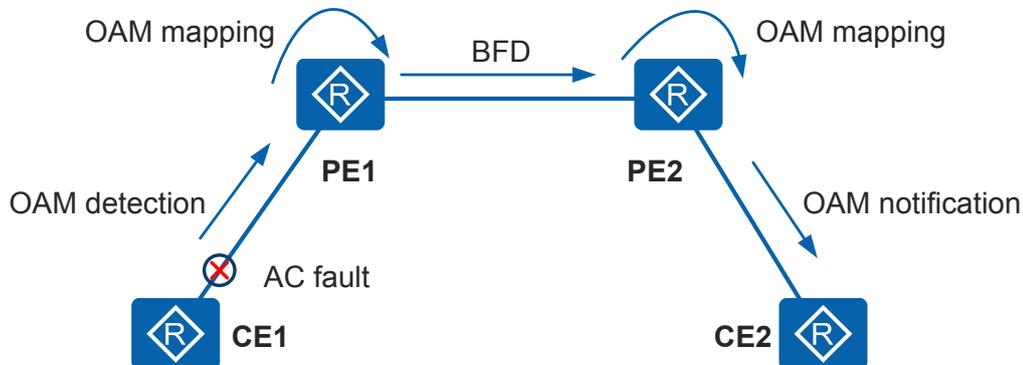
- AC Fault Detection and Notification Mechanism

As shown in [Figure 11-13](#), in the scenario where a fault occurs on an AC between CE1 and PE1, the fault detection and notification mechanism works as follows:

- i. AC OAM on PE1 detects an AC fault.
- ii. According to the OAM mapping on the PE1, the PW status corresponding to the AC is updated.
- iii. BFD transparently sends an OAM fault message to PE2.

- iv. When PE2 receives the BFD fault message, if a secondary PW is set up on the remote PE, the traffic switchover is performed. Otherwise, OAM mapping is performed and then the faulty AC is notified to CE2.

Figure 11-13 AC fault detection and notification mechanism

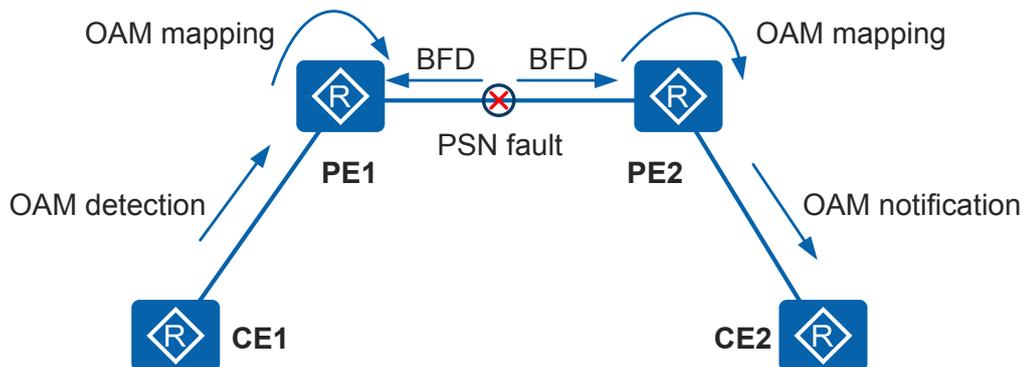


- PSN Fault Detection and Notification Mechanism

As shown in **Figure 11-14**, when a fault occurs in a packet switched network (PSN), the fault detection and notification mechanism works as follows:

- i. The BFD session on the PE detects a fault in the PSN.
- ii. The PE obtains the local AC according to the OAM mapping.
- iii. If a secondary PW is set up, the traffic switchover is performed; otherwise, the OAM mapping is performed and the corresponding AC is mapped and the fault is notified to the local CE.

Figure 11-14 PSN fault detection and notification mechanism



● **PW Switchback Policy**

In the networking of CEs asymmetrically accessing PEs, when PE1 is notified of fault removal on the primary PW, PE1 works based on the PW switchback policy.

The PW switchback policies are as follows:

- No switchback: Traffic is not switched back to the primary PW.
- Immediate switchback: Traffic is immediately switched back to the primary PW.

- Delayed switchback: Traffic is switched back to the primary PW after a delay period.

After the switchback, the PE immediately notifies the peer PE on the secondary PW of the fault. In addition, after a delay period or immediately the PE notifies the peer PE on the secondary PW of fault removal, which prevents packet loss due to transmission delay between PEs.

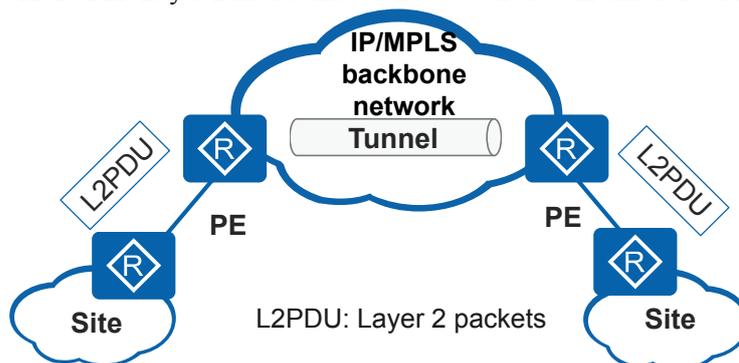
11.3 Application Scenarios for VLL

This section describes application scenarios for application scenarios for VLL.

11.3.1 Point-to-Point Layer 2 Connection Between Sites in Different Cities

VLL technology enables Layer 2 protocol data units (L2PDUs) to traverse a carrier network without being changed. Sites in different cities can use this technology to set up P2P Layer 2 interconnection and communicate as if they were on a LAN.

Figure 11-15 P2P Layer 2 interconnection between sites in different cities



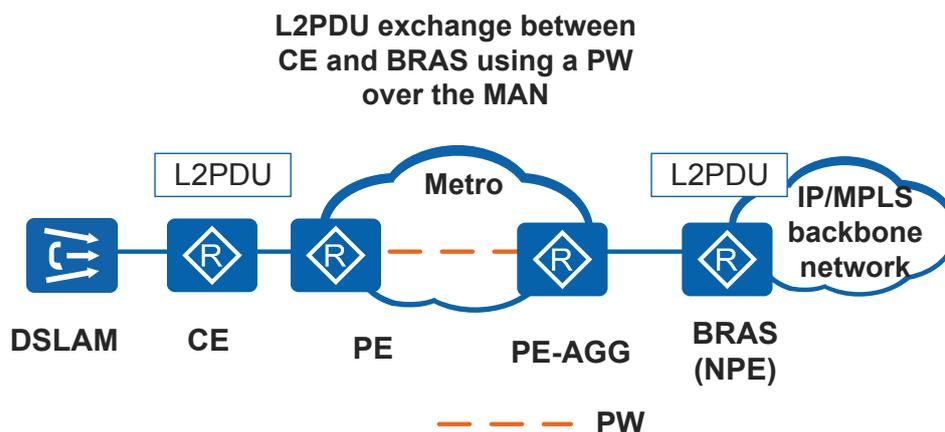
11.3.2 Multi-service Transparent Transmission over PWs on a MAN

In many carrier-class services, carriers use digital subscriber line access multiplexers (DSLAMs) or Ethernet switches to provide access lines such as asymmetric digital subscriber line (ADSL), very-high-data-rate digital subscriber line (VDSL), or Ethernet links and need to control various services. Service control includes Point-to-Point Protocol over Ethernet (PPPoE) access request termination, IP address assignment, user authentication, authorization, and accounting. Terminal access gateway are responsible for service control. As large-capacity and high-performance terminal access gateway (Huawei ME60 for example) are developed, terminal access gateway are moving gradually upward to the metropolitan area network (MAN) egress. This means that DSLAMs are connected to the MAN, and integrated terminal access gateway are deployed at the MAN egress to control services.

Terminal access gateway usually exchange information with users through Layer 2 links. For example, a terminal access gateway obtains user names and passwords through PPPoE

sessions. However, a MAN is a Layer 3 IP/MPLS network. If Layer 2 user information is terminated on the UPE devices connected to DSLAMs, terminal access gateway cannot obtain required information through Layer 2 connections, and therefore they cannot control services. To solve this problem, VLLs can be set up on the MAN to transmit Layer 2 packets exchanged between BRASs and users over PWs.

Figure 11-16 Multi-service transparent transmission over PWs on a MAN



11.4 Summary of VLL Configuration Tasks

VLLs are classified into three modes: CCC, Martini, SVC. If a network spans multiple ASs, you need to configure inter-AS VLL. You can also configure VLL FRR to provide link-layer reliability for VLL networks.

Table 11-6 VLL configuration tasks

Scenario	Description	Task
Configure basic VLL functions	<p>Basic VLL functions are configured on MPLS VLL backbone networks that do not span multiple ASs and do not require VLL FRR. With basic VLL functions configured, sites can set up P2P Layer 2 connections for communication.</p> <p>You need to select a proper VLL mode based on the scale of the current network and network expansion requirements.</p> <ul style="list-style-type: none"> ● The CCC VLL applies to small-scale enterprises with a few sites and simple topologies. ● The Martini VLL applies to large-scale enterprises or LANs of small-scale carriers. VLL in Martini mode is easy to configure and extend and mature in fault locating; therefore, this mode is widely used. ● The SVC VLL is a simplified Martini mode, in which inner labels are manually specified. The SVC mode facilitates planning of label resources on VLL networks, but it increases configuration workload and is not easy to extend. 	<ul style="list-style-type: none"> ● 11.7.1 Configuring the CCC VLL ● 11.7.2 Configuring the Martini VLL ● 11.7.3 Configuring the SVC VLL
Configure inter-AS VLL	Configure inter-AS VLL if the VLL backbone network spans multiple ASs.	11.7.4 Configuring Inter-AS VLL
Configure VLL FRR	Configure VLL FRR to ensure link-layer reliability.	11.7.5 Configuring VLL FRR
Configure and apply a tunnel policy	Tunnel policies are required when VLL services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources.	11.7.7 Configuring and Applying a Tunnel Policy

11.5 Licensing Requirements and Limitations for VLL

This section describes licensing requirements and limitations for VLL.

Involved Network Elements

None

License Requirements

For L2VPN-capable devices, their licensing requirements for the L2VPN function are as follows:

- AR1200-S series: L2VPN is a basic feature of the device and is not under license control.
- AR2200-S&AR3200-S series: By default, L2VPN function is disabled on a new device. To use the L2VPN function, apply for and purchase the following license from the Huawei local office.
 - AR2200-S series: AR2200 value-added service package for data services
 - AR3200-S series: AR3200 value-added service package for data services

Feature Limitations

- VLL cannot be configured on the VLANIF 1.
- The VLANIF interface configured with VLL can only correspond to one member interface. This limitation does not apply to AR1220E-S.

When configuring VLL on the devices, pay attention to the following points:

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S series do not support VLL.

11.6 Default Settings for VLL

This section provides the default settings for VLL.

Table 11-7 Default settings for VLL

Parameter	Default Setting
MPLS L2VPN	Disabled
MTU of a PW	1500
Revertive switching policy for VLL FRR	Delayed revertive switching

11.7 Configuring VLL

This section describes how to configure VLL functions in details.

11.7.1 Configuring the CCC VLL

The CCC VLL does not require signaling negotiation or exchange of control packets; therefore, it consumes few resources and is easy to configure. However, it requires manual configuration and makes network maintenance and expansion difficult. This mode applies to small-scale MPLS networks with simple topologies.

Pre-configuration Tasks

- Enabling basic MPLS capabilities on the PE and P devices of the MPLS backbone network before configuring a local CCC connection

Procedure

- **Configure a local CCC connection.**

To create a local CCC connection, you only need to configure the inbound and outbound interfaces of the CCC connection on the PE device. Perform the following operations on the PE device:

- a. Run **system-view**
The system view is displayed.
- b. Run **mpls l2vpn**
MPLS L2VPN is enabled and the MPLS L2VPN view is displayed.
Enable MPLS L2VPN on the PE device before you configure a VLL.
- c. Run **quit**
Return to the system view.
- d. Run **ccc ccc-connection-name interface interface-type1 interface-number1 [raw | tagged] out-interface interface-type2 interface-number2 [raw | tagged]**
A local CCC connection is created.
The local CCC connection is bidirectional; therefore, only one connection is required.

Verifying the Configuration

After you complete configuring the CCC VLL, run the following commands to check information about the CCC connection and interfaces used by the connection.

- Run the **display vll ccc [ccc-name | type local]** command to check information about a CCC connection.
- Run the **display l2vpn ccc-interface vc-type ccc [down | up]** command to check information about interfaces used by the CCC connection.

11.7.2 Configuring the Martini VLL

The Martini VLL applies to LANs of large-scale enterprises or small-scale carriers.

Pre-configuration Tasks

Before configuring the Martini VLL, you need to complete the following tasks:

- Configuring static routes or an Interior Gateway Protocol (IGP) protocol for the PE and P devices on the MPLS backbone network to implement IP connectivity.
- Enabling basic MPLS capabilities on the PE and P devices and enabling MPLS LDP on PEs.
You need to set up a remote LDP session between the PEs if they are not directly connected.
- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs.
You also need to configure tunnel policies when VLL services need to be transmitted over TE tunnels or when VLL services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [Configuring and Applying a Tunnel Policy](#).

Procedure

Perform the following operations on PEs at both ends of the VC.

1. Run **system-view**
The system view is displayed.
2. Run **mpls l2vpn**
MPLS L2VPN is enabled and the MPLS L2VPN view is displayed.
3. Run **quit**
Return to the system view.
4. Run **interface interface-type interface-number**
The AC interface view is displayed.
The Martini VLL can use the following interfaces as AC interfaces: GE interfaces, GE sub-interfaces, Ethernet interfaces, Ethernet sub-interfaces, Eth-Trunk interfaces, Eth-Trunk sub-interfaces, and VLANIF interfaces.
The sub-interfaces can be dot1q sub-interfaces or QinQ sub-interfaces.
5. Run **mpls l2vc { ip-address | pw-template pw-template-name } * vc-id [tunnel-policy policy-name | [control-word | no-control-word] | [raw | tagged] | mtu mtu-value] ***
A VLL connection in Martini mode is created. VCs of the same encapsulation mode on a PE must have a unique VC ID.
Configure a PW template before you can use it. Refer to [Creating a PW Template and Setting Attributes for the PW Template](#) for details on the PW template configuration.
6. (Optional) Run **bpdu transmit enable**
The interface connected to a VLL is enabled to transparently transmit BPDU packets.
7. (Optional) Run **mpls l2vpn frag**
Fragmentation of L2VPN IPv4 packets is enabled.
8. (Optional) Run **mpls l2vpn service-name service-name**
A name is configured for the L2VPN service, so you can maintain the L2VPN service by clicking the name directly on the NMS graphical user interface (GUI).

Verifying the Configuration

After completing the Martini VLL configuration, you can view Martini VLL connection information on the PEs.

- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number*] command to check Martini VLL connection information on the local PE.
- Run the **display mpls l2vc remote-info** [*vc-id*] command to check peer Martini VLL connection information on the local PE.
- Run the **display mpls l2vc brief** command to check Martini VLL connection brief information on the local PE.

11.7.3 Configuring the SVC VLL

The SVC VLL is a simplified Martini mode, in which inner labels are manually specified. The SVC mode facilitates planning of label resources on VLL networks, but it increases configuration workload and is not easy to extend.

Pre-configuration Tasks

Before configuring the SVC VLL, you need to complete the following tasks:

- Configuring static routes or an IGP protocol for the PE and P devices on the MPLS backbone network to implement IP connectivity.
- Enabling the MPLS for PEs and Ps.
- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs.

You also need to configure tunnel policies when VLL services need to be transmitted over TE tunnels or when VLL services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [Configuring and Applying a Tunnel Policy](#).

Procedure

Perform the following operations on the PE devices at both ends of a VC:

1. Run **system-view**
The system view is displayed.
2. Run **mpls l2vpn**
MPLS L2VPN is enabled and the MPLS L2VPN view is displayed.
3. Run **quit**
Return to the system view.
4. Run **interface** *interface-type interface-number*
The AC interface view is displayed.
The SVC VLL can use the following interfaces as AC interfaces: GE interfaces, GE sub-interfaces, Ethernet interfaces, Ethernet sub-interfaces, Eth-Trunk interfaces, Eth-Trunk sub-interfaces, and VLANIF interfaces.
The sub-interfaces can be dot1q sub-interfaces or QinQ sub-interfaces.
5. Run **mpls static-l2vc** { { **destination** *ip-address* | **pw-template** *pw-template-name* *vc-id* } * | **destination** *ip-address* [*vc-id*] } **transmit-vpn-label** *transmit-label-value* **receive-vpn-label** *receive-label-value* [**tunnel-policy** *tnl-policy-name* | [**control-word** | **no-control-word**]] [**raw** | **tagged**]] *

The SVC VLL is created.

Configure a PW template before you can use it. Refer to [Creating a PW Template and Setting Attributes for the PW Template](#) for details on the PW template configuration.

6. (Optional) Run **bpdu transmit enable**
The interface connected to a VLL is enabled to transparently transmit BPDU packets.
7. (Optional) Run **mpls l2vpn service-name service-name**
A name is configured for the L2VPN service, so you can maintain the L2VPN service by clicking the name directly on the NMS GUI.
8. (Optional) Run **mpls l2vpn frag**
Fragmentation of L2VPN IPv4 packets is enabled.

Verifying the Configuration

After you complete configuring the SVC VLL, run the following commands to check information about the SVC connection and interfaces used by the SVC connection.

- Run the **display mpls static-l2vc [interface interface-type interface-number]** command to check the SVC L2VPN connection information on the PE.
- Run the **display mpls static-l2vc brief** command to check the SVC L2VPN connection brief information on the PE.
- Run the **display l2vpn ccc-interface vc-type static-vc [down | up]** command to check the interface information of the SVC connections in Up/Down state.

11.7.4 Configuring Inter-AS VLL

If VLLs need to be set up over an MPLS backbone that spans multiple ASs, the inter-AS VLL must be configured.

Context

The configuration of inter-AS varies with the VLL implementation mode.

- The CCC mode uses a single label. Therefore, the inter-AS VLL can be set up when static LSPs are set up between ASBRs.
- The Martini and SVC modes can implement inter-AS VLL Option A (VRF-to-VRF).

To configure inter-AS Option A, you must specify an interface (a sub-interface, a physical interface, or a bundled logical interface) for each inter-AS VC on ASBRs.

Procedure

To implement inter-AS Option A, complete basic VLL configurations in each AS and configure the ASBR-PE devices as the CE devices of each other.

- To configure inter-AS CCC VLL, see [11.7.1 Configuring the CCC VLL](#).
- To configure inter-AS Martini VLL using Option A, see [11.7.2 Configuring the Martini VLL](#).
- To configure inter-AS SVC VLL using Option A, create the SVC for each AS. For the detailed configuration, see [11.7.3 Configuring the SVC VLL](#).

NOTE

You do not need to perform any additional configuration for inter-AS implementation on ASBRs and do not need to configure IP addresses for the directly connected interfaces between ASBRs.

Verifying the Configuration

- Run the **display vll ccc** [*ccc-name* | **type local**] command to check information about a CCC VLL.
- Run the **display l2vpn ccc-interface vc-type ccc** [**down** | **up**] command to check information about interfaces used by the CCC VLL.
- Run the **display mpls l2vc** [*vc-id* | **interface interface-type interface-number**] command to check PW information about the local end of a Martini VLL.
- Run the **display mpls l2vc remote-info** [*vc-id*] command on the PE device to check PW information about the remote end of the Martini VLL.
- Run the **display mpls static-l2vc** [**interface interface-type interface-number**] command to check information about an SVC VLL.
- Run the **display l2vpn ccc-interface vc-type static-vc** [**down** | **up**] command to view information about VC interfaces in the Up/Down state on the SVC VLL.

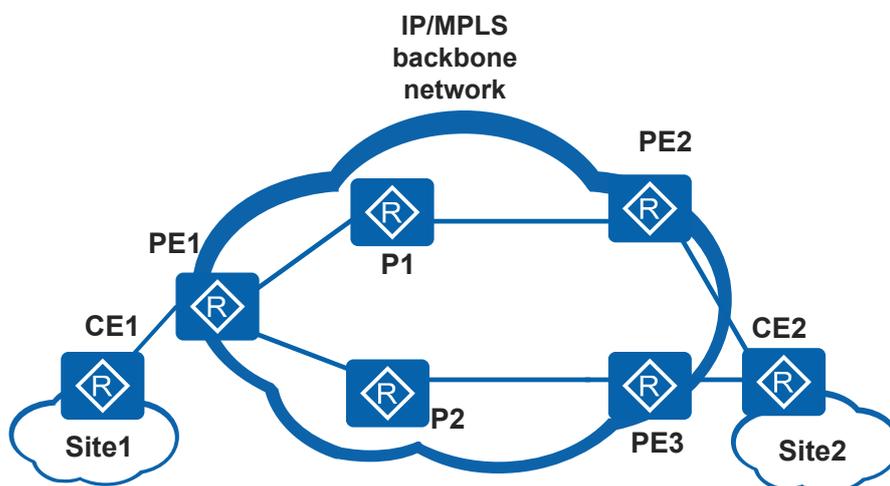
11.7.5 Configuring VLL FRR

You can configure VLL FRR to provide link-layer protection to improve reliability for VLL networks.

Context

Only the Martini VLL modes support VLL FRR. VLL FRR is mainly used in a CE asymmetrical networking, as shown in [Figure 11-17](#).

Figure 11-17 CE asymmetrical networking



In the networking:

- The primary and secondary IP addresses need to be configured on the interface on the CE connected to the PE through a single link. When the primary link is available, the CE in the single-homed site uses the primary IP address to communicate with the remote

- CE. When a fault occurs on the primary link, this CE communicates with the remote CE by using the secondary IP address.
- The secondary PW cannot transmit data when the primary and secondary paths work normally. On the CE in the dual-homed site, if the interface of the secondary PW borrows the IP address of the interface of the primary PW, the following situations occur:
 - The policy of none revertive switching cannot be configured.
 - The local CE has two equal-cost and direct routes to the remote CE. The destination addresses and next hops of the two routes are the same. Actually, the route that passes through the secondary PW is invalid.
 - If CEs exchange routing information by using routing protocols, you need to modify the cost or metric of the AC interface of the secondary path to be greater than that of the AC interface of the primary path. The local CE cannot communicate with the peer CE, but can communicate with other user devices.
 - If CEs use static routes to exchange routing information, you need to modify the preference of the backup route to be lower than that of the primary route (the greater the value, the lower the preference) by using the **ip route-static dest-ip-address mask out-interface preference preference-value** command.

Pre-configuration Tasks

Before configuring VPN FRR, complete the following tasks:

- Configuring basic VLL functions
- Configuring CEs to exchange routing information by using routing protocols or static routes
- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs

You also need to configure tunnel policies when VLL services need to be transmitted over TE tunnels or when VLL services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [Configuring and Applying a Tunnel Policy](#).

Configuration Procedure

Some of the following operations are optional. Perform the operations in the following sequence.

11.7.5.1 Configuring Primary and Secondary PWs

Context

You can configure primary and secondary PWs for PW backup on a network. VLL FRR uses redundant networking to improve L2VPN reliability. When a PW or PE device fails, VLL FRR fast switches traffic to a backup link. VLL FRR is only supported by the Martini modes.

Perform the following configurations on the PEs.

Procedure

- Configure primary and secondary PWs for the Martini VLL.
 - a. Run **system-view**

The system view is displayed.

b. Run **mpls l2vpn**

The MPLS L2VPN view is displayed.

c. Run **quit**

Return to the system view.

d. Run **interface** *interface-type interface-number*

The AC interface view is displayed.

The PE devices to which CE device is dual-homed must use a main interface as the AC interface.

e. Run **mpls l2vc** { *ip-address* | **pw-template** *pw-template-name* } * *vc-id* [**tunnel-policy** *policy-name* | [**control-word** | **no-control-word**] | [**raw** | **tagged**] | **mtu** *mtu-value*] *

The primary PW is configured.

f. (Optional) Run **mpls l2vc** { *ip-address* | **pw-template** *pw-template-name* } * *vc-id* [**tunnel-policy** *policy-name* | [**control-word** | **no-control-word**] | [**raw** | **tagged**] | **mtu** *mtu-value*] * **secondary**

The secondary PW is configured.

If a CE device is single-homed to a PE device, configure primary and secondary PWs. If a CE device is dual-homed to two PE devices, configure the primary PW on each PE device.

Primary and secondary PWs must have different VC IDs.

----End

11.7.5.2 (Optional) Configuring Fast Fault Notification - OAM Mapping

Context

OAM mapping expedites the fault detection and notification on the AC end. OAM mapping can be configured on various types of links. To configure OAM mapping on Ethernet links, the PE and CE devices must support the Ethernet OAM function.

Choose either of the following procedures to configure OAM mapping according to the AC types.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The view of the AC interface is displayed.

Step 3 Run **mpls l2vpn oam-mapping 3ah**

The fault mapping between the AC and the PW is enabled.

 NOTE

- The PW need be configured in homogeneous interworking mode when the AC is an Ethernet. Otherwise, the use device may learn a wrong outbound interface according to ARP.
- Before running the **mpls l2vpn oam-mapping 3ah** command, you need configure Ethernet OAM on the AC link. For details, refer to "EFM Configuration" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Reliability*.
- If the **mpls l2vpn oam-mapping** command is configured, run the **display mpls l2vc interface** command to check the VC status. In the command output, "Local AC OAM State" indicates the status of the AC link; if the **mpls l2vpn oam-mapping** command is not configured, run the **display mpls l2vc interface** command to check the VC status. In the command output, "Local AC OAM State" is always Up, and has no relationship with the AC link status.

---End

11.7.5.3 (Optional) Configuring BFD for PW

Context

BFD for PW is recommended because it speeds up fault detection.

Procedure

For details, see the following topics.

- [12.8.5 Configuring Static BFD for PWs](#)

 NOTE

- BFD for PW on both PEs at the two ends must be configured or deleted simultaneously. Otherwise, the statuses of PWs on the PEs are inconsistent.
- To monitor statuses of tunnels that carry PWs, configure BFD for tunnel. For detailed configurations, see "MPLS LDP Configuration" and "MPLS TE Configuration" in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Manual MPLS*.

11.7.5.4 (Optional) Configuring a Revertive Switchover Policy

Context

Revertive switching policies are classified into the following types:

- Immediate revertive switchover: When the primary PW recovers from a fault, the local PE switches traffic back to the primary PW immediately and notifies the peer PE on the secondary PW of the fault. In FRR mode, the local PE notifies the peer PE on the secondary PW of the recovery after a delay of *resume-time*. In PW redundancy master/slave mode, the parameter *resume-time* is not supported.
This revertive switchover applies to scenarios in which users hope traffic to be restored as soon as possible.
- Delayed revertive switchover: When the primary PW recovers from a fault, traffic is switched back to the primary PW after a period specified by *delay-time*. After traffic is switched back, the local device immediately notifies the peer device on the secondary PW of the fault. If *resume-time* is configured in FRR mode, the local device notifies the peer device on the secondary PW of the recovery after a delay of *resume-time*.

On a large-scale network, packet loss caused by incomplete route convergence may occur during the switchover. To prevent this problem, configure traffic to be switched back after a delay.

- None revertive switchover: When the primary PW recovers from a fault, traffic is not switched back to the primary PW until the secondary PW becomes faulty.

If you do not want traffic to be frequently switched between the primary and secondary PWs, you can use the non-revertive switchover.

By default, the delayed revertive switchover is performed.

A revertive switchover policy is configured on a PE. In asymmetric networking, if the active PW is faulty, the PE to which a CE is connected through a single link switches traffic. When the active PW is restored, configure a revertive switchover policy on this PE. The PE then processes traffic based on the configured revertive switchover policy.

Perform the following operations on the PE (where traffic is switched) to which the CE is connected through a single link.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The AC interface view is displayed.

Step 3 Run **mpls l2vpn reroute** { { **delay** *delay-time* | **immediately** } [**resume** *resume-time*] | **never** }

The revertive switchover policy is configured.

For an asymmetric networking with ACs of the Ethernet type, if the Ethernet OAM function is configured on the PE interface connected to a CE, and a revertive switching policy is also configured, do not set *resume-time* to 0 seconds. Set *resume-time* to 1 second or longer.

NOTE

On the network where CEs are asymmetrically connected to PEs, the secondary PW cannot transmit data when the primary and secondary paths work normally. On the CE in the dual-homed site, if the interface of the secondary PW borrows the IP address of the interface of the primary PW, you cannot configure revertive switchover.

----End

11.7.5.5 Verifying the VLL FRR Configuration

Context

After configuring VLL FRR, you can check information about local and remote PWs, BFD sessions, and L2VPN forwarding. You can also run the **manual-set pw-ac-fault** command to set faults on a PW to verify whether the switchover between the primary and secondary PWs is normal.

Procedure

- Run the **manual-set pw-ac-fault** command on the primary PW to set faults on it to verify whether the switchover between the primary and secondary PWs is normal.
- Run the **display mpls l2vc** [*vc-id* | **interface interface-type interface-number**] command to check information about the local end of the Martini VC.
- Run the **display mpls l2vc remote-info** [*vc-id*] command to check information about the remote end of the Martini VC.
- Run the **display bfd session pw interface interface-type interface-number** [**secondary**] [**verbose**] command to check information about the BFD session.
- Run the **display mpls l2vpn forwarding-info** [*vc-label*] **interface interface-type interface-number** command to check forwarding information about the L2VPN.
- Run the **display mpls l2vc oam-mapping** [**interface interface-type interface-number**] command to check OAM mapping between ACs and PWs.

----End

11.7.6 Configuring the Access of VLL to L3VPN

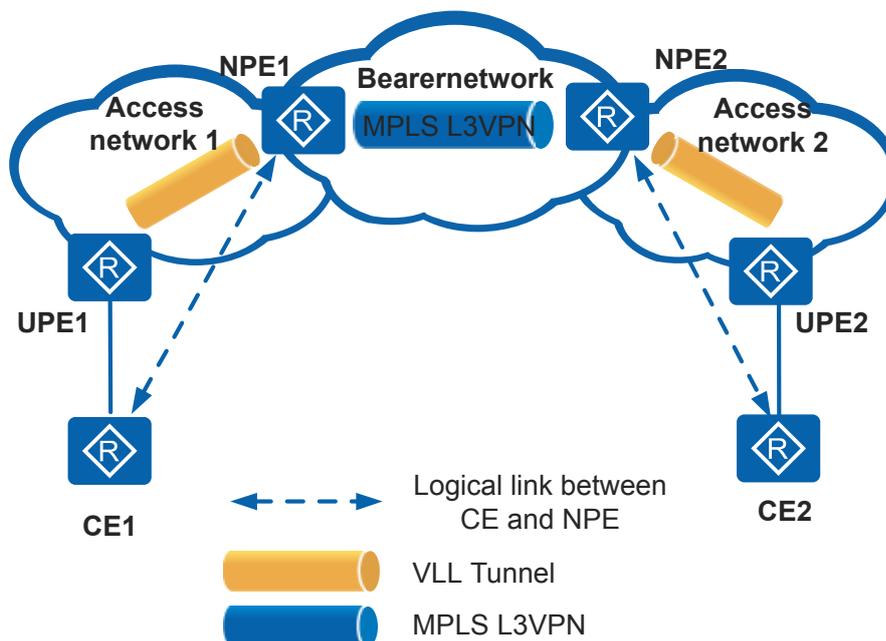
This section describes how to configure VLLs to access L3VPN. To achieve.

11.7.6.1 Before You Start

Applicable Environment

As shown in [Figure 11-18](#), to allow a user to access the public network or an MPLS L3VPN bearer network through an MPLS L2VPN access network, the carrier can deploy a VLL to connect the user to the public network or the MPLS L3VPN.

Figure 11-18 Networking diagram of connecting a VLL to an L3VPN



Pre-configuration Tasks

Before configuring a VLL to access an L3VPN, complete the following tasks:

- Connecting interfaces and configuring their physical parameters so as to make their physical layer Up
- Enabling an IGP on the MPLS access network to implement IP connectivity
- Enabling MPLS L2VPN on UPEs and NPEs
- Creating L2VPN tunnels between UPEs and NPEs
- Creating LDP sessions between NPEs and UPEs
- Creating remote LDP sessions if NPEs and UPEs are not connected directly
- Enabling an IGP on the MPLS bearer network to implement IP connectivity
- Configuring basic functions of L3VPN on NPEs

Data Preparation

To configure a VLL to access an L3VPN, you need the following data.

No.	Data
1	VE interface number
2	VE-Group number
3	Martini VLL: Destination IP address of the L2VC, VC ID, and VC Type

11.7.6.2 Creating an L2VE Interface

Context

Perform the following steps on NPEs. This part describes how to configure an L2VE interface that terminates the L2VPN, and how to bind the L2VE interface to the relevant VE group.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface virtual-ethernet** *interface-number*

A VE interface is created and the VE interface view is displayed.

Step 3 Run **ve-group** *ve-group-id* **l2-terminate**

The VE interface is set to an L2VE interface that terminates VLL, and the interface is bound to a VE-Group.

---End

11.7.6.3 Creating an L3VE Interface

Context

Perform the following steps on NPEs. This part describes how to configure an L3VE interface that terminates the L3VPN, and how to bind the L3VE interface to the relevant VE group.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface virtual-ethernet** *interface-number*

A VE interface is created and the VE interface view is displayed.

Step 3 Run **ve-group** *ve-group-id* **l3-access**

The VE interface is set to an L3VE interface that accesses the MPLS L3VPN, and it is bound to a VE-Group.

Step 4 (Optional) Run either of the following commands:

- Run **l3ve track pw-state**

Associates the L3VE interface state with the PW state.

In the L2VPN access L3VPN solution, if the primary and secondary PWs have been established and corresponding L3VE interfaces have gone Up when the primary RSG recovers, the downstream traffic may switch back to the primary RSG. However, if the downstream traffic switches back to the primary RSG before the primary/secondary PW switchover is complete, the downstream traffic may be temporarily dropped. To solve this problem, run the **l3ve track pw-state** command to associate the L3VE interface status with the PW status.

This configuration is mainly used in dynamic PW scenarios.

- Run **l3ve track oam-state**

Associates the L3VE interface state with the OAM state.

In L2VPN accessing L3VPN scenarios in which PW protection is configured, if OAM (BFD, MPLS OAM, or MPLS-TP OAM) detects that services are interrupted between an AGG and a CSG, OAM does not trigger the L3VE interface to withdraw reverse routes. Therefore, reverse traffic is still forwarded along the faulty PW until route convergence is complete, which causes traffic loss. To resolve this problem, run the **l3ve track oam-state** command to enable the association between the L3VE interface status and the OAM status.

This configuration is mainly used in static PW scenarios.

----End

11.7.6.4 Associating the L2VE Interface with a VLL

Context

Perform the following steps on NPEs. This part describes how to associate an L2VE interface with a Martini VLL or a SVC VLL. At present, an L2VE interface can only be bound to a Martini VLL or a SVC VLL.

Procedure

- Configure the Martini VLL.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **mpls l2vpn**

The MPLS-L2VPN view is displayed.
 - c. Run **quit**

Return to the system view.
 - d. Run **interface virtual-ethernet interface-number.subinterface-number**

The L2VE sub-interface view is displayed.

At present, only the L2VE sub-interface can be configured with L2VPN. The VC type of the Martini VLL for the VE sub-interface is VLAN.

- e. Run **dot1q termination vid** *low-pe-vid*

Setting the single VLAN ID for Dot1q termination on a sub-interface.

- f. Run **mpls l2vc** *ip-address vc-id* [[**control-word** | **no-control-word**]] [**raw** | **tagged**] | **tunnel-policy** *policy-name*] *

A Martini VLL is created.

The tunnel policy for a Martini VLL defaults to LSP and no load balancing is performed. If a tunnel of another type is needed, you can specify **tunnel-policy** *policy-name* to apply the corresponding tunnel policy.

To create a Martini VLL, you need to specify the IP address and VC ID of the destination PE. The VC IDs of PEs at both ends of the VC must be consistent.

- Configure the SVC VLL.

- a. Run **system-view**

The system view is displayed.

- b. Run **mpls l2vpn**

The MPLS L2VPN is enabled.

- c. Run **quit**

Return to the system view.

- d. Run **interface virtual-ethernet** *interface-number.subinterface-number*

The L2VE sub-interface view is displayed.

- e. Run **dot1q termination vid** *low-pe-vid*

Setting the single VLAN ID for Dot1q termination on a sub-interface.

- f. Run **mpls static-l2vc** { { **destination** *ip-address* | **pw-template** *pw-template-name vc-id* } * **transmit-vpn-label** *transmit-label-value* **receive-vpn-label** *receive-label-value* [[**tunnel-policy** *tnl-policy-name*]] [{ **control-word** | **no-control-word** }] | [{ **raw** | **tagged** }] | [**secondary**]] *

A static VC is created.

 **NOTE**

The parameters **raw** and **tagged** are needed only for the Ethernet link.

---End

11.7.6.5 Configuring the Access of a User to L3VPN

Context

Perform the following steps on NPEs.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface virtual-ethernet** *interface-number.subinterface-number*

The L3VE interface view is displayed.

At present, only the L2VE sub-interface can be configured with L2VPN.

Step 3 Run **dot1q termination vid** *low-pe-vid*

Setting the single VLAN ID for Dot1q termination on a sub-interface.

Step 4 Run **ip binding vpn-instance** *vpn-instance-name*

The L3VE interface is associated with a VPN instance.

Step 5 Run **ip address** *ip-address* { *mask* | *mask-length* }

An IP address is configured for the L3VE interface.

 **NOTE**

The IP address is a private network address of MPLS L3VPN.

----End

11.7.6.6 Verifying the configuration of Martini VLL to Access L3VPN

Context

After configuring a Martini VLL to access L3VPN successfully, you can view the binding relationship between VE interfaces and the VE group, and information about the Martini VLL.

Procedure

- Run the **display virtual-ethernet ve-group** [*ve-group-id* | *slot slot-id*] command to check the binding relationship between VE interfaces and the VE-Group.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number*] command to check information about a Martini VLL.
- Run the **display vll ccc** [*ccc-name* | **type** { **local** | **remote** }] command to check the status of the local CCC.

----End

11.7.7 Configuring and Applying a Tunnel Policy

You also need to configure tunnel policies when VLL services need to be transmitted over TE tunnels or when VLL services need to be load balanced among multiple tunnels to fully use network resources.

Context

Layer 2 data on the VLL network is transmitted over tunnels. By default, LSP tunnels are used to transmit data, and each service is transmitted by only one LSP tunnel.

If the default tunnel configuration cannot meet VLL service requirements, apply tunnel policies to VLLs. You can configure either of the following types of tunnel policies:

- Tunnel type prioritization policy: This policy can change the type of tunnels selected for VPN data transmission or select multiple tunnels for load balancing.
- Tunnel binding policy: This policy can bind multiple TE tunnels to provide QoS guarantee for a VPN.

 **NOTE**

CCC VLL does not support tunnel policies.

Pre-configuration Tasks

Before configuring and applying a tunnel policy, complete the following task:

Creating a tunnel (an LSP, a GRE, or an MPLS TE tunnel) to transmit VLL services

For details on how to create an LSP tunnel, a GRE, and a TE tunnel, see *MPLS LDP Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS, GRE Configuration* and MPLS TE Configuration in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS*.

Perform the following operations on the PE devices that need to use a tunnel policy:

Procedure

Step 1 Configure a tunnel policy.

Use either of the following methods to configure a tunnel policy.

Configure a tunnel type prioritization policy.

By default, no tunnel policy is configured. LSP tunnels are used to transmit VLL data and each VLL service is transmitted over one LSP tunnel.

1. Run **system-view**
The system view is displayed.
2. Run **tunnel-policy** *policy-name*
A tunnel policy is created, and tunnel policy view is displayed.
3. (Optional) Run **description** *description-information*
The description of the tunnel policy is configured.
4. Run **tunnel select-seq** { **cr-lsp** | **gre** | **lsp** } * **load-balance-number** *load-balance-number*
The sequence in which each type of tunnel is selected and the number of tunnels participating in load balancing are set.

Configure a tunnel binding policy.

1. Run **system-view**
The system view is displayed.
2. Run **interface tunnel** *interface-number*
The tunnel interface view of the MPLS TE tunnel is displayed.

3. Run **mpls te reserved-for-binding**
The binding capability of the TE tunnel is enabled.
4. Run **mpls te commit**
The MPLS TE configuration is committed for the configuration to take effect.
5. Run **quit**
Return to the system view.
6. Run **tunnel-policy** *policy-name*
A tunnel policy is created.
7. (Optional) Run **description** *description-information*
The description of the tunnel policy is configured.
8. Run **tunnel binding destination** *dest-ip-address* **te** { **tunnel** *interface-number* }
&<1-16> [**ignore-destination-check**] [**down-switch**]
The TE tunnel is bound to a specified tunnel policy.

 **NOTE**

If **down-switch** is specified in the command, the system selects available tunnels in an order of LSP, CR-LSP, and GRE when the bound tunnels are unavailable.

Step 2 Apply the tunnel policy.

- In Martini VLL Mode
Perform the following operations on the AC interfaces of the PE devices:
 - Run **mpls l2vc** { *ip-address* | **pw-template** *pw-template-name* } * *vc-id* **tunnel-policy** *policy-name* [[**control-word** | **no-control-word**] | [**raw** | **tagged**] | **mtu** *mtu-value* | **secondary**] *
A Martini VLL connection is created and the tunnel policy is applied to the connection.
- In SVC VLL Mode
Perform the following operations on the AC interfaces of the PE devices:
 - Run **mpls static-l2vc** { { **destination** *ip-address* | **pw-template** *pw-template-name* } * | **destination** *ip-address* [*vc-id*] } **transmit-vpn-label** *transmit-label-value* **receive-vpn-label** *receive-label-value* **tunnel-policy** *tnl-policy-name* [[**control-word** | **no-control-word**] | [**raw** | **tagged**] | **secondary**] *
An SVC VLL connection is created and the tunnel policy is applied to the connection.

----End

Verifying the Configuration

After configuring a tunnel policy and applying it to a VLL, you can check information about the tunnel policy applied to the VLL and tunnels in the system.

- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.
- Run the **display interface tunnel** *interface-number* command to check detailed information about a specified tunnel interface.

- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check information about the specified tunnel policy.
- Run the **display mpls static-l2vc** [*vc-id* | **interface** *interface-type interface-number* | **state** { **down** | **up** }] command to check tunnel information about the SVC VLL.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number* | **remote-info** [*vc-id* | **verbose**] | **state** { **down** | **up** }] command to check tunnel information about the Martini VLL.

11.7.8 Configuring the Alarm Report Function

Context

You can configure the alarm report function, which help you obtain real-time running status of the VLL network and facilitate operation and maintenance.

Procedure

- Configure alarm report for the CCC VLL.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **snmp-agent trap enable feature-name l2vpn trap-name { hwcccvcdown | hwcccvcup }**

Alarm report for the CCC VLL is enabled.

By default, alarm report for the CCC VLL is disabled.
- Configure alarm report for the Martini VLL.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **snmp-agent trap enable feature-name l2vpn trap-name { hwpwvcbkup | hwpwvcdeleted | hwpwvcdown | hwpwvcstatuschange | hwpwvcswitchptow | hwpwvcswitchwtop | hwpwvcup }**

Alarm report for the Martini VLL is enabled.

By default, alarm report for the Martini VLL is disabled.
- Configure alarm report for the SVC VLL.
 - a. Run **system-view**

The system view is displayed.
 - b. Run **snmp-agent trap enable feature-name l2vpn trap-name { hwsvcdeleted | hwsvcdown | hwsvcswitchptow | hwsvcswitchwtop | hwsvcup }**

Alarm report for the SVC VLL is enabled.

By default, alarm report for the SVC VLL is disabled.

----End

Verifying the Configuration

After completing the alarm report for VLL, you can run the following command to check whether alarm report is enabled.

- Run the **display snmp-agent trap feature-name l2vpn all** command to check whether alarm report is enabled for the L2VPN module.

11.8 Maintaining VLL

This section describes how to maintain VLL, including collecting, querying, and clearing VLL statistics, checking VLL network connectivity, and resetting BGP TCP connections.

11.8.1 Monitoring the Running Status of VLL

Context

This section describes how to monitor the VLL running status by viewing the VLL connection information.

During the routine maintenance, you can run the following commands in any view to know the running status of VLL.

Procedure

- Run the **display vll ccc [ccc-name | type local]** command to check information about the CCC connection.
- Run the **display mpls static-l2vc [interface interface-type interface-number]** command to check information about the SVC VLL connection.
- Run the **display mpls l2vc [vc-id | interface interface-type interface-number]** command to check information about the local Martini VLL connection on the PE.
- Run the **display mpls l2vpn label-space** command to check information about label space distribution and different types of labels in the label cache.
- Run the **display mpls l2vpn vpws [interface interface-type interface-number [verbose]]** command to check information about VPWS service.

----End

11.8.2 Checking Connectivity of the VLL Network

Prerequisites

After a VLL network is established, perform a virtual circuit connectivity verification (VCCV) to check connectivity of the VLL network.

VCCV is an end-to-end fault detection and diagnosis mechanism which includes VCCV ping and VCCV tracer. VCCV ping is a tool for manually testing VC connectivity; while VCCV tracer is tool for manually locating an abnormal node on a PW.

VCCV supports the following modes: control word channel mode and Label Alert mode. By default, VCCV in Label Alert mode is enabled. Before using the control word channel mode,

you need to run the **control-word** command to enable the control word function. After that, VCCV in control word channel mode can be used.

 **NOTE**

When locating the fault on the VLL network in Martini mode, you can use either VCCV in control word channel mode or VCCV in normal mode.

Procedure

- Check the connectivity of the VLL network.

Checking the connectivity of the VLL network in Martini mode.

- Control word channel

```
ping vc pw-type pw-id [ -c echo-number | -m time-value | -s data-bytes | -t timeout-value | -exp exp-value | -r reply-mode | -v ] * control-word [ remote remote-ip-address peer-pw-id | draft6 ] * [ ttl ttl-value ] [ uniform ]
```

```
ping vc pw-type pw-id [ -c echo-number | -m time-value | -s data-bytes | -t timeout-value | -exp exp-value | -r reply-mode | -v ] * control-word remote remote-ip-address peer-pw-id sender sender-address [ ttl ttl-value ] [ uniform ]
```

- Label Alert channel

```
ping vc pw-type pw-id [ -c echo-number | -m time-value | -s data-bytes | -t timeout-value | -exp exp-value | -r reply-mode | -v ] * label-alert [ no-control-word ] [ remote remote-ip-address | draft6 ] * [ uniform ]
```

- Normal mode

```
ping vc pw-type pw-id [ -c echo-number | -m time-value | -s data-bytes | -t timeout-value | -exp exp-value | -r reply-mode | -v ] * normal [ no-control-word ] [ remote remote-ip-address peer-pw-id ] [ ttl ttl-value ] [ uniform ]
```

- Locate the fault on the VLL network.

Locating the fault on the VLL network in Martini mode.

- Control word channel

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * control-word [ draft6 ] [ full-lsp-path ] [ uniform ]
```

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * control-word remote remote-ip-address [ ptn-mode | full-lsp-path ] [ uniform ]
```

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * control-word remote remote-pw-id draft6 [ full-lsp-path ] [ uniform ]
```

- Label Alert channel

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * label-alert [ no-control-word ] [ remote remote-ip-address ] [ full-lsp-path ] [ draft6 ] [ uniform ]
```

- Normal mode

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * normal [ no-control-word ] [ remote remote-ip-address ] [ full-lsp-path ] [ draft6 ] [ uniform ]
```

----End

11.9 Configuration Examples for VLL

This section describes VLL configuration examples including the networking requirements, configuration notes, and configuration roadmap.

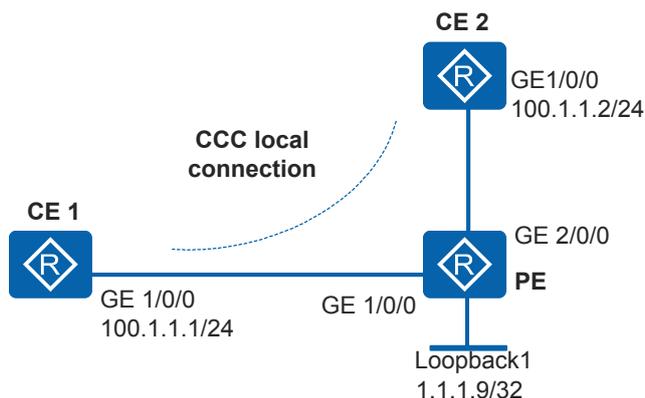
11.9.1 Example for Configuring a Local CCC Connection

Networking Requirements

As shown in [Figure 11-19](#), sites of an enterprise at different geographical locations connect to a PE on the ISP network through CE1 and CE2. To simplify configuration, the enterprise hopes that the two CEs communicate with each other like on a LAN. That is, data packets of each CE traverse the ISP network without being modified by the PE. The enterprise will not increase sites in the future and wants to use exclusive VPN resources on the ISP network to protect data security.

To meet requirements of the enterprise, create a local CCC connection to enable CE1 and CE2 to exchange Layer 2 information directly.

Figure 11-19 Local CCC connection



Configuration Roadmap

The enterprise hopes that the two CEs communicate with each other like on a LAN. This requirement can be satisfied by a VLL solution. Because the enterprise will not increase sites in the future and the two CE1 connected to the same PE, a local CCC connection can be set up between the CEs to meet the customer requirements.

The configuration roadmap is as follows:

1. Configure the basic MPLS capabilities on the PE and enable the MPLS L2VPN. Enabling MPLS L2VPN is the prerequisite for VLL configuration.
2. Create a local connection between CE1 and CE2 on PE. The local CCC connection is bidirectional, so only one connection is required.

Procedure

Step 1 Configure the CEs.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 24
[CE1-GigabitEthernet1/0/0] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] ip address 100.1.1.2 24
[CE2-GigabitEthernet1/0/0] quit
```

Step 2 Configure PE.

Configure the LSR ID and enable MPLS and MPLS L2VPN.

```
<Huawei> system-view
[Huawei] sysname PE
[PE] interface loopback 1
[PE-LoopBack1] ip address 1.1.1.9 32
[PE-LoopBack1] quit
[PE] mpls lsr-id 1.1.1.9
[PE] mpls
[PE-mpls] quit
[PE] mpls l2vpn
[PE-l2vpn] quit
```

Create a local connection between CE1 and CE2.

```
[PE] ccc ce1-ce2 interface gigabitethernet 1/0/0 out-interface gigabitethernet 2/0/0
```

Step 3 Verify the configuration.

After completing the configuration, check the CCC information on the PE. The command output shows that a local CCC connection has been set up and the status is Up.

```
[PE] display vll ccc
total ccc vc : 1
local ccc vc : 1, 1 up
remote ccc vc : 0, 0 up

name: ce1-ce2, type: local, state: up,
intf1: GigabitEthernet1/0/0 (up),access-port: false

intf2: GigabitEthernet2/0/0 (up),access-port: false
VC last up time : 2010/07/24 12:31:31
VC total up time: 0 days, 2 hours, 12 minutes, 51 seconds
```

Run the **display l2vpn ccc-interface vc-type all** command. The command output shows that the VC type is ccc and the VC status is Up.

```
[PE] display l2vpn ccc-interface vc-type all
Total ccc-interface of CCC : 2
up (2), down (0)
Interface                               Encap Type           State   VC Type
GigabitEthernet1/0/0                    ethernet             up      ccc
GigabitEthernet2/0/0                    ethernet             up      ccc
```

CE1 and CE2 can ping each other.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=180 ms
```

```
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=60 ms
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=10 ms
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=70 ms
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=60 ms
```

```
--- 100.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 10/76/180 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE

```
#
sysname PE
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
ccc ce1-ce2 interface GigabitEthernet1/0/0 out-interface GigabitEthernet2/0/0
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
return
```

- Configuration file of CE2

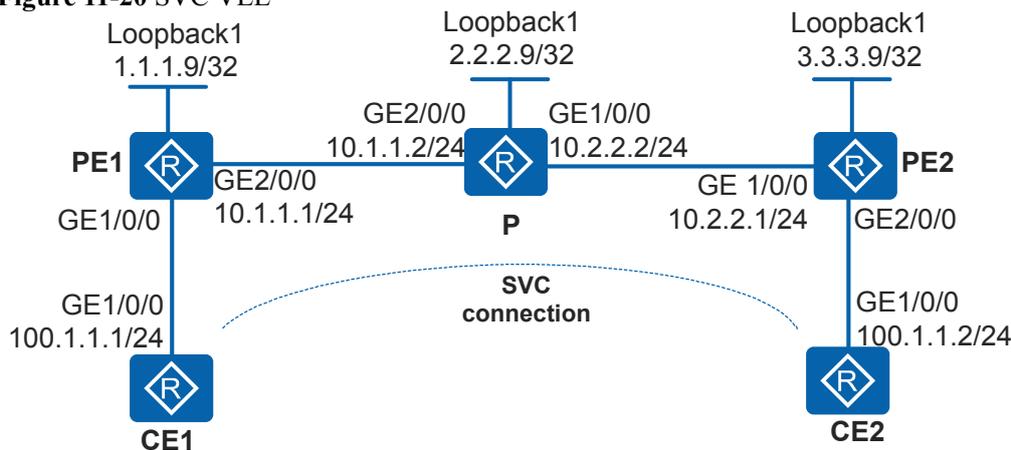
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
```

11.9.2 Example for Configuring a VLL Connection in SVC Mode

Networking Requirements

The MPLS network of an ISP provides the L2VPN service for sites of two users. Each user has two sites at fixed locations, which connect to the MPLS network through CE1 and CE2. The users hope that hosts in different sites can communicate at Layer 2.

Figure 11-20 SVC VLL



Configuration Roadmap

The users expect direct Layer 2 communication between their sites. VLL can be configured to satisfy this requirement. The two PEs have fixed users, so signaling information can be manually configured (SVC mode).

The configuration roadmap is as follows:

1. Configure an IGP on the MPLS backbone network to implement IP interworking.
2. Configure basic MPLS functions and LDP on the MPLS backbone network and set up an LDP LSP tunnel. The LDP LSP tunnel is used as a dedicated tunnel to transmit data of private networks on the public network.
3. On the PEs, enable MPLS L2VPN, create a static VC connection, and manually configure VC labels. Enabling MPLS L2VPN is the prerequisite for VLL configuration, and creating a static VC connection is the most important step in configuring VLL of the SVC mode.

Procedure

Step 1 Configure IP addresses to each interface according to [Figure 11-20](#).

Configure CE1. The configuration on PE1, P, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure IGP on the MPLS backbone network. (In this example, OSPF is used.)

When configuring OSPF, advertise the 32-bit addresses of loopback interfaces on PEs and P. The loopback interface addresses are the LSR IDs.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] area 0
```

```
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Step 3 Configure basic MPLS functions and LDP on the MPLS backbone network. That is, set up LDP LSPs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

Configure the P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After completing the configuration, LDP sessions are set up between PE1 and P, and between PE2 and P. Run the **display mpls ldp session** command. The command output shows that the status of the LDP session is **Operational**.

Take the display on PE1 for example:

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0             Operational DU   Passive  0000:00:05  22/22
-----
TOTAL: 1 session(s) Found.
```

Step 4 Enable MPLS L2VPN and create static VCs on PEs.

Configure PE1: Create a static VC on GE1/0/0, which is connected to CE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls static-l2vc destination 3.3.3.9 transmit-vc-label 100 receive-vc-label 200
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2: Create a static VC on GE2/0/0, which is connected to CE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls static-l2vc destination 1.1.1.9 transmit-vc-label 200 receive-vc-label 100
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

View the L2VPN connection information of the SVC on the PE. The command output shows that a static L2VC connection is established.

Take PE1 for example.

```
[PE1] display mpls static-l2vc interface gigabitethernet 1/0/0
*Client Interface      : GigabitEthernet1/0/0 is up
AC Status              : up
VC State               : up
VC ID                  : 0
VC Type                : Ethernet
Destination            : 3.3.3.9
Transmit VC Label     : 100
Receive VC Label      : 200
Label Status           : 0
Token Status           : 0
Control Word           : Disable
VCCV Capabilty        : alert ttl lsp-ping bfd
active state           : active
Link State             : up
Tunnel Policy          : --
PW Template Name      : --
Main or Secondary     : Main
load balance type     : flow
Access-port           : false
VC tunnel/token info  : 1 tunnels/tokens
NO.0 TNL Type         : lsp , TNL ID : 0x3
Backup TNL Type       : lsp , TNL ID : 0x0
Create time           : 0 days, 0 hours, 4 minutes, 31 seconds
UP time               : 0 days, 0 hours, 2 minutes, 14 seconds
Last change time      : 0 days, 0 hours, 2 minutes, 14 seconds
VC last up time       : 2012/08/16 19:05:13
VC total up time      : 0 days, 0 hours, 2 minutes, 14 seconds
CKey                  : 4
NKey                   : 3
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --
BFDD for PW           : unavailable
```

Run the **display l2vpn ccc-interface vc-type static-vc up** command, you can view information about an interface on which the VC Type is displayed as static-vc and State is Up.. Take the display on PE1 for example.

```
[PE1] display l2vpn ccc-interface vc-type static-vc up
Total ccc-interface of SVC VC: 1
up (1), down (0)
Interface                Encap Type      State   VC Type
GigabitEthernet1/0/0    ethernet        up      static-vc
```

CE1 and CE2 can ping each other.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=46 ms
  Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=91 ms
  Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=74 ms
  Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=88 ms
  Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=82 ms
--- 100.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 46/76/91 ms
```

---End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
mpls ldp
#
interface GigabitEthernet1/0/0
mpls static-l2vc destination 3.3.3.9 transmit-vpn-label 100 receive-vpn-label 200
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 10.1.1.0 0.0.0.255
#
return
```

- Configuration file of P

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet2/0/0
```

```
ip address 10.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 10.2.2.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 10.2.2.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
mpls static-l2vc destination 1.1.1.9 transmit-vpn-label 200 receive-vpn-label 100
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
```

11.9.3 Example for Configuring a VLL Connection in Martini Mode

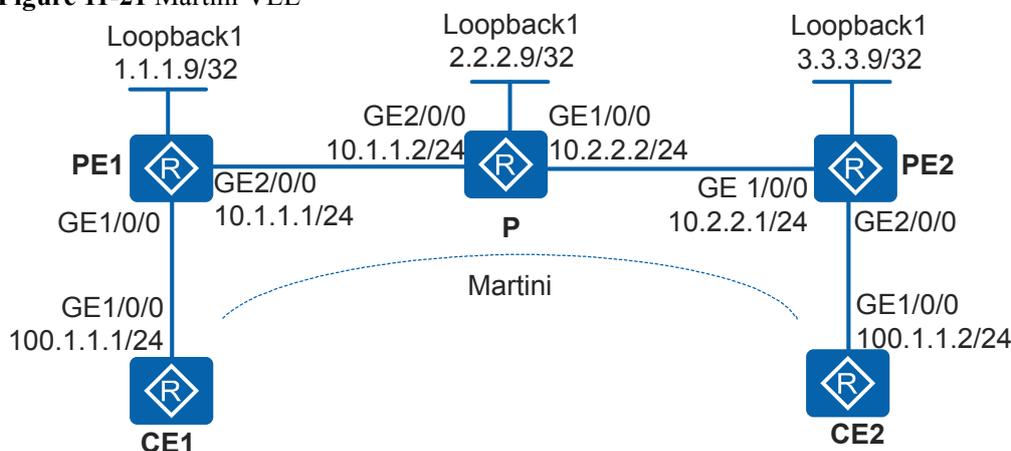
Networking Requirements

As shown in [Figure 11-21](#), the MPLS network of an ISP provides the L2VPN service for users. Many users connect to the MPLS network through PE1 and PE2, and users on the PEs

change frequently. A proper VPN solution is required to provide secure VPN services for users and to simplify configuration when new users connect to the network.

A Martini VLL connection can be set up between CE1 and CE2 to meet these requirements.

Figure 11-21 Martini VLL



Configuration Roadmap

Because users on the PEs change frequently, manual configuration is inefficient and may cause configuration errors. In this scenario, the two PEs can set up a remote LDP connection and use the LDP protocol to synchronize user information (VC IDs). This implementation is the Martini mode.

The configuration roadmap is as follows:

1. Configure an IGP on the PE and P devices on the backbone network to ensure reachability between them, and enable MPLS.
2. This example uses the default tunnel policy to set up an LSP tunnel. The LSP tunnel is used as a dedicated tunnel to transmit data of private networks on the public network.
3. Set up a remote LDP session between the PEs to exchange VC labels between the PEs.
4. Enable MPLS L2VPN and create VC connections on the PEs. Enabling MPLS L2VPN is the prerequisite for VLL configuration.

Procedure

Step 1 Configure IP addresses for interfaces on the CE, PE and P devices according to [Figure 11-21](#).

Configure CE1. The configuration on PE1, P, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure IGP on the MPLS backbone network. (In this example, OSPF is used.)

When configuring OSPF, advertise the 32-bit addresses of loopback interfaces on PEs and P. The loopback interface addresses are the LSR IDs.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Step 3 Configure the basic MPLS capabilities and MPLS LDP on the MPLS network.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

Configure the P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

Step 4 Set up a remote LDP session between PEs.

Configure PE1.

```
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure PE2.

```
[PE2] mpls ldp remote-peer 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] quit
```

After the configuration, run the **display mpls ldp session** command on PE1 to view the establishment of the LDP session. You can find that an LDP session is set up between PE1 and PE2.

Take the display on PE1 for example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0             Operational DU   Passive  0000:00:11  46/45
3.3.3.9:0             Operational DU   Passive  0000:00:01  8/8
-----
TOTAL: 2 session(s) Found.
```

Step 5 Enable MPLS L2VPN and create VCs on the PEs.

Configure PE1: Create a VC on GE1/0/0, which is connected to CE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 3.3.3.9 101
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2: Create a VC on GE2/0/0, which is connected to CE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc 1.1.1.9 101
[PE2-GigabitEthernet2/0/0] quit
```

Step 6 Verify the configuration.

View the L2VPN connection information on the PEs, and you can see that an L2VC is set up and is in Up state.

Take the display on PE1 for example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is
up
 Administrator PW      : no
 session state         : up
 AC status             : up
 VC state              : up
 Label state           : 0
 Token state           : 0
 VC ID                 : 101
 VC type               : Ethernet
 destination           : 3.3.3.9
 local group ID        : 0          remote group ID      : 0
 local VC label        : 1024       remote VC label      : 1024
 local AC OAM State    : up
 local PSN OAM State   : up
 local forwarding state : forwarding
 local status code     : 0x0
 remote AC OAM state   : up
 remote PSN OAM state  : up
 remote forwarding state : forwarding
 remote status code    : 0x0
 ignore standby state  : no
```

```
BFD for PW           : unavailable
VCCV State           : up
manual fault         : not set
active state         : active
forwarding entry     : exist
link state           : up
local VC MTU         : 1500           remote VC MTU           : 1500
local VCCV           : alert ttl lsp-ping bfd
remote VCCV          : alert ttl lsp-ping bfd
local control word   : disable       remote control word   : disable
tunnel policy name   : --
PW template name     : --
primary or secondary : primary
load balance type    : flow
Access-port          : false
Switchover Flag      : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type      : lsp , TNL ID : 0x5
  Backup TNL type    : lsp , TNL ID : 0x0
create time          : 0 days, 0 hours, 27 minutes, 15 seconds
up time              : 0 days, 0 hours, 2 minutes, 22 seconds
last change time     : 0 days, 0 hours, 2 minutes, 22 seconds
VC last up time      : 2011/09/26 15:29:03
VC total up time     : 0 days, 0 hours, 2 minutes, 22 seconds
CKey                 : 5
NKey                 : 4
PW redundancy mode   : frr
AdminPw interface    : --
AdminPw link state   : --
Diffserv Mode        : uniform
Service Class        : --
Color                : --
DomainId             : --
Domain Name          : --
```

CE1 and CE2 can ping each other.

Take the display on CE1 for example.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=31 ms
  Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=10 ms
  Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=28 ms
--- 100.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/15/31 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
```

```
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
  remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
  mpls l2vc 3.3.3.9 101
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
ospf 1
  area 0.0.0.0
    network 1.1.1.9 0.0.0.0
    network 10.1.1.0 0.0.0.255
#
return
```

● Configuration file of P

```
#
  sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 10.2.2.2 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.2 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack1
  ip address 2.2.2.9 255.255.255.255
#
ospf 1
  area 0.0.0.0
    network 2.2.2.9 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 10.2.2.0 0.0.0.255
#
return
```

● Configuration file of PE2

```
#
  sysname PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls ldp
#
```

```

mpls ldp remote-peer 1.1.1.9
remote-ip 1.1.1.9
#
interface GigabitEthernet1/0/0
ip address 10.2.2.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
mpls l2vc 1.1.1.9 101
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 10.2.2.0 0.0.0.255
#
return
    
```

● Configuration file of CE2

```

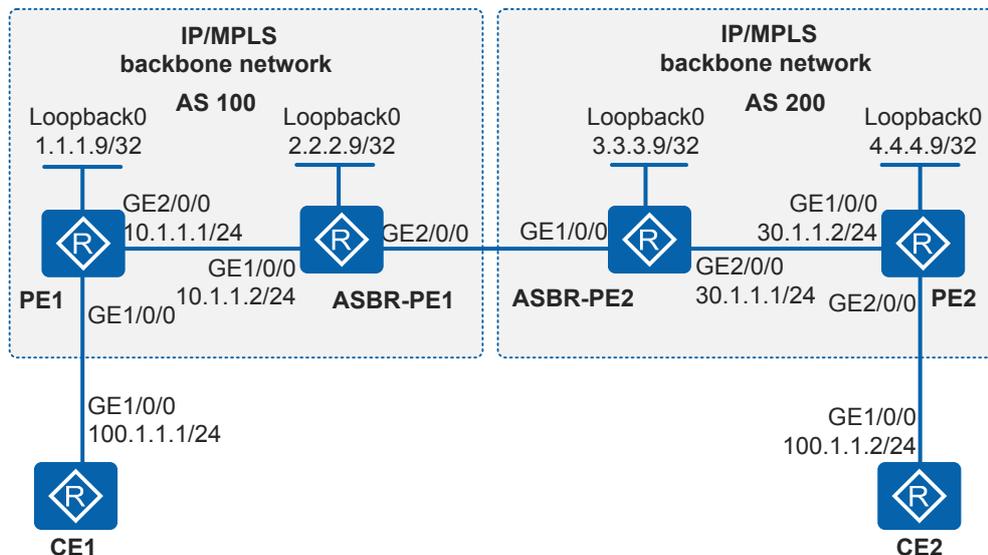
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
    
```

11.9.4 Example for Configuring Inter-AS Martini VLL (Option A)

Networking Requirements

As shown in [Figure 11-22](#), the MPLS network of an ISP provides the L2VPN service for users. PE1 becomes to AS 100 and PE2 belongs to AS 200. Many users connect to the MPLS network through PE1 and PE2, and users on the PEs change frequently. A proper VPN solution is required to provide secure VPN services for users and to simplify configuration when new users connect to the network.

Figure 11-22 Inter-AS Martini VLL (Option A)



Configuration Roadmap

The PEs connect to different ASs (AS100 and AS200) of the ISP, so an inter-AS VPN solution is required. To simplify configuration when new users connect to the network, configure Martini VLL using inter-AS Option A.

The configuration roadmap is as follows:

1. Run an IGP protocol on the backbone network so that the devices in the same AS can communicate with each other.
2. Configure the basic MPLS capabilities on the backbone network and set up dynamic LSPs between PEs and ASBR-PEs in the same AS. If PEs and ASBR-PEs are not directly connected, set up a remote LDP session.
3. Establish MPLS L2VC connections between the PEs and ASBR-PEs in the same AS.

Procedure

Step 1 Configure IP addresses for interfaces according to [Figure 11-22](#).

Configure CE1. The configuration on PE1, ASBR-PE1, ASBR-PE2, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol on the MPLS backbone network.

PEs and ASBR-PEs on the backbone network can communicate with each other by using IGP.

In this example, IS-IS is used as IGP.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] isis 1
[PE1-isis-1] network-entity 10.0000.0000.0001.00
[PE1-isis-1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] isis enable 1
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 0
[PE1-LoopBack0] isis enable 1
[PE1-LoopBack0] quit
```

After the configuration, the ASBR and PE in the same AS can establish an IS-IS adjacency. Run the **display isis peer** command, and you can see that the IS-IS adjacency is in Up state, and the PEs can learn each other's loopback address.

Take the display on PE1 for example.

```
[PE1] display isis peer

Peer information for ISIS(1)

System Id      Interface      Circuit Id      State HoldTime Type      PRI
-----
0000.0000.0002 GE2/0/0        0000.0000.0002.01 Up    23s      L1 (L1L2) 64
```

```
0000.0000.0002 GE2/0/0 0000.0000.0002.01 Up 22s L2 (L1L2) 64
Total Peer(s): 2
```

The ASBR and PE in the same AS can ping each other.

Take the display on PE1 for example.

```
[PE1] ping 2.2.2.9
PING 2.2.2.9: 56 data bytes, press CTRL_C to break
  Reply from 2.2.2.9: bytes=56 Sequence=1 ttl=255 time=180 ms
  Reply from 2.2.2.9: bytes=56 Sequence=2 ttl=255 time=90 ms
  Reply from 2.2.2.9: bytes=56 Sequence=3 ttl=255 time=60 ms
  Reply from 2.2.2.9: bytes=56 Sequence=4 ttl=255 time=60 ms
  Reply from 2.2.2.9: bytes=56 Sequence=5 ttl=255 time=100 ms

--- 2.2.2.9 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 60/98/180 ms
```

Step 3 Enable MPLS and configure dynamic LSPs.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

After this step, an LSP is established between the PE and ASBR-PE in the same AS.

Take the display on ASBR-PE1 for example.

```
[ASBR-PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
1.1.1.9:0        Operational DU   Active  0000:00:19  79/79
-----
TOTAL: 1 session(s) Found.
```

Step 4 Configure the MPLS L2VC connection.

Configure the L2VC connection on the PE and ASBR-PE and connect the PE to the CE.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 2.2.2.9 100
[PE1-GigabitEthernet1/0/0] quit
```

Configure ASBR-PE1.

```
[ASBR-PE1] mpls l2vpn
[ASBR-PE1-l2vpn] quit
```

```
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] mpls l2vc 1.1.1.9 100
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

Configure ASBR-PE2.

```
[ASBR-PE2] mpls l2vpn
[ASBR-PE2-l2vpn] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] mpls l2vc 4.4.4.9 100
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc 3.3.3.9 100
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

Display information about the L2VPN connection on PE1. You can see that an L2VC has been set up and the VC status is Up.

Take the display on PE1 and ASBR-PE2 for example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 100
VC type               : Ethernet
destination           : 2.2.2.9
local group ID        : 0          remote group ID      : 0
local VC label        : 8195       remote VC label       : 8195
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state : forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : active
forwarding entry      : exist
link state             : up
local VC MTU          : 1500       remote VC MTU        : 1500
local VCCV             : alert ttl lsp-ping bfd
remote VCCV           : alert ttl lsp-ping bfd
local control word    : disable     remote control word   : disable
tunnel policy name    : --
PW template name      : --
primary or secondary  : primary
load balance type     : flow
Access-port           : false
Switchover Flag       : false
VC tunnel/token info  : 1 tunnels/tokens
  NO.0 TNL type       : lsp , TNL ID : 0x10031
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 1 days, 22 hours, 15 minutes, 9 seconds
```

```
up time : 0 days, 22 hours, 54 minutes, 57 seconds
last change time : 0 days, 22 hours, 54 minutes, 57 seconds
VC last up time : 2010/10/09 19:26:37
VC total up time : 1 days, 20 hours, 42 minutes, 30 seconds
CKey : 16
NKey : 15
PW redundancy mode : frr
AdminPw interface : --
AdminPw link state : --
Diffserv Mode : uniform
Service Class : --
Color : --
DomainId : --
Domain Name : --
[ASBR-PE2] display mpls l2vc interface gigabitethernet 1/0/0
*client interface : GigabitEthernet1/0/0 is up
Administrator PW : no
session state : up
AC status : up
VC state : up
Label state : 0
Token state : 0
VC ID : 100
VC type : Ethernet
destination : 4.4.4.9
local group ID : 0 remote group ID : 0
local VC label : 8195 remote VC label : 8195
local AC OAM State : up
local PSN OAM State : up
local forwarding state : forwarding
local status code : 0x0
remote AC OAM state : up
remote PSN OAM state : up
remote forwarding state: forwarding
remote status code : 0x0
ignore standby state : no
BFD for PW : unavailable
VCCV State : up
manual fault : not set
active state : active
forwarding entry : exist
link state : up
local VC MTU : 1500 remote VC MTU : 1500
local VCCV : alert ttl lsp-ping bfd
remote VCCV : alert ttl lsp-ping bfd
local control word : disable remote control word : disable
tunnel policy name : --
PW template name : --
primary or secondary : primary
load balance type : flow
Access-port : false
Switchover Flag : false
VC tunnel/token info : 1 tunnels/tokens
NO.0 TNL type : lsp , TNL ID : 0x10031
Backup TNL type : lsp , TNL ID : 0x0
create time : 1 days, 22 hours, 15 minutes, 9 seconds
up time : 0 days, 22 hours, 54 minutes, 57 seconds
last change time : 0 days, 22 hours, 54 minutes, 57 seconds
VC last up time : 2010/10/09 19:26:37
VC total up time : 1 days, 20 hours, 42 minutes, 30 seconds
CKey : 17
NKey : 18
PW redundancy mode : frr
AdminPw interface : --
AdminPw link state : --
Diffserv Mode : uniform
Service Class : --
Color : --
```

```
DomainId          : --  
Domain Name      : --
```

CE1 and CE2 can ping each other.

Take the display on CE1 for example.

```
[CE1] ping 100.1.1.2  
PING 100.1.1.2: 56 data bytes, press CTRL_C to break  
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=172 ms  
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=156 ms  
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=156 ms  
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=156 ms  
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=156 ms  
  
--- 100.1.1.2 ping statistics ---  
5 packet(s) transmitted  
5 packet(s) received  
0.00% packet loss  
round-trip min/avg/max = 156/159/172 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#  
sysname CE1  
#  
interface GigabitEthernet1/0/0  
ip address 100.1.1.1 255.255.255.0  
#  
return
```

- Configuration file of PE1

```
#  
sysname PE1  
#  
mpls lsr-id 1.1.1.9  
mpls  
#  
mpls l2vpn  
#  
mpls ldp  
#  
isis 1  
network-entity 10.0000.0000.0001.00  
#  
interface GigabitEthernet1/0/0  
mpls l2vc 2.2.2.9 100  
#  
interface GigabitEthernet2/0/0  
ip address 10.1.1.1 255.255.255.0  
isis enable 1  
mpls  
mpls ldp  
#  
interface LoopBack0  
ip address 1.1.1.9 255.255.255.255  
isis enable 1  
#  
return
```

- Configuration file of ASBR-PE1

```
#  
sysname ASBR-PE1  
#  
mpls lsr-id 2.2.2.9
```

```
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 mpls l2vc 1.1.1.9 100
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of ASBR-PE2

```
#
 sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 mpls l2vc 4.4.4.9 100
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of PE2

```
#
 sysname PE2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
 isis enable 1
```


Configuration Roadmap

VLL FRR can be configured to ensure highly stable communication between CE1 and CE2. If a few sites will be added in the future, Martini VLL FRR can be configured.

The configuration roadmap is as follows:

1. Configure OSPF on the backbone network to implement interworking between backbone devices.
2. Set up an MPLS TE tunnel between PE1 and PE3, which will be used by the primary PW.
Set up an MPLS LSP tunnel between PE1 and PE2, which will be used by the secondary PW.
3. Configure a PW template on each PE to simplify the PW configuration. Configure a tunnel policy to enable the primary PW to use the MPLS TE tunnel.
4. Configure BFD for PW between PE1 and PE2 and between PE1 and PE3 to quickly detect a PW fault.
5. Configure Ethernet in the First Mile (EFM) between CE2 and PE2, and PE3 and CE2 to detect link connectivity.
6. Enable OAM mapping on the PEs so that L2VPN traffic can be quickly switched to the secondary PW. When fault on the primary PW is recovered, L2VPN traffic can be switched back to the primary PW.
7. Configure association between EFM and interfaces on CE2. When EFM detects a PW fault, the interface is logically Down, quickly switching traffic to a right forwarding path.
8. Set the revertive switchover policy to immediate switchover on PE1 to prevent data loss because CE2 can quickly detect a PW fault.

Procedure

Step 1 Assign IP addresses to interfaces.

Configure PE1. The configuration on P, PE2, PE3, CE1, and CE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet1/0/0] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 172.2.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack1] quit
```

Step 2 Configure an IGP protocol on the MPLS backbone network so that the PE and P devices can communicate with each other.

Configure PE1. The configuration on P, PE2, and PE3 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.0.255
```

```
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After completing the configuration, run the **display ip routing-table** command on each PE. The command output shows that PE1 and PE2, and PE1 and PE3 have learned the routes to each other's Loopback1 interface.

Step 3 Enable MPLS on backbone devices to set up an MPLS TE tunnel between PE1 and PE3 and an LDP LSP between PE1 and PE2.

- Set up an MPLS TE tunnel between PE1 and PE3.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] mpls te
[PE1-mpls] mpls rsvp-te
[PE1-mpls] mpls te cspf
[PE1-mpls] quit
[PE1] ospf 1
[PE1-ospf-1] opaque-capability enable
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] mpls-te enable
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls te
[PE1-GigabitEthernet2/0/0] mpls rsvp-te
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE1-Tunnel0/0/1] tunnel-protocol mpls te
[PE1-Tunnel0/0/1] destination 3.3.3.9
[PE1-Tunnel0/0/1] mpls te tunnel-id 100
[PE1-Tunnel0/0/1] mpls te commit
[PE1-Tunnel0/0/1] quit
```

Configure the P device.

```
[P] mpls lsr-id 4.4.4.9
[P] mpls
[P-mpls] mpls te
[P-mpls] mpls rsvp-te
[P-mpls] quit
[P] ospf 1
[P-ospf-1] opaque-capability enable
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] mpls-te enable
[P-ospf-1-area-0.0.0.0] quit
[P-ospf-1] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls te
[P-GigabitEthernet1/0/0] mpls rsvp-te
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls te
[P-GigabitEthernet2/0/0] mpls rsvp-te
[P-GigabitEthernet2/0/0] quit
```

Configure PE3.

```
[PE3] mpls lsr-id 3.3.3.9
[PE3] mpls
[PE3-mpls] mpls te
[PE3-mpls] mpls rsvp-te
[PE3-mpls] mpls te cspf
[PE3-mpls] quit
[PE3] ospf 1
```

```
[PE3-ospf-1] opaque-capability enable
[PE3-ospf-1] area 0
[PE3-ospf-1-area-0.0.0.0] mpls-te enable
[PE3-ospf-1-area-0.0.0.0] quit
[PE3-ospf-1] quit
[PE3] interface gigabitethernet 1/0/0
[PE3-GigabitEthernet1/0/0] mpls
[PE3-GigabitEthernet1/0/0] mpls te
[PE3-GigabitEthernet1/0/0] mpls rsvp-te
[PE3-GigabitEthernet1/0/0] quit
[PE3] interface tunnel 0/0/1
[PE3-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE3-Tunnel0/0/1] tunnel-protocol mpls te
[PE3-Tunnel0/0/1] destination 1.1.1.9
[PE3-Tunnel0/0/1] mpls te tunnel-id 101
[PE3-Tunnel0/0/1] mpls te commit
[PE3-Tunnel0/0/1] quit
```

- Set up an LDP LSP between PE1 and PE2.

Configure PE1.

```
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 2.2.2.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After completing the configuration, run the **display tunnel-info all** command on the PEs. The command output shows that an MPLS TE tunnel has been set up between PE1 and PE3, and an LSP tunnel has been set up between PE1 and PE2.

The display on PE1 is used as an example:

```
[PE1] display tunnel-info all
* -> Allocated VC Token
Tunnel ID      Type                Destination          Token
-----
0x1            cr lsp              3.3.3.9              1
0x2            lsp                 3.3.3.9              2
0x3            lsp                 2.2.2.9              3
0x4            lsp                 2.2.2.9              4
```

Step 4 Create a remote LDP session between PE1 and PE3.

When configuring a remote LDP session, specify the loopback interface address of the LDP remote peer as the IP address.

NOTE

PE1 and PE2 in this example are directly connected; therefore, you do not need to configure a remote LDP session between them.

Configure PE1.

```
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure PE3.

```
[PE3] mpls ldp
[PE3-mpls-ldp] quit
[PE3] mpls ldp remote-peer 1.1.1.9
[PE3-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE3-mpls-ldp-remote-1.1.1.9] quit
```

After completing the configuration, run the **display mpls ldp session** command on the PEs. The command output shows that the status of the remote LDP peer relationship is **Operational**. This indicates that remote LDP session has been set up.

The display on PE1 is used as an example:

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0             Operational DU   Passive  0000:00:02  9/9
3.3.3.9:0             Operational DU   Passive  0000:00:00  1/1
-----
TOTAL: 2 session(s) Found.
```

Step 5 Configure tunnel policies on the PEs.

Configure PE1.

```
[PE1] tunnel-policy p1
[PE1-tunnel-policy-p1] tunnel select-seq cr-lsp load-balance-number 1
[PE1-tunnel-policy-p1] quit
```

Configure PE3.

```
[PE3] tunnel-policy p1
[PE3-tunnel-policy-p1] tunnel select-seq cr-lsp load-balance-number 1
[PE3-tunnel-policy-p1] quit
```

Step 6 Use a PW template to create PWs on the PEs.

Create primary and secondary PWs on PE1. Create a PW on each of PE2 and PE3.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] pw-template 1to2
[PE1-pw-template-1to2] peer-address 2.2.2.9
[PE1-pw-template-1to2] control-word
[PE1-pw-template-1to2] quit
[PE1] pw-template 1to3
[PE1-pw-template-1to3] peer-address 3.3.3.9
[PE1-pw-template-1to3] control-word
[PE1-pw-template-1to3] quit
[PE1] interface gigabitEthernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls l2vc pw-template 1to3 200 tunnel-policy p1
[PE1-GigabitEthernet3/0/0] mpls l2vc pw-template 1to2 201 secondary
[PE1-GigabitEthernet3/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] pw-template 2to1
[PE2-pw-template-2to1] peer-address 1.1.1.9
[PE2-pw-template-2to1] control-word
```

```
[PE2-pw-template-2to1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc pw-template 2to1 201
[PE2-GigabitEthernet2/0/0] quit
```

Configure PE3.

```
[PE3] mpls l2vpn
[PE3-l2vpn] quit
[PE3] pw-template 3to1
[PE3-pw-template-3to1] peer-address 1.1.1.9
[PE3-pw-template-3to1] control-word
[PE3-pw-template-3to1] quit
[PE3] interface gigabitethernet 2/0/0
[PE3-GigabitEthernet2/0/0] mpls l2vc pw-template 3to1 200 tunnel-policy pl
[PE3-GigabitEthernet2/0/0] quit
```

Step 7 Configure devices to ensure network connectivity.

Configure two default routes on CE2, specify GE1/0/0 and GE2/0/0 as outbound interfaces, and assign preference to the route with GE1/0/0 as the outbound interface.

Configure CE2.

```
[CE2] ip route-static 0.0.0.0 0.0.0.0 gigabitethernet 1/0/0 192.168.1.2
[CE2] ip route-static 0.0.0.0 0.0.0.0 gigabitethernet 2/0/0 192.168.2.2
      preference 100
```

Step 8 Configure BFD for PW on the PEs.

Configure PE1.

```
[PE1] bfd
[PE1-bfd] quit
[PE1] bfd pe1tope3 bind pw interface gigabitethernet 3/0/0
[PE1-bfd-lsp-session-pe1tope3] discriminator local 100
[PE1-bfd-lsp-session-pe1tope3] discriminator remote 200
[PE1-bfd-lsp-session-pe1tope3] min-tx-interval 100
[PE1-bfd-lsp-session-pe1tope3] min-rx-interval 100
[PE1-bfd-lsp-session-pe1tope3] commit
[PE1-bfd-lsp-session-pe1tope3] quit
[PE1] bfd pe1tope2 bind pw interface gigabitethernet 3/0/0 secondary
[PE1-bfd-lsp-session-pe1tope2] discriminator local 101
[PE1-bfd-lsp-session-pe1tope2] discriminator remote 201
[PE1-bfd-lsp-session-pe1tope2] min-tx-interval 100
[PE1-bfd-lsp-session-pe1tope2] min-rx-interval 100
[PE1-bfd-lsp-session-pe1tope2] commit
[PE1-bfd-lsp-session-pe1tope2] quit
```

Configure PE3.

```
[PE3] bfd
[PE3-bfd] quit
[PE3] bfd pe3tope1 bind pw interface gigabitethernet 2/0/0
[PE3-bfd-lsp-session-pe3tope1] discriminator local 200
[PE3-bfd-lsp-session-pe3tope1] discriminator remote 100
[PE3-bfd-lsp-session-pe3tope1] min-tx-interval 100
[PE3-bfd-lsp-session-pe3tope1] min-rx-interval 100
[PE3-bfd-lsp-session-pe3tope1] commit
[PE3-bfd-lsp-session-pe3tope1] quit
```

Configure PE2.

```
[PE2] bfd
[PE2-bfd] quit
[PE2] bfd pe2tope1 bind pw interface gigabitethernet 2/0/0
[PE2-bfd-lsp-session-pe2tope1] discriminator local 201
[PE2-bfd-lsp-session-pe2tope1] discriminator remote 101
[PE2-bfd-lsp-session-pe2tope1] min-tx-interval 100
```

```
[PE2-bfd-lsp-session-pe2tope1] min-rx-interval 100
[PE2-bfd-lsp-session-pe2tope1] commit
[PE2-bfd-lsp-session-pe2tope1] quit
```

After completing the configuration, you can find that BFD sessions have been set up between PE1 and PE2, and between PE1 and PE3. Run the **display bfd session all** command on each PE. The command output shows that both BFD sessions are Up.

The display on PE1 is used as an example:

```
[PE1] display bfd session all
-----
Local Remote      PeerIpAddr      State    Type           InterfaceName
-----
100   200      --.--.--.--    Up      S_PW(M)       GigabitEthernet3/0/0
101   201      --.--.--.--    Up      S_PW(S)       GigabitEthernet3/0/0
-----
Total UP/DOWN Session Number : 2/0
```

Step 9 Enable the OAM mapping function.

Before enabling the OAM mapping function on the PEs, enable EFM on AC interfaces to detect link connectivity.

1. Enable EFM to detect link connectivity.

Configure PE2. The configuration on PE3 and CE2 is similar to the configuration on PE2 and is not mentioned here.

```
[PE2] efm enable
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] efm enable
[PE2-GigabitEthernet2/0/0] quit
```

After completing the configuration, run the **display efm session all** command on each device. You can find that the status of the EFM protocol on each interface is **detect**. The display on PE2 is used as an example:

```
[PE2] display efm session all
Interface          EFM State      Loopback
Timeout
-----
GigabitEthernet2/0/0  detect
--
```

2. Enable the OAM mapping function.

Configure PE2.

```
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vpn oam-mapping 3ah
[PE2-GigabitEthernet2/0/0] quit
```

Configure PE3.

```
[PE3] interface gigabitethernet 2/0/0
[PE3-GigabitEthernet2/0/0] mpls l2vpn oam-mapping 3ah
[PE3-GigabitEthernet2/0/0] quit
```

After completing the configuration, run the **display mpls l2vc oam-mapping interface** command on each PE. The command output shows information about OAM mapping. You can find that the AC OAM status is **Up**. The display on PE2 is used as an example:

```
[PE2] display mpls l2vc oam-mapping interface gigabitethernet 2/0/0
AC OAM
Info:
```

```

EOAM Type      :
802.3ah
AC OAM State   :
Up
OAM-mapping   :
Enable
PSN
info:
VC-ID         :
201
VC status     :
Primary
Active State  :
Active
Link State    :
Up
BFD for PW    :
Disable
BFD for LSP   : 0      TunnelNum: 1      PSN State :
up
  
```

Step 10 Configure association between EFM and interfaces.

Configure association between EFM and interfaces on CE2. When EFM detects a PW fault, the interface is logically Down.

Configure CE2.

```

[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] efm trigger if-down
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface gigabitethernet 2/0/0
[CE2-GigabitEthernet2/0/0] efm trigger if-down
[CE2-GigabitEthernet2/0/0] quit
  
```

Step 11 Configure the revertive switchover policy.

When the primary PW becomes faulty, GE1/0/0 on CE2 becomes Down. When the primary PW recovers, GE1/0/0 on CE2 becomes Up. If traffic is not switched back to the primary PW on PE1 immediately (delayed switchover is configured by default), data loss will occur.

Configure PE1.

```

[PE1] interface gigabitethernet 3/0/0
[PE1-GigabitEthernet3/0/0] mpls l2vpn reroute immediately
[PE1-GigabitEthernet3/0/0] quit
  
```

Step 12 Verify the configuration.

After completing the configurations, run the **display mpls l2vc interface** command on PE1. You can find that primary and secondary PWs have been set up, the VC status for both is Up, the status of the primary PW is **active**, and the status of the secondary PW is **inactive**.

```

[PE1] display mpls l2vc interface gigabitethernet 3/0/0
*client interface      : GigabitEthernet3/0/0 is up
Administrator PW      : no
session state         : up
AC status              : up
VC state               : up
Label state           : 0
Token state            : 0
VC ID                  : 200
VC type                : Ethernet
destination            : 3.3.3.9
local group ID        : 0          remote group ID      : 0
local VC label         : 1025      remote VC label       : 1024
local AC OAM State     : up
local PSN OAM State   : up
  
```

```
local forwarding state : forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state: forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : active
forwarding entry      : exist
link state             : up
local VC MTU          : 1500          remote VC MTU          : 1500
local VCCV             : cw alert ttl lsp-ping bfd
remote VCCV           : cw alert ttl lsp-ping bfd
local control word     : enable       remote control word   : enable
tunnel policy name    : p1
PW template name      : lto3
primary or secondary  : primary
load balance type     : flow
Access-port           : false
Switchover Flag       : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type       : cr lsp, TNL ID : 0x1
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 0 hours, 2 minutes, 25 seconds
up time               : 0 days, 0 hours, 0 minutes, 41 seconds
last change time      : 0 days, 0 hours, 0 minutes, 41 seconds
VC last up time       : 2013/12/20 20:13:46
VC total up time      : 0 days, 0 hours, 0 minutes, 41 seconds
CKey                  : 2
NKey                  : 1
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --

*client interface     : GigabitEthernet3/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 201
VC type               : Ethernet
destination           : 2.2.2.9
local group ID        : 0          remote group ID       : 0
local VC label        : 1026       remote VC label        : 1025
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state: forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state: forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : inactive
forwarding entry      : exist
link state             : up
```

```

local VC MTU          : 1500          remote VC MTU          : 1500
local VCCV            : cw alert ttl lsp-ping bfd
remote VCCV           : cw alert ttl lsp-ping bfd
local control word    : enable        remote control word    : enable
tunnel policy name    : --
PW template name      : lto2
primary or secondary  : secondary
load balance type     : flow
Access-port           : false
VC tunnel/token info  : 1 tunnels/tokens
  NO.0 TNL type       : lsp , TNL ID : 0x3
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 0 hours, 2 minutes, 21 seconds
up time               : 0 days, 0 hours, 1 minutes, 34 seconds
last change time      : 0 days, 0 hours, 1 minutes, 34 seconds
VC last up time       : 2013/12/20 20:12:54
VC total up time      : 0 days, 0 hours, 1 minutes, 34 seconds
CKey                  : 4
NKey                  : 3
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --

reroute policy        : immediately, resume 10 s
reason of last reroute : New LDP mapping message was received
time of last reroute  : 0 days, 0 hours, 11 minutes, 12 seconds
delay timer ID        : --          residual time :--
resume timer ID       : --          residual time :--
  
```

Run the **display ip routing-table** command on CE2. You can find that the outbound interface on CE2 for the default route is GE1/0/0. This indicates that traffic is transmitted along the primary path.

The display on CE2 is used as an example:

```

[CE2] display ip routing-table 0.0.0.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto  Pre  Cost    Flags NextHop          Interface
-----
0.0.0.0/0           Static 60   0        D   192.168.1.2
GigabitEthernet1/0/0
  
```

CE2 can ping address 192.168.3.1 of CE1 successfully.

```

[CE2] ping 192.168.3.1
PING 192.168.3.1: 56 data bytes, press CTRL_C to break
Reply from 192.168.3.1: bytes=56 Sequence=1 ttl=255 time=2 ms
Reply from 192.168.3.1: bytes=56 Sequence=2 ttl=255 time=3 ms
Reply from 192.168.3.1: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 192.168.3.1: bytes=56 Sequence=4 ttl=255 time=6 ms
Reply from 192.168.3.1: bytes=56 Sequence=5 ttl=255 time=6 ms

--- 192.168.3.1 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 2/3/6 ms
  
```

Set the status of GE1/0/0 on PE3 to Down manually.

```
[PE3] interface gigabitethernet 1/0/0
[PE3-GigabitEthernet1/0/0] shutdown
[PE3-GigabitEthernet1/0/0] quit
```

Run the **display bfd session all** command on PE1. The command output shows that the BFD session of the primary PW is Down.

```
[PE1] display bfd session all
```

Local	Remote	PeerIpAddr	State	Type	InterfaceName
100	200	--.---.---.---	Down	S_PW(M)	GigabitEthernet3/0/0
101	201	--.---.---.---	Up	S_PW(S)	GigabitEthernet3/0/0

Total UP/DOWN Session Number : 1/1

Run the **display mpls l2vc interface** command on PE1. You can find that the VC status of the primary PW changes to Down, the status of the primary PW changes to **inactive**, and the status of the secondary PW changes to **active**.

```
[PE1] display mpls l2vc interface gigabitethernet 3/0/0
*client interface      : GigabitEthernet3/0/0 is up
Administrator PW      : no
session state         : down
AC status             : up
VC state              : down
Label state           : 0
Token state           : 0
VC ID                 : 200
VC type               : Ethernet
destination           : 3.3.3.9
local group ID        : 0          remote group ID      : 0
local VC label        : 1025       remote VC label       : 0
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : not forwarding
local status code     : 0x1
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : inactive
forwarding entry      : not exist
link state             : down
local VC MTU          : 1500       remote VC MTU        : 0
local VCCV             : cw alert ttl lsp-ping bfd
remote VCCV           : none
local control word     : enable     remote control word  : none
tunnel policy name    : p1
PW template name      : lto3
primary or secondary  : primary
load balance type     : flow
Access-port           : false
Switchover Flag       : false
VC tunnel/token info  : 0 tunnels/tokens
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 0 hours, 44 minutes, 51 seconds
up time               : 0 days, 0 hours, 0 minutes, 0 seconds
last change time      : 0 days, 0 hours, 1 minutes, 27 seconds
VC last up time       : 2013/12/20 20:44:26
VC total up time      : 0 days, 0 hours, 28 minutes, 0 seconds
CKey                  : 2
NKey                  : 1
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
```

```

Domain Name           : --
*client interface     : GigabitEthernet3/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 201
VC type               : Ethernet
destination           : 2.2.2.9
local group ID        : 0
local VC label        : 1026
remote group ID       : 0
remote VC label       : 1025
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state : forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : active
forwarding entry      : exist
link state             : up
local VC MTU          : 1500
remote VC MTU         : 1500
local VCCV             : cw alert ttl lsp-ping bfd
remote VCCV           : cw alert ttl lsp-ping bfd
local control word    : enable
remote control word   : enable
tunnel policy name    : --
PW template name      : lto2
primary or secondary  : secondary
load balance type     : flow
Access-port           : false
VC tunnel/token info  : 1 tunnels/tokens
  NO.0 TNL type       : lsp , TNL ID : 0x3
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 0 hours, 44 minutes, 47 seconds
up time               : 0 days, 0 hours, 44 minutes, 0 seconds
last change time      : 0 days, 0 hours, 44 minutes, 0 seconds
VC last up time       : 2013/12/20 20:12:54
VC total up time      : 0 days, 0 hours, 44 minutes, 0 seconds
CKey                  : 4
NKey                  : 3
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --

reroute policy        : immediately, resume 10 s
reason of last reroute : New LDP mapping message was received
time of last reroute  : 0 days, 0 hours, 11 minutes, 58 seconds
delay timer ID        : -- residual time :--
resume timer ID       : -- residual time :--
  
```

Run the **display interface gigabitethernet 1/0/0** command on CE2 to check interface status. You can find that the **Line protocol current state** field is displayed as **DOWN (EFM down)**, indicating that PE3 has notified CE2 of the primary PW fault.

```

[CE2] display interface gigabitethernet 1/0/0
GigabitEthernet1/0/0 current state : UP
  
```

```
Line protocol current state : DOWN (EFM down)
Description:...
```

Check the routing table on CE2. You can find that the outbound interface of the default route has changed to GE2/0/0. This indicates that L2VPN traffic has been switched to the secondary path.

```
[CE2] display ip routing-table 0.0.0.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto  Pre  Cost           Flags NextHop         Interface
-----
0.0.0.0/0          Static 100  0              D    192.168.2.2
GigabitEthernet2/0/0
```

Remove the fault on GE1/0/0 on PE3 manually.

```
[PE3] interface gigabitethernet 1/0/0
[PE3-GigabitEthernet1/0/0] undo shutdown
[PE3-GigabitEthernet1/0/0] quit
```

Check the routing table on CE2. You can find that the outbound interface of the default route has changed to GE1/0/0. This indicates that L2VPN traffic has been switched back to the primary path.

```
[CE2] display ip routing-table 0.0.0.0
Route Flags: R - relay,
D - download to fib
-----
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto  Pre  Cost           Flags NextHop         Interface
-----
0.0.0.0/0          Static  60  0              D    192.168.1.2
GigabitEthernet1/0/0
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 192.168.1.2 255.255.255.0
 ip address 192.168.2.2 255.255.255.0 sub
#
interface GigabitEthernet2/0/0
 ip address 192.168.3.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
bfd
#
mpls lsr-id 1.1.1.9
mpls
mpls te
mpls rsvp-te
mpls te cspf
```

```
#
mpls l2vpn
#
pw-template lto2
  peer-address 2.2.2.9
  control-word
#
pw-template lto3
  peer-address 3.3.3.9
  control-word
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
  remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
  ip address 172.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip address 172.2.1.1 255.255.255.0
  mpls
  mpls te
  mpls rsvp-te
#
interface GigabitEthernet3/0/0
  mpls l2vc pw-template lto3 200 tunnel-policy p1
  mpls l2vc pw-template lto2 201 secondary
  mpls l2vpn reroute immediately resume 10
#
interface LoopBack1
  ip address 1.1.1.9 255.255.255.255
#
interface Tunnel0/0/1
  ip address unnumbered interface LoopBack1
  tunnel-protocol mpls te
  destination 3.3.3.9
  mpls te tunnel-id 100
  mpls te commit
#
ospf 1
  opaque-capability enable
  area 0.0.0.0
    network 1.1.1.9 0.0.0.0
    network 172.1.1.0 0.0.0.255
    network 172.2.1.0 0.0.0.255
  mpls-te enable
#
tunnel-policy p1
  tunnel select-seq cr-lsp load-balance-number 1
#
bfd peltop2 bind pw interface GigabitEthernet3/0/0
  secondary
  discriminator local
  101
  discriminator remote
  201
  min-tx-interval 100
  min-rx-interval 100

commit

#
bfd peltop3 bind pw interface
GigabitEthernet3/0/0
  discriminator local
```

```
100
 discriminator remote
200
 min-tx-interval 100
 min-rx-interval 100
 commit
#
return
```

- Configuration file of the P device

```
#
sysname P
#
mpls lsr-id 4.4.4.9
mpls
 mpls te
 mpls rsvp-te
#
interface GigabitEthernet1/0/0
 ip address 172.2.1.2 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface GigabitEthernet2/0/0
 ip address 172.3.1.1 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface LoopBack1
 ip address 4.4.4.9 255.255.255.255
#
ospf 1
 opaque-capability enable
 area 0.0.0.0
  network 4.4.4.9
0.0.0.0
  network 172.2.1.0
0.0.0.255
  network 172.3.1.0 0.0.0.255
  mpls-te enable
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
efm enable
#
bfd
#
mpls lsr-id 2.2.2.9
mpls
#
mpls l2vpn
#
pw-template 2to1
 peer-address 1.1.1.9
 control-word
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
```

```
mpls l2vc pw-template 2to1
201
mpls l2vpn oam-mapping
3ah
efm enable
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
bfd pe2tope1 bind pw interface GigabitEthernet2/0/0
discriminator local
201
discriminator remote
101
min-tx-interval 100
min-rx-interval 100

commit

#
return
```

● Configuration file of PE3

```
#
sysname PE3
#
efm enable
#
bfd
#
mpls lsr-id 3.3.3.9
mpls
mpls te
mpls rsvp-te
mpls te cspf
#
mpls l2vpn
#
pw-template 3to1
peer-address 1.1.1.9
control-word
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
remote-ip 1.1.1.9
#
interface GigabitEthernet1/0/0
ip address 172.3.1.2 255.255.255.0
mpls
mpls te
mpls rsvp-te
#
interface GigabitEthernet2/0/0
mpls l2vc pw-template 3to1 200 tunnel-policy
p1
mpls l2vpn oam-mapping
3ah
efm enable
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
interface Tunnel0/0/1
ip address unnumbered interface LoopBack1
```

```
tunnel-protocol mpls te
destination 1.1.1.9
mpls te tunnel-id 101
mpls te commit
#
ospf 1
opaque-capability enable
area 0.0.0.0
network 3.3.3.9
0.0.0.0
network 172.3.1.0 0.0.0.255
mpls-te enable
#
tunnel-policy p1
tunnel select-seq cr-lsp load-balance-number 1
#
bfd pe3tope1 bind pw interface GigabitEthernet2/0/0
discriminator local
200
discriminator remote
100
min-tx-interval 100
min-rx-interval 100

commit

#
return
```

● Configuration file of CE2

```
#
sysname CE2
#
efm enable
#
interface GigabitEthernet1/0/0
ip address 192.168.1.3 255.255.255.0
efm enable
efm trigger if-down
#
interface GigabitEthernet2/0/0
ip address 192.168.2.3 255.255.255.0
efm enable
efm trigger if-down
#
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet1/0/0 192.168.1.2
ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet2/0/0 192.168.2.2 preference
100
#
return
```

11.9.6 Example for Configuring VLL to Use a GRE Tunnel

Networking Requirements

 **NOTE**

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S cannot be used in this scenario.

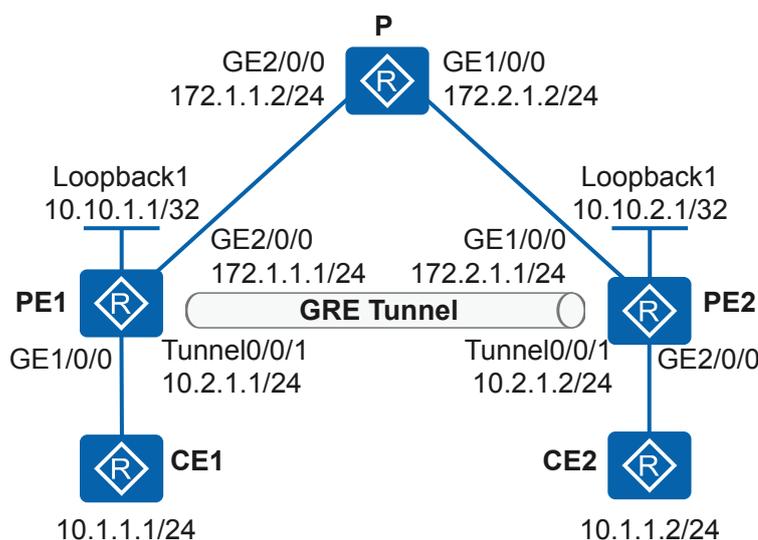
An ISP network provides the L2VPN service for users. Many users connect to the MPLS network through PE1 and PE2, and users on the PEs change frequently. A proper VPN solution is required to provide secure VPN services for users and to simplify configuration when new users connect to the network.

A Martini VLL connection can be set up between CE1 and CE2 to meet these requirements. By default, the system uses Label Switched Paths (LSPs) for Martini VLL, and does not

perform load balancing. When the P does not provide MPLS functions, VLL cannot be implemented.

To solve the problem, apply a tunnel policy to Martini VLL to specify that VLL services are transmitted over a GRE tunnel.

Figure 11-24 Networking diagram for configuring VLL to use a GRE tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on the PE and P devices on the backbone network to ensure reachability between them.
2. Enable MPLS and MPLS LDP on PEs. Set up a remote LDP session between the PEs to exchange VC labels between the PEs.
3. Enable MPLS L2VPN on PEs. Enabling MPLS L2VPN is the prerequisite for VLL configuration.
4. Create GRE tunnel interfaces on PEs and establish a GRE tunnel between PEs.
5. Create VC connections on PEs. Because the P does not support MPLS functions, configure a tunnel policy and apply it when you create VC connections so that VLL services can be transmitted over a GRE tunnel.

Procedure

Step 1 Configure interface IP addresses and a routing protocol on the PEs and P.

Configure PE1. The configurations of PE2 and P are similar to the configuration of PE1, and are not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
```

```
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 10.10.1.1 255.255.255.255
[PE1-LoopBack1] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 10.10.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configurations are complete, OSPF neighbor relationships can be set up between PE1, P, and PE2. Run the **display ospf peer** command. You can see that the neighbor status is **Full**. Run the **display ip routing-table** command. You can see that PEs have learnt the routes to Loopback1 of each other.

Step 2 Configure basic MPLS functions and LDP on PEs and establish a remote LDP session between PEs.

Configure PE1.

```
[PE1] mpls lsr-id 10.10.1.1
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] mpls ldp remote-peer 10.10.2.1
[PE1-mpls-ldp-remote-10.10.2.1] remote-ip 10.10.2.1
[PE1-mpls-ldp-remote-10.10.2.1] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 10.10.2.1
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] mpls ldp remote-peer 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] remote-ip 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] quit
```

After the configurations are complete, run the **display mpls ldp session** command on PE1 to view the LDP session status. You can see that an LDP session is set up between PE1 and PE2.

The display on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
10.10.2.1:0           Operational DU    Passive  0000:00:01  1/1
-----
TOTAL: 1 session(s) Found.
```

Step 3 Enable MPLS L2VPN on PEs.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
```

Step 4 Create GRE tunnel interfaces on PEs and establish a GRE tunnel between PEs.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip address 10.2.1.1 255.255.255.0
[PE1-Tunnel0/0/1] tunnel-protocol gre
[PE1-Tunnel0/0/1] source 10.10.1.1
[PE1-Tunnel0/0/1] destination 10.10.2.1
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] ip address 10.2.1.2 255.255.255.0
[PE2-Tunnel0/0/1] tunnel-protocol gre
[PE2-Tunnel0/0/1] source 10.10.2.1
[PE2-Tunnel0/0/1] destination 10.10.1.1
[PE2-Tunnel0/0/1] quit
```

After the configurations are complete, the tunnel interfaces go Up and can ping each other.

The display on PE1 is used as an example.

```
[PE1] ping -a 10.2.1.1 10.2.1.2
PING 10.2.1.2: 56 data bytes, press CTRL_C to break
Reply from 10.2.1.2: bytes=56 Sequence=1 ttl=255 time=1 ms
Reply from 10.2.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
Reply from 10.2.1.2: bytes=56 Sequence=3 ttl=255 time=1 ms
Reply from 10.2.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
Reply from 10.2.1.2: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.2.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/1/1 ms
```

Step 5 Configure a tunnel policy, create VC connections, and apply the policy to the VC connections so that VLL services can be transmitted over a GRE tunnel.

Configure PE1.

```
[PE1] tunnel-policy gre1
[PE1-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE1-tunnel-policy-gre1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 10.10.2.1 39 tunnel-policy gre1
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] tunnel-policy gre1
[PE2-tunnel-policy-gre1] tunnel select-seq gre load-balance-number 1
[PE2-tunnel-policy-gre1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc 10.10.1.1 39 tunnel-policy gre1
[PE2-GigabitEthernet2/0/0] quit
```

Step 6 Verify the configuration.

After the configurations are complete, check the L2VPN connection on PEs. You can see that an L2VC connection has been set up and is in Up state.

The display on PE1 is used as an example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
```

```

Administrator PW      : no
session state        : up
AC status            : up
VC state             : up
Label state          : 0
Token state          : 0
VC ID                : 39
VC type              : Ethernet
destination           : 10.10.2.1
local group ID       : 0          remote group ID      : 0
local VC label       : 1025      remote VC label       : 1024
local AC OAM State   : up
local PSN OAM State  : up
local forwarding state : forwarding
local status code    : 0x0
remote AC OAM state  : up
remote PSN OAM state : up
remote forwarding state : forwarding
remote status code   : 0x0
ignore standby state : no
BFD for PW           : unavailable
VCCV State           : up
manual fault         : not set
active state         : active
forwarding entry     : exist
link state           : up
local VC MTU         : 1500      remote VC MTU         : 1500
local VCCV           : alert ttl lsp-ping bfd
remote VCCV          : alert ttl lsp-ping bfd
local control word   : disable   remote control word   : disable
tunnel policy name   : gre1
PW template name     : --
primary or secondary : primary
load balance type    : flow
Access-port          : false
Switchover Flag      : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type      : gre , TNL ID : 0x2
  Backup TNL type    : lsp , TNL ID : 0x0
create time          : 0 days, 2 hours, 37 minutes, 1 seconds
up time              : 0 days, 0 hours, 2 minutes, 11 seconds
last change time     : 0 days, 0 hours, 2 minutes, 11 seconds
VC last up time      : 2013/02/20 18:58:24
VC total up time     : 0 days, 2 hours, 35 minutes, 58 seconds
CKey                 : 2
NKey                 : 1
PW redundancy mode   : frr
AdminPw interface    : --
AdminPw link state   : --
Diffserv Mode        : uniform
Service Class        : --
Color                : --
DomainId             : --
Domain Name          : --
  
```

Run the **display tunnel-info tunnel-id** command on PEs according to the tunnel ID in the preceding command output. You can view details of the specified tunnel ID.

```

[PE1] display tunnel-info tunnel-id 2
Tunnel ID:          0x2
Tunnel Token:       2
Type:               gre
Destination:        10.10.2.1
Out Slot:           0
Instance ID:        0
Interface:          Tunnel0/0/1
  
```

CE1 and CE2 can ping each other successfully.

The display on CE1 is used as an example.

```
[CE1] ping 10.1.1.2
PING 10.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=31 ms
  Reply from 10.1.1.2: bytes=56 Sequence=2 ttl=255 time=10 ms
  Reply from 10.1.1.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.1.1.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.1.1.2: bytes=56 Sequence=5 ttl=255 time=28 ms
--- 10.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 2/15/31 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 10.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 10.10.1.1
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 10.10.2.1
remote-ip 10.10.2.1
#
interface GigabitEthernet1/0/0
mpls l2vc 10.10.2.1 39 tunnel-policy gre1
#
interface GigabitEthernet2/0/0
ip address 172.1.1.1 255.255.255.0
#
interface LoopBack1
ip address 10.10.1.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.2.1.1 255.255.255.0
tunnel-protocol gre
source 10.10.1.1
destination 10.10.2.1
#
ospf 1
area 0.0.0.0
network 10.10.1.1 0.0.0.0
network 172.1.1.0 0.0.0.255
#
tunnel-policy gre1
tunnel select-seq gre load-balance-number 1
#
return
```

- Configuration file of P

```
#
sysname P
```

```
#
interface GigabitEthernet2/0/0
ip address 172.1.1.2 255.255.255.0
#
interface GigabitEthernet1/0/0
ip address 172.2.1.2 255.255.255.0
#
ospf 1
area 0.0.0.0
network 172.1.1.0 0.0.0.255
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 10.10.2.1
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 10.10.1.1
remote-ip 10.10.1.1
#
interface GigabitEthernet1/0/0
ip address 172.2.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
mpls l2vc 10.10.1.1 39 tunnel-policy gre1
#
interface LoopBack1
ip address 10.10.2.1 255.255.255.255
#
interface Tunnel0/0/1
ip address 10.2.1.2 255.255.255.0
tunnel-protocol gre
source 10.10.2.1
destination 10.10.1.1
#
ospf 1
area 0.0.0.0
network 10.10.2.1 0.0.0.0
network 172.2.1.0 0.0.0.255
#
tunnel-policy gre1
tunnel select-seq gre load-balance-number 1
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.1.1.2 255.255.255.0
#
return
```

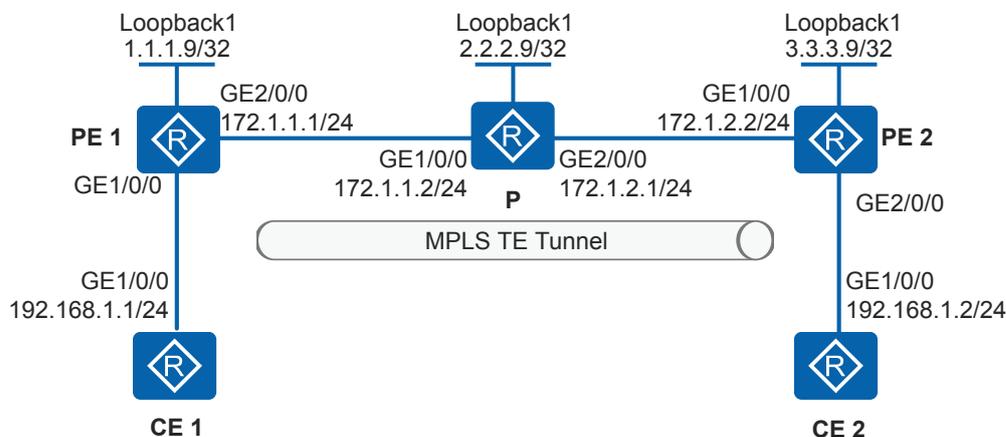
11.9.7 Example for Configuring a VLL Using an MPLS TE Tunnel

Networking Requirements

As shown in [Figure 11-25](#), the MPLS network of an ISP provides the L2VPN service for users. Many users connect to the MPLS network through PE1 and PE2, and users connected

to the PE devices change frequently. A proper VPN solution is required to provide secure VPN services for users and to simplify configuration when new users connect to the network.

Figure 11-25 Networking for configuring a Martini VLL



Configuration Roadmap

As the number of access users is large and the users frequently change, a VLL in Martini mode is recommended between the PEs to simplify configuration when new users connect to the network. To ensure reliable transmission of VPN services, the highly reliable MPLS TE tunnel is recommended as the public network tunnel.

The configuration roadmap is as follows:

1. Assign an IP address to each interface, and configure an IGP on the PE and P devices on the backbone network to implement interworking between the devices.
2. Create an MPLS TE tunnel and configure a tunnel policy to transmit VLL data.
3. Create a VLL in Martini mode between the PEs to simplify configuration when new users connect to the network. Create a remote LDP session between the PEs to transmit local VC labels to the remote device. Create a VC connection between the PEs and apply the tunnel policy to select the MPLS TE tunnel.

Procedure

Step 1 Configure an IP address and routing protocol for each interface.

Assign IP addresses to interfaces, and configure an IGP on the PE and P devices of the backbone network according to [Figure 11-25](#) to implement interworking between the devices.

Configure CE1. The configuration on CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 192.168.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Step 2 Set up an MPLS TE tunnel and create a tunnel binding policy.

- Enable MPLS, MPLS TE, and RSVP-TE globally on PE1, P, and PE2, and on all interfaces along the tunnel. Enable CSPF on the ingress of the tunnel.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] mpls te
[PE1-mpls] mpls rsvp-te
[PE1-mpls] mpls te cspf
[PE1-mpls] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls te
[PE1-GigabitEthernet2/0/0] mpls rsvp-te
[PE1-GigabitEthernet2/0/0] quit
```

Configure the P device.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] mpls te
[P-mpls] mpls rsvp-te
[P-mpls] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls te
[P-GigabitEthernet1/0/0] mpls rsvp-te
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls te
[P-GigabitEthernet2/0/0] mpls rsvp-te
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] mpls te
[PE2-mpls] mpls rsvp-te
[PE2-mpls] mpls te cspf
[PE2-mpls] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls te
[PE2-GigabitEthernet1/0/0] mpls rsvp-te
[PE2-GigabitEthernet1/0/0] quit
```

- Configure OSPF TE on the MPLS backbone network to advertise TE information.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] opaque-capability enable
```

```
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] mpls-te enable
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

- Configure tunnel interfaces for the MPLS TE tunnel.

On the ingress of the tunnel, create a tunnel interface and set the IP address, tunnel protocol, destination IP address, tunnel ID. Then, run the **mpls te commit** command to commit the configuration.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE1-Tunnel0/0/1] tunnel-protocol mpls te
[PE1-Tunnel0/0/1] destination 3.3.3.9
[PE1-Tunnel0/0/1] mpls te tunnel-id 100
[PE1-Tunnel0/0/1] mpls te commit
[PE1-Tunnel0/0/1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] ip address unnumbered interface loopback 1
[PE2-Tunnel0/0/1] tunnel-protocol mpls te
[PE2-Tunnel0/0/1] destination 1.1.1.9
[PE2-Tunnel0/0/1] mpls te tunnel-id 100
[PE2-Tunnel0/0/1] mpls te commit
[PE2-Tunnel0/0/1] quit
```

After the configuration is complete, run the **display mpls te tunnel-interface** command on the PE devices at both ends of the tunnel. The command output shows that an MPLS TE tunnel is set up successfully. The command output on PE1 is used as an example.

```
[PE1] display mpls te tunnel-interface
-----
                                Tunnel0/0/1
-----
Tunnel State Desc      : UP
Active LSP             : Primary LSP
Session ID             : 100
Ingress LSR ID        : 1.1.1.9           Egress LSR ID: 3.3.3.9
Admin State           : UP                Oper State   : UP
Primary LSP State     : UP
Main LSP State        : READY             LSP ID      : 1
```

- Configure a tunnel binding policy.

Configure PE1.

```
[PE1] interface tunnel 0/0/1
[PE1-Tunnel0/0/1] mpls te reserved-for-binding
[PE1-Tunnel0/0/1] mpls te commit
[PE1-Tunnel0/0/1] quit
[PE1] tunnel-policy 1
[PE1-tunnel-policy-1] tunnel binding destination 3.3.3.9 te tunnel 0/0/1
[PE1-tunnel-policy-1] quit
```

Configure PE2.

```
[PE2] interface tunnel 0/0/1
[PE2-Tunnel0/0/1] mpls te reserved-for-binding
[PE2-Tunnel0/0/1] mpls te commit
[PE2-Tunnel0/0/1] quit
[PE2] tunnel-policy 1
[PE2-tunnel-policy-1] tunnel binding destination 1.1.1.9 te tunnel 0/0/1
[PE2-tunnel-policy-1] quit
```

Step 3 Create a remote LDP session between PE1 and PE2.

Configure PE1.

```
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure PE2.

```
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] mpls ldp remote-peer 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] quit
```

After the configuration is complete, run the **display mpls ldp session** command on PE1 to view the LDP session status. The command output shows that the LDP session status is **Operational**, indicating that a remote LDP session is established between PE1 and PE2.

The command output on PE1 is used as an example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID           Status          LAM  SsnRole  SsnAge         KASent/Rcv
-----
3.3.3.9:0        Operational    DU   Passive  0000:00:00     1/1
-----
TOTAL: 1 session(s) Found.
```

Step 4 Create a VC connection between the PE devices, and apply a tunnel binding policy to the connection.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 3.3.3.9 101 tunnel-policy 1
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] mpls l2vc 1.1.1.9 101 tunnel-policy 1
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

Check the L2VPN connections on the PE devices. You can see that an L2VC is set up and is in Up state.

The command output on PE1 is used as an example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 101
VC type               : Ethernet
```

```
destination          : 3.3.3.9
local group ID       : 0           remote group ID       : 0
local VC label       : 1026        remote VC label       : 1032
local AC OAM State   : up
local PSN OAM State  : up
local forwarding state : forwarding
local status code    : 0x0
remote AC OAM state  : up
remote PSN OAM state : up
remote forwarding state : forwarding
remote status code   : 0x0
ignore standby state : no
BFD for PW           : unavailable
VCCV State           : up
manual fault         : not set
active state         : active
forwarding entry     : exist
link state           : up
local VC MTU         : 1500        remote VC MTU         : 1500
local VCCV           : alert ttl lsp-ping bfd
remote VCCV          : alert ttl lsp-ping bfd
local control word   : disable     remote control word   : disable
tunnel policy name   : 1
PW template name     : --
primary or secondary : primary
load balance type    : flow
Access-port          : false
Switchover Flag      : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type      : cr lsp, TNL ID : 0x1
  Backup TNL type    : lsp , TNL ID : 0x0
create time          : 0 days, 4 hours, 16 minutes, 25 seconds
up time              : 0 days, 4 hours, 15 minutes, 58 seconds
last change time     : 0 days, 4 hours, 15 minutes, 58 seconds
VC last up time      : 2013/09/16 09:57:04
VC total up time     : 0 days, 4 hours, 15 minutes, 58 seconds
CKey                 : 4
NKey                 : 3
PW redundancy mode   : frr
AdminPw interface    : --
AdminPw link state   : --
Diffserv Mode        : uniform
Service Class        : --
Color                : --
DomainId             : --
Domain Name          : --
```

CE1 and CE2 can ping each other.

The command output on CE1 is used as an example.

```
[CE1] ping 192.168.1.2
PING 192.168.1.2: 56 data bytes, press CTRL_C to break
Reply from 192.168.1.2: bytes=56 Sequence=1 ttl=255 time=10 ms
Reply from 192.168.1.2: bytes=56 Sequence=2 ttl=255 time=1 ms
Reply from 192.168.1.2: bytes=56 Sequence=3 ttl=255 time=10 ms
Reply from 192.168.1.2: bytes=56 Sequence=4 ttl=255 time=1 ms
Reply from 192.168.1.2: bytes=56 Sequence=5 ttl=255 time=10 ms

--- 192.168.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/6/10 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 192.168.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
 mpls te
 mpls rsvp-te
 mpls te cspf
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
 remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
 mpls l2vc 3.3.3.9 101 tunnel-policy 1
#
interface GigabitEthernet2/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
interface Tunnel0/0/1
 ip address unnumbered interface LoopBack1
 tunnel-protocol mpls te
 destination 3.3.3.9
 mpls te tunnel-id 100
 mpls te reserved-for-binding
 mpls te commit
#
ospf 1
 opaque-capability enable
 area 0.0.0.0
 network 1.1.1.9 0.0.0.0
 network 172.1.1.0 0.0.0.255
 mpls-te enable
#
tunnel-policy 1
 tunnel binding destination 3.3.3.9 te Tunnel0/0/1
#
return
```

- Configuration file of the P device

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
 mpls te
 mpls rsvp-te
#
```

```
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface GigabitEthernet2/0/0
 ip address 172.1.2.1 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
ospf 1
 opaque-capability enable
 area 0.0.0.0
 network 2.2.2.9 0.0.0.0
 network 172.1.1.0 0.0.0.255
 network 172.1.2.0 0.0.0.255
 mpls-te enable
#
return
```

● Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 3.3.3.9
mpls
 mpls te
 mpls rsvp-te
 mpls te cspf
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
 remote-ip 1.1.1.9
#
interface GigabitEthernet1/0/0
 ip address 172.1.2.2 255.255.255.0
 mpls
 mpls te
 mpls rsvp-te
#
interface GigabitEthernet2/0/0
 mpls l2vc 1.1.1.9 101 tunnel-policy 1
#
interface LoopBack1
 ip address 3.3.3.9 255.255.255.255
#
interface Tunnel0/0/1
 ip address unnumbered interface LoopBack1
 tunnel-protocol mpls te
 destination 1.1.1.9
 mpls te tunnel-id 100
 mpls te reserved-for-binding
 mpls te commit
#
ospf 1
 opaque-capability enable
 area 0.0.0.0
 network 3.3.3.9 0.0.0.0
 network 172.1.2.0 0.0.0.255
 mpls-te enable
#
tunnel-policy 1
```

```
tunnel binding destination 1.1.1.9 te Tunnel10/0/1
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 192.168.1.2 255.255.255.0
#
return
```

11.10 Troubleshooting VLL

This section describes the common configuration errors and troubleshooting methods.

11.10.1 The VC of a Martini VLL Connection Cannot Go Up

Fault Description

After Martini VLL is configured, the local end cannot ping the remote end. The VC status is Down.

Procedure

Step 1 Check whether the two ends use the same encapsulation type and MTU value.

Run the **display mpls l2vc vc-id** command to check VC information.

```
<Huawei> display mpls l2vc 102
Total LDP VC : 1      0 up      1 down

*client interface      : Vlanif10 is up
Administrator PW      : no
session state         : down
AC status             : up
VC state              : down
Label state           : 0
Token state           : 0
VC ID                 : 102
VC type               : VLAN
destination           : 2.2.2.2
local VC label        : 1032      remote VC label      : 0
control word          : disable
remote control word   : none
forwarding entry      : not exist
local group ID        : 0
remote group ID       : 0
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : not forwarding
local status code     : 0x1
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state           : inactive
link state             : down
local VC MTU          : 1500      remote VC MTU        : 0
local VCCV             : alert ttl lsp-ping bfd
remote VCCV           : none
tunnel policy name    : --
PW template name      : --
```

```
primary or secondary : primary
.....
```

If the two PEs use different encapsulation types or MTU values, configure the same type of AC interfaces on the PEs and run the **mpls mtu** command to set the same MTU value on the PEs.

If the two PEs use the same encapsulation type and MTU but the fault persists, go to step 2.

 **NOTE**

A VC can go Up only when both ends use the same encapsulation type and MTU value.

Step 2 Check whether VC IDs on the two PEs are the same.

```
<Huawei> display mpls l2vc 102
Total LDP VC : 1      0 up      1 down

*client interface      : Vlanif10 is up
Administrator PW      : no
session state         : up
AC status              : up
VC state               : down
Label state           : 0
Token state           : 0
VC ID                  : 102
VC type                : VLAN
.....
```

If the VC IDs are different, run the **undo mpls l2vc** command on one end to delete the existing VC ID, and then run the **mpls l2vc** command to set the VC ID to the same as that on the other end.

If the two ends use the same VC ID but the fault persists, go to step 3.

 **NOTE**

A VC can go Up only when both ends use the same VC ID.

Step 3 Check whether the control words on the two PEs are the same.

```
<Huawei> display mpls l2vc 102
Total LDP VC : 1      0 up      1 down

*client interface      : Vlanif10 is up
Administrator PW      : no
session state         : up
AC status              : up
VC state               : down
Label state           : 0
Token state           : 0
VC ID                  : 102
VC type                : VLAN
destination            : 2.2.2.2
local VC label         : 1032      remote VC label      : 1500
control word           : disable
remote control word    : none
forwarding entry       : not exist
local group ID         : 0
remote group ID        : 0
local AC OAM State     : up
local PSN OAM State    : up
local forwarding state : not forwarding
local status code      : 0x1
BFD for PW             : unavailable
VCCV State             : up
manual fault           : not set
active state           : inactive
link state             : down
```

```
local VC MTU      : 1500      remote VC MTU      : 0
local VCCV        : alert ttl lsp-ping bfd
remote VCCV       : none
tunnel policy name : --
PW template name  : --
primary or secondary : primary
.....
```

A VC can go Up only when both ends use the same control word. If the control words on two ends are different, run the **undo mpls l2vc** command to delete a VC connection on one end and then run the **mpls l2vc** command to create a VC connection and configure the same control word for two ends.

----End

11.11 References for VLL

This section lists the references for VLL.

The following table lists the references for VLL.

Document	Description	Remarks
RFC 4664	Framework for Layer 2 Virtual Private Networks (L2VPNs)	-
RFC 4665	Service Requirements for Layer-2 Provider Provisioned Virtual Private Networks	-
RFC 6074	Provisioning Models and Endpoint Identifiers in L2VPN Signaling	-

12 PWE3 Configuration

About This Chapter

This chapter describes principles, applications, and configurations of the Pseudo-Wire Emulation Edge to Edge (PWE3).

[12.1 Overview of PWE3](#)

This section describes the PWE3 definition, purpose, and functions.

[12.2 Relationship Between PWE3 and L2VPN](#)

PWE3 is a set of point-to-point (P2P) Layer 2 Virtual Private Network (L2VPN) technologies. Martini L2VPN is only one of the PWE3 technologies. PWE3 uses some Martini L2VPN techniques, including Label Distribution Protocol (LDP) signaling and encapsulation modes. In addition, PWE3 extends the Martini L2VPN.

[12.3 Understanding PWE3](#)

This section describes the implementation of PWE3.

[12.4 Application Scenarios for PWE3](#)

This section describes application scenarios for PWE3.

[12.5 Summary of PWE3 Configuration Tasks](#)

When configuring PWE3, you can configure static PW, dynamic PW, TDM PWE3, or PW switching. If a network spans multiple autonomous systems (ASs), you need to configure inter-AS PWE3. When a reliability solution is required, you also need to perform other configurations, such as Bidirectional Forwarding Detection (BFD) for PW and PWE3 fast reroute (FRR).

[12.6 Licensing Requirements and Limitations for PWE3](#)

[12.7 Default Settings for PWE3](#)

[12.8 Configuring PWE3](#)

This section describes how to configure PWE3 functions in details.

[12.9 Maintaining PWE3](#)

This section describes how to maintain PWE3, including verifying connectivity of a PW and locating a fault on a PW.

[12.10 Configuration Examples for PWE3](#)

This section describes PWE3 configuration examples including the networking requirements, and configuration roadmap, configuration procedure, and configuration files.

12.11 References for PWE3

This section provides references for PWE3.

12.1 Overview of PWE3

This section describes the PWE3 definition, purpose, and functions.

Definition

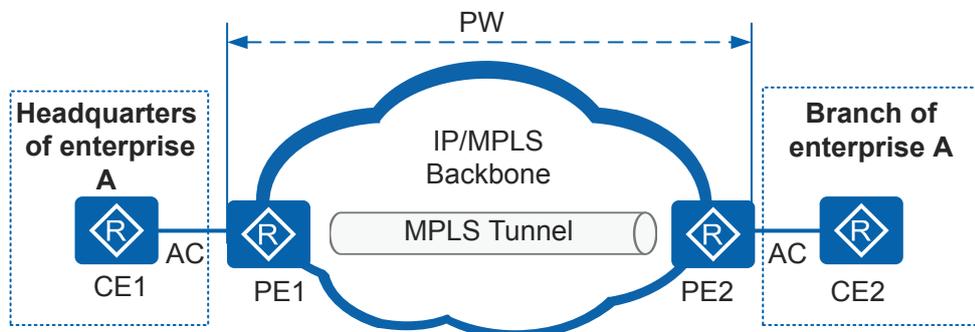
Pseudo-Wire Emulation Edge to Edge (PWE3) is a point-to-point (P2P) technology that transmits Layer 2 services on a multiprotocol label switching Layer 2 virtual private network (MPLS L2VPN). PWE3 simulates essential attributes of a service such as Asynchronous Transfer Mode (ATM), Frame Relay (FR), Ethernet, low-speed Time Division Multiplexing (TDM) circuit, Synchronous Optical Network (SONET), or Synchronous Digital Hierarchy (SDH) on a Packet Switched Network (PSN). PWE3 complies with RFC 4447 (only FEC 128 is supported currently), which is developed based on draft-martini-l2circuit-trans-mpls.

Purpose

The IP network has rapidly developed in recent years because of its flexible upgrade, scalability, and interoperability. Limited by the transmission mode and services, the traditional communications network has low flexibility. To make full use of existing or public network resources during upgrade and expansion of the traditional communications network, PWE3 is used to integrate the traditional communications network and PSN.

PWE3 often applies to the broadband metro access network (MAN) or mobile bearer network to transmit various services including Ethernet, ATM, TDM, FR, and PPP. As shown in [Figure 12-1](#), the headquarters of company A and its branch are located on the traditional communications network such as ATM or FR. PWE3 is used to establish a PW between PE1 and PE2 so that the headquarters of company A and its branch can communicate over the MPLS network. PWE3 integrates original access modes with existing IP backbone network to reduce repetitious network construction, saving operation costs.

Figure 12-1 PWE3 networking



PWE3 allows various services to be transmitted and supports migration from the mobile network to Long Term Evolution (LTE). PWE3 protects carrier investments when ATM and TDM services are migrated to the IP network.

12.2 Relationship Between PWE3 and L2VPN

PWE3 is a set of point-to-point (P2P) Layer 2 Virtual Private Network (L2VPN) technologies. Martini L2VPN is only one of the PWE3 technologies. PWE3 uses some Martini L2VPN techniques, including Label Distribution Protocol (LDP) signaling and encapsulation modes. In addition, PWE3 extends the Martini L2VPN.

PWE3 is extended Martini and has the same signaling process as Martini.

12.2.1 Extensions to the Control Plane

Signaling Extension

PWE3 advertises the PW status using LDP signaling Notification messages. A PW is torn down only when PW configurations are deleted or the LDP session is interrupted. PWE3 reduces control packets exchanged between PEs and signaling costs. LDP signaling used by PWE3 is compatible with common LDP and Martini.

Multi-Segment Extension

Multi-segment PWE3 extends networking modes.

- Multi-segment PWE3 has low requirements for the number of LDP sessions supported by an access device. That is, the costs of LDP sessions on the access device are reduced.
- The access node on a multi-segment PW provides PW aggregation. This allows for more flexible networking and easily divides a network into access, aggregation, and core layers.

TDM Interface Extension

PWE3 supports more low-speed Time Division Multiplexing (TDM) interfaces. PWE3 supports packet sequencing, clock extraction, and clock synchronization using the Real-time Transport Protocol (RTP) on the forwarding plane and the control word.

PWE3 has the following advantages by supporting low-speed TDM interfaces:

- Adds encapsulation types (packets on low-speed TDM interfaces can be encapsulated).
- Integrates the PSTN, TV, and data networks.
- Substitutes the traditional Digital Data Network (DDN) service.

Other Extensions

PWE3 has the following extensions at the control plane:

- Fragmentation negotiation mechanism
- PW connectivity detection, such as VCCV and OAM, which speeds up network convergence and enhances network reliability

12.2.2 Extensions at the Data Plane

- Extensions of the real-time information

- Importing RTP for clock extraction and time synchronization
- Guaranteeing the bandwidth, jitter, and delay of telecom signals
- Re-transmission of disordered packets

12.3 Understanding PWE3

This section describes the implementation of PWE3.

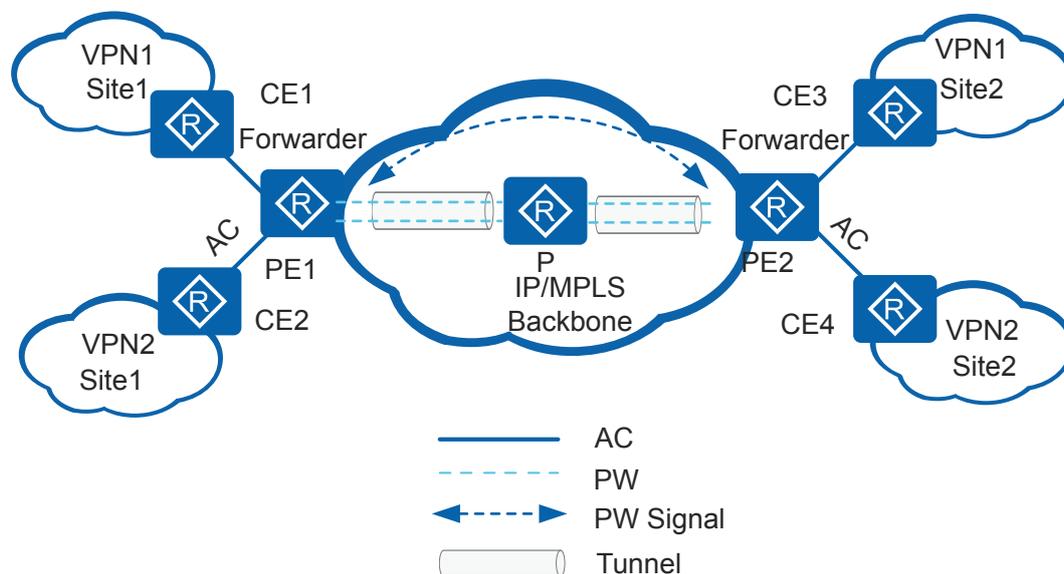
12.3.1 Implementation

PWE3 Architecture

PWE3 uses the Label Distribution Protocol (LDP) as the signaling protocol, and transmits Layer 2 packets of Customer Edges (CEs) through tunnels such as MPLS LSPs, Generic Routing Encapsulation (GRE) tunnels, or multiprotocol label switching traffic engineering (MPLS TE) tunnels. As shown in [Figure 12-2](#), PWE3 uses the following entities:

- Attachment circuit (AC)
- Pseudo wire (PW)
- Forwarder
- Tunnels
- PW signaling protocol

Figure 12-2 PWE3 architecture



The following uses the flow direction of VPN1 packets from CE1 to CE3 as an example to show the basic direction of data flows.

- CE1 sends Layer 2 packets to PE1 through an AC.
- After PE1 receives the packets, the forwarder selects a PW to forward the packets.

- PE1 generates double MPLS labels (private and public network labels) based on the forwarding entry of the PW. The private network label is used to identify the PW, and the public network label identifies the tunnel to PE2 on the public network.
- After Layer 2 packets arrive at PE2 through the tunnel on the public network, the Penultimate Hop Popping (PHP) device (a P device) pops out the public network label, and PE2 pops out the private network label.
- The forwarder of PE2 selects an AC to forward the Layer 2 packets to CE3.

PWE3 Classification

PWs are classified into the following types:

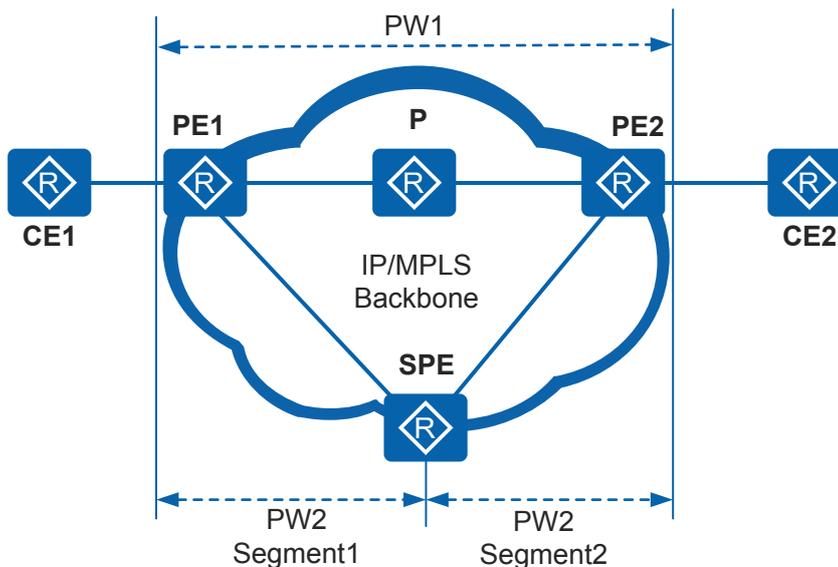
- Static PW and dynamic PW in terms of implementation
 Martini VLL uses LDP to establish dynamic PWs. PWE3 also allows static PWs established using the manually configured PW information.
- Single-segment PW and multi-segment PW in terms of networking modes
 - Single-segment PW: Only one PW is set up between two PEs, and PW label switching is not required. For example, PW1 in Figure 12-3 is a single-segment PW.
 - Multi-segment PW: Multiple segments of the PW exist between two PEs. The forwarding mechanism of PEs on the multi-segment PW is the same as that of the single-segment PW. The difference is that PW labels need to be switched on the Switching PE (SPE). For example, PW2 in Figure 12-3 is a multi-segment PW.

NOTE

A multi-segment PW is required if a signaling connection or directly connected tunnel cannot be established between two PEs. With the multi-segment PW, PWE3 makes networking flexible.

The two classification modes are independent of each other. PWE3 supports the mixed PW, that is, a static PW at one end and a dynamic PW at the other end.

Figure 12-3 Single-segment PW and multi-segment PW



Setup, Maintenance, and Teardown of a Dynamic PW

A dynamic PW uses LDP and encapsulates VC information in the type-length-value (TLV) of LDP packets. PEs need to establish an LDP session, and PW labels are allocated in Downstream Unsolicited (DU) mode and retained in liberal label retention mode.

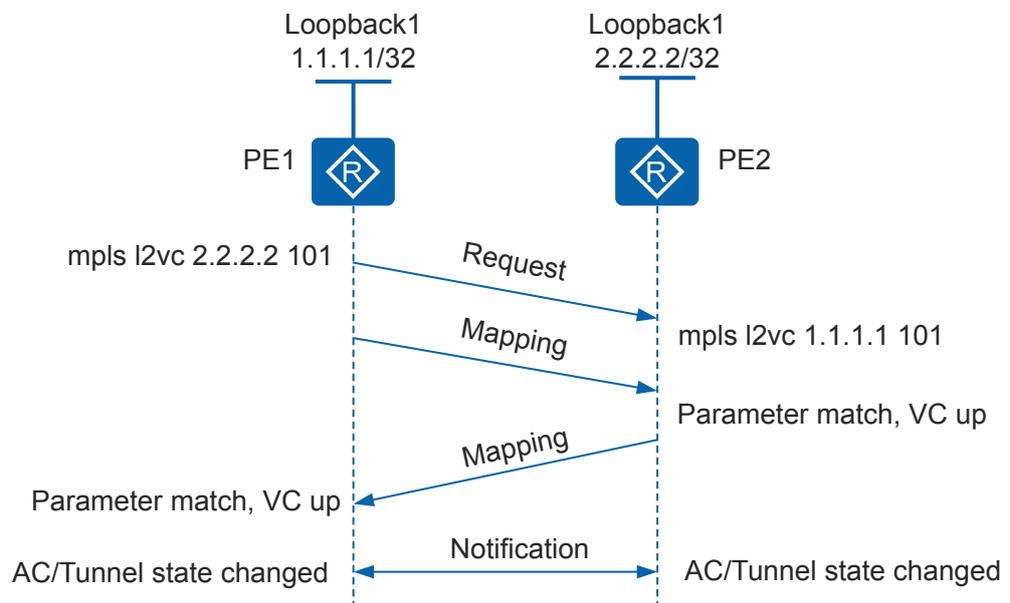
NOTE

If a P exists between PEs, a remote LDP session is established between the PEs. If PEs are directly connected, a common LDP session is established.

After PWE3 configuration is complete on two PEs of the PW and an LDP session is established between PE1 and PE2, PE1 and PE2 start to establish a dynamic PW, as shown in [Figure 12-4](#).

1. PE1 sends a Request message and a Mapping message that contain the local private network label and relevant attributes to PE2.
2. After receiving the Request message from PE1, PE2 sends a Mapping message to PE1.
3. After receiving the Mapping message, PE2 checks whether the same PW parameters as those in the Mapping message are configured locally. If the configured PW parameters such as the VC ID, VC type, MTU, and control word status are the same, PE2 sets the local PW in Up state.
4. After receiving the Mapping message from PE2, PE1 checks whether the locally configured PW parameters are the same as those in the Mapping message. If they are the same, PE1 sets the local PW in Up state. A dynamic PW is set up between PE1 and PE2.
5. After the PW is set up, PE1 and PE2 send Notification messages to report their status.

Figure 12-4 Setup and maintenance of a single-segment PW



When the AC or the tunnel is Down, Martini and PWE3 take different measures:

- Martini sends a Withdraw message to the peer, requesting to tear down the PW. After the AC or tunnel becomes Up, two PEs need to perform negotiation again to establish a PW.

- PWE3 sends a Notification message to notify the peer that packets cannot be forwarded. The PW is not torn down. When the AC or tunnel becomes Up, PWE3 sends a Notification message to notify the peer that packets can be forwarded.

Both PEs tear down the PW only when the PW configuration is deleted or the signaling protocol is interrupted, for example, the public network or PW is Down. On an unstable network, Notification messages can be sent to prevent repeated PW setup and deletion due to link flapping.

Figure 12-5 Process of tearing down a single-segment PW

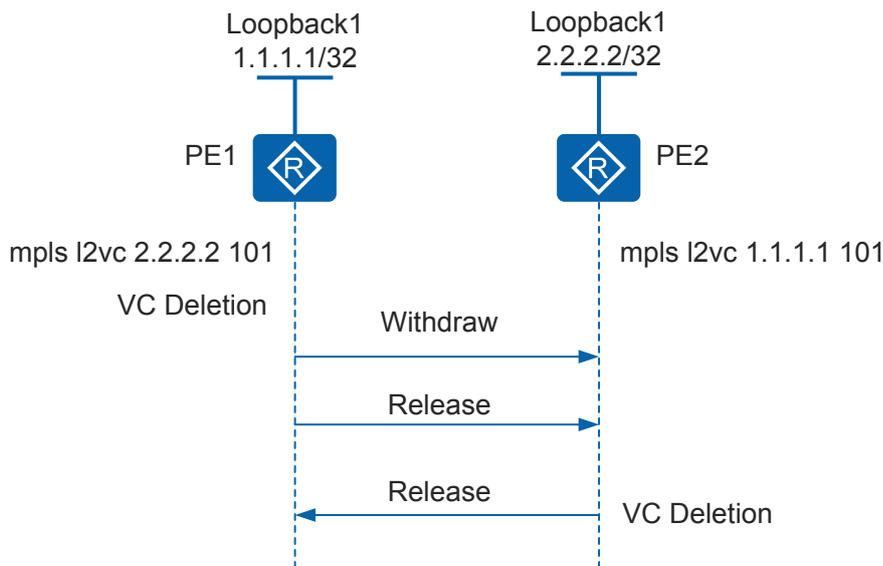


Figure 12-5 shows the PW teardown process.

1. After the PW configuration on PE1 is deleted, PE1 deletes the local VC label and sends Withdraw and Release messages to PE2.

NOTE

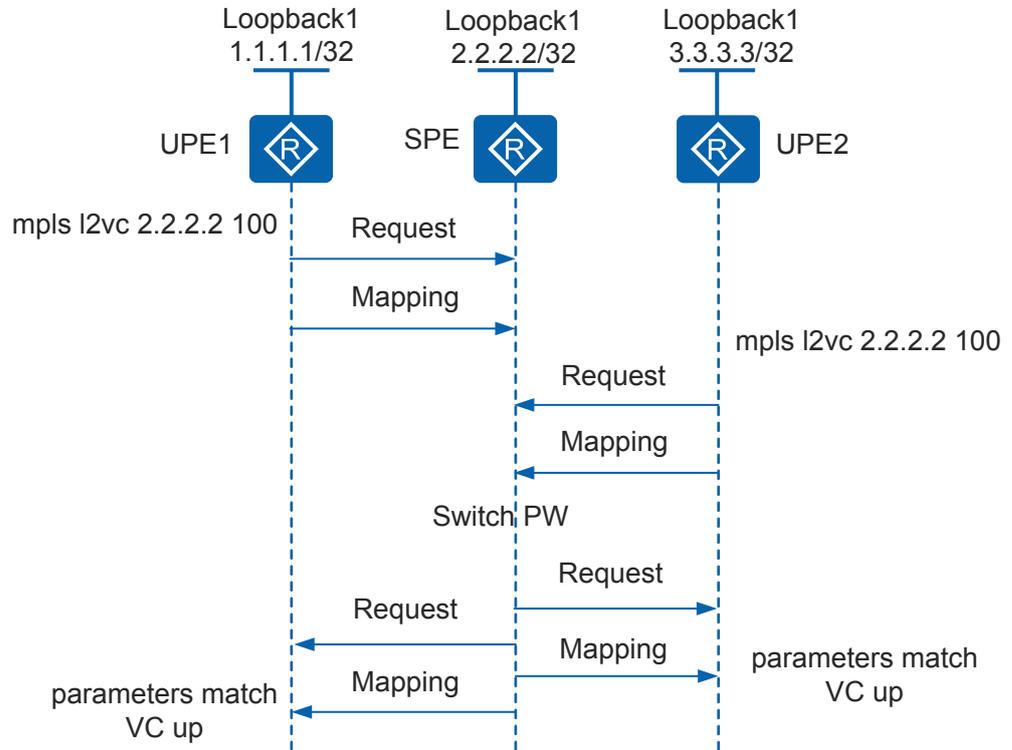
A Withdraw message is used to inform the peer to withdraw labels. A Release message is used to respond to a Withdraw message and request the peer to send a Withdraw message to withdraw labels. To tear down a PW more quickly, PE1 sends a Withdraw message and a Release message consecutively.

2. After receiving Withdraw and Release messages from PE1, PE2 delete the remote VC label and sends a Release message to PE1.
3. After PE1 receives a Release message from PE2, the PW between PE1 and PE2 is torn down.

As shown in **Figure 12-6**, one or more SPEs are deployed between two PEs for a multi-segment PW. PE1 and PE2 establish connections with the SPE and the SPE combines two segments of the PW.

During signaling negotiation, the SPE forwards parameters in the Mapping message from UPE1 to UPE2. Similarly, the SPE forwards parameters in the Mapping message from UPE2 to UPE1. If parameters on UPE1 and UPE2 are the same, the PW status becomes Up. Similar to the Mapping message, Release, Withdraw, and Notification messages are transmitted segment by segment.

Figure 12-6 Signaling exchange on a multi-segment PW

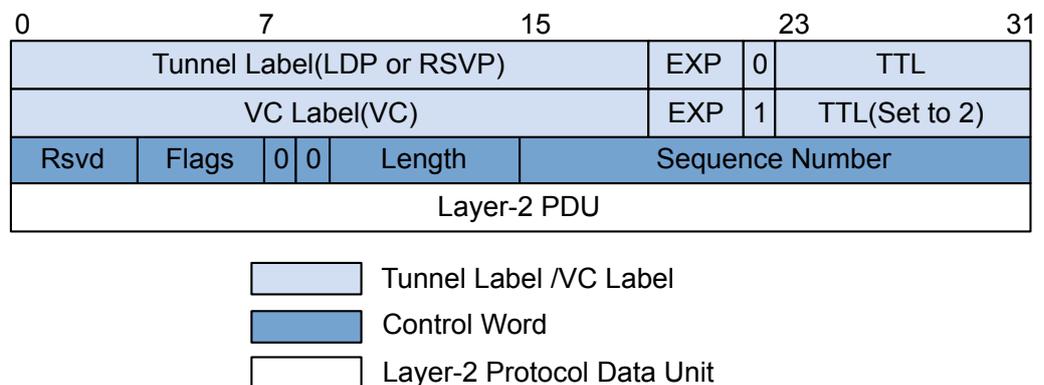


12.3.2 Control Word

The Control Word is negotiated at the control plane, and is used for packet sequence detection, packet fragmentation, and packet reassembly at the forwarding plane. In the PWE3 protocols, ATM Adaptation Layer Type 5 (AAL5) requires support for the CW. The negotiation of the CW at the control plane is simple. If the CW is supported after the negotiation, the negotiation result needs to be delivered to the forwarding module, which detects the packet sequence and reassembles the packet.

The CW is a 4-byte encapsulated packet header, as shown in [Figure 12-7](#). It is used to transmit packets on an MPLS packet switched network (PSN).

Figure 12-7 Position of the CW in a packet



The CW has the following functions:

- Carries the sequence number for forwarding packets.
If the forwarding plane supports the CW, a 32-bit CW is added before the data packet to indicate the packet sequence. When load balancing is supported, the packets may be out of sequence. The CW can be used to number the packets so that the peer can reassemble the packets.
- Pads the packets to prevent the packets from being too short.
For example, if two PEs connect to each other over an Ethernet and CEs connect to PEs over PPP links, PPP negotiation cannot succeed because the size of PPP control packets cannot meet the minimum MTU requirements of an Ethernet. To prevent this problem, you can pad the CW as padding bits to the PPP control packet.
- Carries the control information of the Layer 2 frame header.
In certain cases, the frame does not need to be transmitted completely in the L2VPN packets on the network. The frame header is stripped at the ingress and added at the egress. This method, however, cannot be used if the information in the frame header needs to be carried. You can use the CW to solve this problem. The CW can carry the negotiated information between the ingress PE and the egress PE.

At the control plane, the negotiation succeeds only when both ends or neither end supports the CW. At the forwarding plane, the negotiation result at the control plane determines whether the CW is added to a packet.

12.3.3 VCCV

As MPLS is widely deployed and the MPLS network transmits various types of traffic, the ISP must provide the capability to monitor the label switched path (LSP) status and locate MPLS forwarding faults. Virtual Circuit Connectivity Verification (VCCV) provides the service capability.

VCCV is an end-to-end PW fault detection and diagnosis mechanism, and tests and checks connectivity of the PW forwarding path. VCCV provides the control channel through which connectivity verification (CV) messages are sent between the PW ingress and egress.

Two VCCV modes are available: VCCV ping and VCCV tracer.

- VCCV ping, as an extension of LSP ping, is used to manually test connectivity of a virtual circuit (VC). VCCV ping sends MPLS Echo Request messages through a PW to determine connectivity of the PW. VCCV defines a series of messages transmitted between PEs to verify connectivity of PWs. VCCV ping can be performed in control word channel mode or label alert channel mode:
 - Control word channel: End-to-end detection between UPEs is supported.
 - Label alert channel: Segment by segment detection between the UPE and SPE and end-to-end detection are supported.
- VCCV tracer, as an extension of LSP tracer, is used to locate the faulty node on a PW. VCCV tracer sends MPLS Echo Request messages through a PW to collect information about nodes on the PW. VCCV tracer is classified into PWE3 single-segment tracer and PWE3 multi-segment tracer.

To ensure that both VCCV packets and PW packets are transmitted along the same path, VCCV packets must be encapsulated in the same way and transmitted in the same channel as PW packets.

12.3.4 PWE3 FRR

Definition

As L2VPN is widely used, networks especially L2VPNs that carry real-time services such as VoIP and IPTV require high reliability.

Pseudo-Wire Emulation Edge to Edge Fast Reroute (PWE3 FRR) is a feasible solution to improve L2VPN reliability. It uses Operations, Administration and Maintenance (OAM) and Bidirectional Forwarding Detection (BFD) to detect and report faults on the L2VPN and fast switch traffic, then improving L2VPN reliability.

PWE3 FRR Implementation

PWE3 FRR is implemented as follows:

- The BFD mechanism is used to fast detect a fault on a PW. As a unified detection mechanism used on the entire network, BFD can detect a fault in milliseconds. As the BFD cost is low, if a great number of PWs exist, BFD for PWs greatly reduces the system cost.
- The OAM mapping between a PW and an AC can be created. When a PW or a PE is faulty, a CE can take measures in a timely manner. For example, the CE can rapidly switch traffic to the secondary path.
- OAM messages are transparently transmitted on a PW so that this mechanism implements end-to-end fault detection on PWs and provides PW protection.

In this way, OAM integrates with BFD to implement PWE3 FRR.

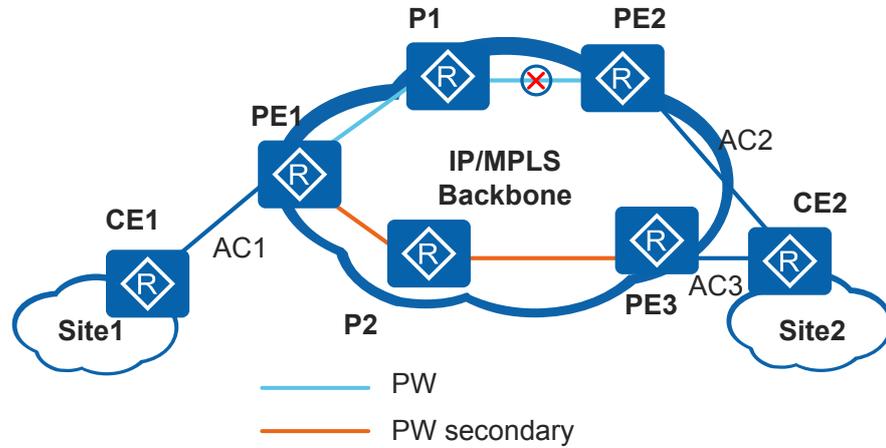
Switchover Mechanism

A fault on the PWE3 FRR network triggers traffic switchover. The device where traffic is switched also reports the fault and terminates fault notification.

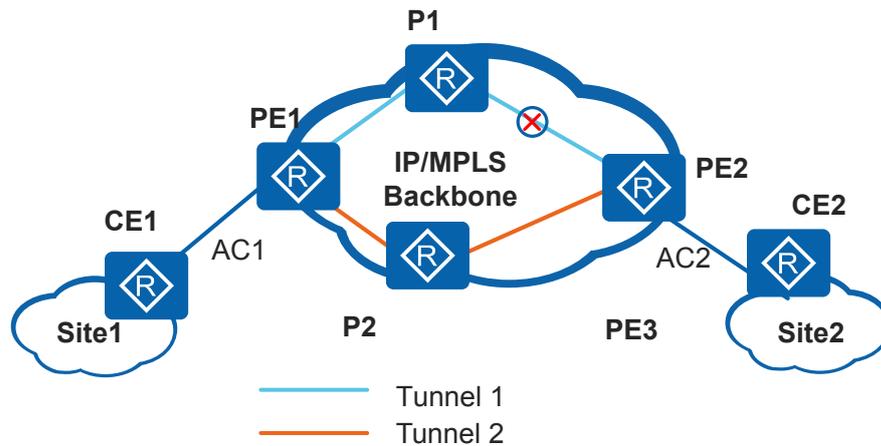
The node that performs fault switchover and terminates fault notification varies with the networking:

- In a networking with asymmetrically connected CEs as shown in [Figure 12-8](#), PE1 and CE2 terminate fault propagation. When PE1 detects a fault, it switches traffic and does not send a fault notification to CE1. CE2 receives the fault notification from PE2 and switches traffic to the secondary link.

Figure 12-8 Networking with asymmetrically connected CEs



- As shown in **Figure 12-9**, two tunnels back up each other on a backbone network. PE1 and PE2 terminate fault notification, and only need to switch the tunnel.
- **Figure 12-9** Tunnel backup on a backbone network



When a CE detects a fault on the primary link, the CE checks whether the secondary link is available. If so, the CE switches traffic to the secondary link. If not, the CE sends an alarm about the service fault.

In a networking with asymmetrically connected CEs as shown in **Figure 12-8**, when PE1 detects a fault on the primary PW, it processes the fault as follows:

- If PE1 also detects a local AC fault, it reports a service fault. In this case, the fault cannot be rectified.
- If PE1 does not detect a local AC fault or a secondary PW fault, it switches traffic to the secondary PW.
- If PE1 does not detect a local AC fault but detects a secondary PW fault, it reports a service fault, but does not switch traffic.

PW Revertive Switchover Policy

In a networking with asymmetrically connected CEs as shown in [Figure 12-8](#), when PE1 is notified of fault recovery on the primary PW, PE1 works based on the configured revertive switchover policy.

The PW revertive switchover policies are as follows:

- None revertive switchover: Traffic is not switched to the primary PW.
- Immediate revertive switchover: Traffic is immediately switched to the primary PW.
- Delayed revertive switchover: Traffic is switched to the primary PW after a delay.

After the switchover, PE1 immediately notifies PE3 on the secondary PW of the fault so that CE2 fast switches the AC through association. In addition, the PE notifies the peer PE on the secondary PW of fault recovery immediately or after a delay, preventing packet loss due to transmission delay between PEs.

12.3.5 Inter-AS Technology

The MPLS VPN solution is widely used, serving an increasing number of users in a large number of applications. As more sites are developed in an enterprise, sites in different geographical locations often connect to different ISP networks. Consider, for example, the inter-AS issue facing operators who manage different metropolitan area networks (MANs) or backbone networks that span different autonomous systems (AS).

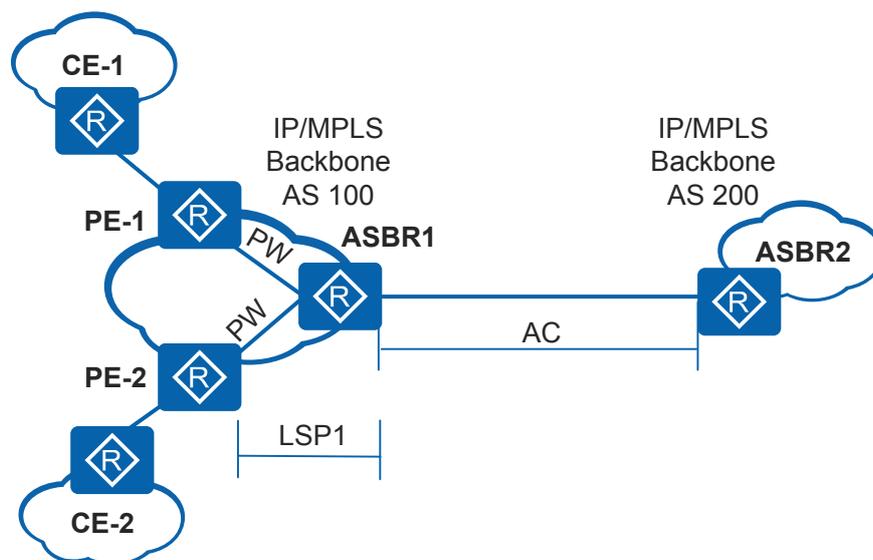
Generally, an MPLS VPN architecture runs within an AS in which VPN routing information is flooded on demand. The VPN routing information within the AS cannot be flooded to the AS of other service providers (SPs). To implement exchange of VPN routing information between different ASs, the inter-AS MPLS VPN model is introduced. The inter-AS MPLS VPN model is an extension of the existing protocol and MPLS VPN framework. This model allows route prefixes and labels to be advertised over the links between different carrier networks.

PWE3 supports inter-AS Option A.

Inter-AS Option A

With Option A, ASBRs of two ASs are directly connected and function as PEs in their respective ASs. The two ASBRs regard each other as a CE.

Figure 12-10 Inter-AS Option A networking



In Figure 12-10, ASBR1 in AS 100 regards ASBR2 in AS 200 as a CE. Similarly, ASBR2 regards ASBR1 a CE.

Option A has the following advantages:

Inter-AS Option A is easy to implement. The two PEs used as ASBRs perform IP forwarding but not MPLS forwarding. In addition, the two PEs do not require special configurations.

Option A has low scalability.

- The PE needs to manage all L2VPN information, which occupies many resources.
- As the ASBR works as the PE in an AS, each PW requires an AC interface, which can be a sub-interface, physical interface, or bundled logical interface.
- If a VPN spans multiple ASs, intermediate ASs must support VPN services. As a result, the configuration workload is heavy and the intermediate ASs are affected.

Therefore, inter-AS Option A is applicable only when there are a small number of inter-AS L2VPNs.

12.4 Application Scenarios for PWE3

This section describes application scenarios for PWE3.

12.4.1 PWE3 Carrying Enterprise Leased Line Services on a MAN

Usage Scenario

Figure 12-11 shows a typical single-segment PWE3 networking. A carrier establishes a MAN to provide PWE3 services. A customer has two branches far from each other. If the customer use leased lines to connect the branches, it will result in high costs. The customer can request the carrier to establish a PWE3 connection between PE1 in branch A and PE2 in branch B.

PWE3 ensures stable Layer 2 communication between branches A and B, facilitates networking, and allows branches A and B to communicate with each other like on a LAN.

Figure 12-11 PWE3 carrying enterprise leased line services on a MAN

PWE3 Deployment

1. IP addresses and IGPs are configured on the carrier MPLS backbone network so that PEs can communicate.
2. MPLS is enabled on the carrier MPLS backbone network, and a TE tunnel is configured between PE1 and PE2. Usually, two TE tunnels are configured to provide tunnel protection.
3. MPLS L2VPN is enabled on PE1 and PE2 and a remote MPLS LDP session is set up between them.
4. PWE3 is configured on AC interfaces of PE1 and PE2 so that PE1 and PE2 can communicate over an MPLS L2VC.

12.5 Summary of PWE3 Configuration Tasks

When configuring PWE3, you can configure static PW, dynamic PW, TDM PWE3, or PW switching. If a network spans multiple autonomous systems (ASs), you need to configure inter-AS PWE3. When a reliability solution is required, you also need to perform other configurations, such as Bidirectional Forwarding Detection (BFD) for PW and PWE3 fast reroute (FRR).

Table 12-1 PWE3 configuration tasks

Scenario	Description	Task
Configure PWE3	<p>When configuring PWE3, you can configure static PW, dynamic PW, TDM PWE3, or PW switching. The application scenario of each is as follows:</p> <ul style="list-style-type: none"> ● Configuring static PW: does not require signaling negotiation or exchange of control packets; therefore, it consumes few resources and is easy to configure. However, it requires manual configuration, which makes network maintenance and expansion difficult. Static PW applies to small-scale MPLS networks with simple topologies. ● Configuring dynamic PW: applies to large-scale enterprises or local area networks (LANs) of small-scale carriers. ● Configuring PW switching: PW labels need to be exchanged during multi-segment PW forwarding. PW switching applies to the following scenarios: <ul style="list-style-type: none"> - Two PEs are not in the same AS, and no signaling connection or tunnel can be set up between the two PEs. - The signaling of two PEs differs from each other. - If the access device supports MPLS, but is incapable of setting up a large number of LDP sessions, you can use User Facing Provider Edge (UFPE) as the UPE. In addition, you can use the SPE as the switching node of LDP sessions, which is similar to a signaling reflector. 	<ul style="list-style-type: none"> ● 12.8.1 Configuring a Static PW ● 12.8.2 Configuring a Dynamic PW ● 12.8.3 Configuring PW Switching ● 12.8.4 Configuring TDM PWE3

Scenario	Description	Task
	<ul style="list-style-type: none"> Configuring TDM PWE3: applies when TDM services are transmitted using PWE3. TDM PWE3 encapsulates TDM service data and transparently transmits the encapsulated data through PWs to implement TDM service transmission. 	
Configure inter-AS PWE3	Configure inter-AS PWE3 if the PWE3 backbone network spans multiple ASs.	12.8.7 Configuring Inter-AS PWE3
Configure PWE3 reliability	<p>When you transmit key services over the PWE3 network, the following reliability solutions can be used:</p> <ul style="list-style-type: none"> Configuring BFD for PW: BFD for PW can rapidly detect a fault on the PW and notify the forwarding plane of the fault, ensuring fast traffic switchover. Configuring PWE3 FRR: PWE3 FRR ensures link-layer reliability for the PWE3 network. 	<ul style="list-style-type: none"> 12.8.5 Configuring Static BFD for PWs 12.8.6 Configuring PWE3 FRR
Configure and apply a tunnel policy	This task is required when PWE3 services need to be transmitted over TE tunnels or when multiple tunnels need to perform load balancing to fully use network resources.	12.8.8 Configuring and Applying a Tunnel Policy

12.6 Licensing Requirements and Limitations for PWE3

Involved Network Elements

None

License Requirements

For L2VPN-capable devices, their licensing requirements for the L2VPN function are as follows:

- AR1200-S series: L2VPN is a basic feature of the device and is not under license control.

- AR2200-S&AR3200-S series: By default, L2VPN function is disabled on a new device. To use the L2VPN function, apply for and purchase the following license from the Huawei local office.
 - AR2200-S series: AR2200 value-added service package for data services
 - AR3200-S series: AR3200 value-added service package for data services

Feature Limitations

- PWE3 cannot be configured on the VLANIF 1.
- The VLANIF interface configured with PWE3 can only correspond to one member interface. This limitation does not apply to AR1220E-S.

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S series do not support PWE3.

12.7 Default Settings for PWE3

[Table 12-2](#) lists the default settings for PWE3.

Table 12-2 Default settings for PWE3

Parameter	Default Setting
MPLS L2VPN	Disabled
MTU	1500
VCCV	Enabled
Control word	Disabled
Jitter buffer depth	8 ms
Number of TDM frames encapsulated in CESoPSN or SAToP packets	8

12.8 Configuring PWE3

This section describes how to configure PWE3 functions in details.

12.8.1 Configuring a Static PW

A static PW does not use signaling protocols to transmit L2VPN packets. Packets are transmitted over the tunnel between PEs.

Pre-configuration Tasks

Before configuring a static PW, complete the following tasks:

- Configuring an IGP protocol on PEs and Ps on the MPLS backbone network to ensure IP connectivity

- Enabling MPLS on PEs and Ps
- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs

You also need to configure tunnel policies when PWE3 services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).

Configuration Procedure

To configure a static PW, perform the following operations on the device. Creating a PW template and setting attributes for the PW template is optional.

12.8.1.1 Enabling MPLS L2VPN

Context

Before configuring a static PW, you must enable MPLS L2VPN.

Perform the following operations on the PEs at both ends of a PW.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mpls l2vpn**

MPLS L2VPN is enabled.

By default, MPLS L2VPN is disabled.

----End

12.8.1.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template

Context

A PW template defines common attributes for PWs, so it can be shared by different PWs. You can run the **pw-template** command to define some common attributes in a PW template to simplify PW configuration. After creating a PW on an interface, you can apply a PW template to the interface.

On the device, you can bind a PW template to a PW or reset the PW.

You can set the attributes for a PW through commands or a PW template. The attributes include the peer, tunnel policy, and control word. Importing the PW template can simplify the configuration of PWs with similar attributes.

 **NOTE**

- Some PW attributes such as the MTU, PW type, and encapsulation type are obtained from the interface connecting the PE to the CE.
- If you specify a PW attribute through a command, the same PW attribute specified in the PW template does not take effect on the PW to which this PW template is applied.

Perform the following operations on the PEs.

Procedure

1. Run **system-view**
The system view is displayed.
2. Run **pw-template** *pw-template-name*
A PW template is created and the PW template view is displayed.
3. Run **peer-address** *ip-address*
A remote IP address is assigned to a PW template.
4. Run **control-word**
The control word function is enabled.
By default, the control word function is disabled.
For dynamic single-segment PWs, dynamic multi-segment PWs, and static single-segment PWs, VCCV in control word mode must be enabled on the UPEs. For static and mixed multi-segment PWs, VCCV in control word mode must be enabled on both the UPEs and SPEs.
5. Run **mtu** *mtu-value*
The MTU in the PW template is specified.
By default, the MTU in a PW template is 1500.
6. Run **tnl-policy** *policy-name*
A tunnel policy is configured for the PW template.
Configure a tunnel policy before you can apply this policy. If no tunnel policy is configured, an LSP tunnel is used and load balancing is not implemented. For details on how to configure a tunnel policy, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).



NOTICE

After modifying the attributes of a PW template, run the **reset pw pw-template** command in the user view to make the modification take effect. This may cause PW disconnection and reconnection. If multiple PWs use this template simultaneously, system operation is affected.

12.8.1.3 Creating a Static PW

Context

When creating a static PW, specify the VC label.

Perform the following operations on the PEs at both ends of a PW.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The AC interface view is displayed.

Step 3 Create a static PW.

1. To create a primary PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value [ tunnel-policy tnl-policy-name | [ control-word | no-control-word ] | [ raw | tagged ] } *
```

2. (Optional) To create a secondary PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value [ tunnel-policy tnl-policy-name | [ control-word | no-control-word ] | [ raw | tagged ] } * secondary
```

NOTE

- When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.
- You can create a secondary PW only after a primary PW is created.
- The combination of the VC ID and VC type must be unique on each node. The VC IDs at both ends of a switching PW can be the same. Primary and secondary PWs must have different VC IDs.
- Primary and secondary PWs must use the same control word; otherwise, many packets may be lost during service switching.

Step 4 (Optional) Run **mpls l2vpn service-name** *service-name*

A name is configured for the L2VPN service.

After a name is configured for the L2VPN service, you can maintain the L2VPN service by clicking the name directly on the NMS GUI.

----End

12.8.1.4 Verifying the Static PW Configuration

Prerequisites

The configurations of the static PW are complete.

Procedure

- (Optional) Run the **display pw-template** [*pw-template-name*] command to check information about the PW template.
- Run the **display mpls static-l2vc** [*vc-id*] **interface** *interface-type interface-number* **state** { **down** | **up** } command to check the information about static VCs.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.

- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check information about the specified tunnel policy.

----End

12.8.2 Configuring a Dynamic PW

This section describes how to configure a dynamic PW. A dynamic PW uses the extended Label Distribution Protocol (LDP) to transmit Layer 2 information and VC labels.

Pre-configuration Tasks

Before configuring a dynamic PW, complete the following tasks:

- Configuring an Interior Gateway Protocol (IGP) protocol on PEs and Ps on the Multiprotocol Label Switching (MPLS) backbone network to ensure IP connectivity
- Configuring basic MPLS functions on the backbone network
- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs

You also need to configure tunnel policies when Pseudo-Wire Emulation Edge to Edge (PWE3) services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).

- Setting up a remote LDP session between the PEs

Configuration Procedure

To configure a dynamic PW, perform the following operations on the device. Creating a PW template and setting attributes for the PW template is optional.

12.8.2.1 Enabling MPLS L2VPN

Context

Before configuring a PW, you must enable MPLS L2VPN.

Perform the following operations on the PEs or UPEs at both ends of a PW.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mpls l2vpn**

MPLS L2VPN is enabled.

NOTE

If the non-Huawei device does not have the capability of processing L2VPN label requests, you need to run the **mpls l2vpn no-request-message** command on the Huawei device to enable the two devices to communicate.

----End

12.8.2.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template

Context

A PW template defines common attributes for PWs, so it can be shared by different PWs. You can run the **pw-template** command to define some common attributes in a PW template to simplify PW configuration. After creating a PW on an interface, you can apply a PW template to the interface.

On the device, you can bind a PW template to a PW or reset the PW.

You can set the attributes for a PW through commands or a PW template. The attributes include the peer, tunnel policy, and control word. Importing the PW template can simplify the configuration of PWs with similar attributes.

NOTE

- Some PW attributes such as the MTU, PW type, and encapsulation type are obtained from the interface connecting the PE to the CE.
- If you specify a PW attribute through a command, the same PW attribute specified in the PW template does not take effect on the PW to which this PW template is applied.

Perform the following operations on the PEs.

Procedure

1. Run **system-view**
The system view is displayed.
2. Run **pw-template** *pw-template-name*
A PW template is created and the PW template view is displayed.
3. Run **peer-address** *ip-address*
A remote IP address is assigned to a PW template.
4. Run **control-word**
The control word function is enabled.
By default, the control word function is disabled.
For dynamic single-segment PWs, dynamic multi-segment PWs, and static single-segment PWs, VCCV in control word mode must be enabled on the UPEs. For static and mixed multi-segment PWs, VCCV in control word mode must be enabled on both the UPEs and SPEs.
5. Run **mtu** *mtu-value*
The MTU in the PW template is specified.
By default, the MTU in a PW template is 1500.
6. Run **tnl-policy** *policy-name*
A tunnel policy is configured for the PW template.
Configure a tunnel policy before you can apply this policy. If no tunnel policy is configured, an LSP tunnel is used and load balancing is not implemented. For details on how to configure a tunnel policy, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).



NOTICE

After modifying the attributes of a PW template, run the **reset pw pw-template** command in the user view to make the modification take effect. This may cause PW disconnection and reconnection. If multiple PWs use this template simultaneously, system operation is affected.

12.8.2.3 Creating a Dynamic PW

Context

To create a dynamic PW, VCs using the same encapsulation type must have different IDs.

Perform the following operations on the PEs.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface interface-type interface-number**

The AC interface view is displayed.

Step 3 Create a PW.

1. To create a primary PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id [ tunnel-policy policy-name | [ control-word | no-control-word ] | [ raw | tagged ] | mtu mtu-value ] *
```

2. (Optional) To create a secondary PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id [ tunnel-policy policy-name | [ control-word | no-control-word ] | [ raw | tagged ] | mtu mtu-value ] * secondary
```

NOTE

- When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.
- A dynamic PW requires that the IDs of VCs using the same encapsulation type are different. Changing the encapsulation type may cause a VC ID collision.
- You can create a secondary PW only after a primary PW is created.
- The combination of the VC ID and VC type must be unique on each node. The VC IDs at both ends of a switching PW can be the same. Primary and secondary PWs must have different VC IDs.
- Primary and secondary PWs must use the same control word configuration; otherwise, many packets may be lost during service switching.

Step 4 (Optional) Run **mpls l2vpn service-name service-name**

A name is configured for the L2VPN service.

After a name is configured for the L2VPN service, you can maintain the L2VPN service by clicking the name directly on the NMS GUI.

----End

12.8.2.4 Verifying the Dynamic PW Configuration

Prerequisites

The configurations of the dynamic PW are complete.

Procedure

- Run the **display pw-template** [*pw-template-name*] command to check information about the PW template.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number* | **remote-info** [*vc-id* | **verbose**] | **state** { **down** | **up** }] command to check the information about virtual circuits in LDP mode.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.
- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check information about the specified tunnel policy.

---End

12.8.3 Configuring PW Switching

You can configure a multi-segment PW to exchange PW labels.

Context

To forward packets through a multi-segment PW, configure PW switching so that PW labels can be exchanged. PW switching must be configured on the Superstratum PE (SPE) where a large number of MPLS LDP sessions can be established.

In the following cases, PW switching is required:

- Two PEs are in different ASs, and no signaling connection or tunnel can be set up between the two PEs.
- Two PEs use different signaling.
- If the access device can run MPLS but does not support a large number of LDP sessions, the User Facing Provider Edge (UFPE) can be used as a UPE and the SPE as a node for switching LDP sessions. The SPE is similar to a signaling reflector.

PW switching supports three modes: static mode, dynamic mode, and mixed mode:

- When static PWs are used between the SPE and two connected PEs, configure static PW switching.
- When dynamic PWs are used between the SPE and two connected PEs, configure dynamic PW switching.
- When a static PW and a dynamic PW are used between the SPE and two connected PEs, configure mixed PW switching.

Pre-configuration Tasks

Before configuring multi-segment PW switching, complete the following tasks:

- Enabling MPLS L2VPN on PEs

- Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs
You also need to configure tunnel policies when PWE3 services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).
- [Configuring a Static PW](#) on UPEs if PW switching is performed between two static PWs
- [Configuring a Dynamic PW](#) on UPEs if PW switching is performed between two dynamic PWs

Procedure

- Configuring static PW switching
Perform the following operations on the SPE.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **mpls switch-l2vc ip-address vc-id trans trans-label rcv received-label [tunnel-policy policy-name] between ip-address vc-id trans trans-label rcv received-label [tunnel-policy policy-name] encapsulation encapsulation-type [control-word [cc { alert | cw } * cv lsp-ping] | [no-control-word] [cc alert cv lsp-ping]] [control-word-transparent]**
Static PW switching is configured.
To configure static PW switching, you must configure PW labels on each SPE.
A static multi-segment PW is established as follows:
 - On the UPE, when the AC status is Up and the PSN tunnel exists, the PW status is Up.
 - On the SPE, as long as the PSN tunnels exist on both sides, the PW is in Up state even if the SPE and UPE use different PW encapsulation types.It is recommended that you set the same PW encapsulation type by specifying the *encapsulation-type* parameter on the SPE and UPE to facilitate management.
- Configuring dynamic PW switching
Perform the following operations on the SPE.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **mpls switch-l2vc ip-address vc-id [tunnel-policy policy-name] between ip-address vc-id [tunnel-policy policy-name] encapsulation encapsulation-type [control-word-transparent]**
Dynamic PW switching is configured.
Dynamic PW switching is easy to configure. Two neighboring endpoints (UPE or SPE) will send remote labels to the SPE through signaling. The two UPEs will send control word and VCCV to the SPE through signaling.
The PW encapsulation type (specified by the *encapsulation-type* parameter) on the SPE must be the same as that on the UPE. Otherwise, the PW cannot go Up.

- Configuring mixed PW switching

 **NOTE**

When you configure mixed PW switching, *ip-address vc-id* before **between** specifies the VC ID of a dynamic PW and *ip-address vc-id* after **between** specifies the VC ID of a static PW. The two values cannot be interchanged.

Perform the following operations on the SPE.

- a. Run **system-view**

The system view is displayed.

- b. Run **mpls switch-l2vc** *ip-address vc-id* [**tunnel-policy** *policy-name*] **between** *ip-address vc-id* **trans** *trans-label* **recv** *received-label* [**tunnel-policy** *policy-name*] **encapsulation** *encapsulation-type* [**mtu** *mtu-value*] [**control-word** [**cc** { **alert** | **cw** } * **cv** **lsp-ping**]] [**no-control-word**] [**cc** **alert** **cv** **lsp-ping**]] [**control-word-transparent**]

Mixed PW switching is configured.

To configure mixed PW switching, configure a PW label for the static PW. The PW encapsulation type (specified by the *encapsulation-type* parameter) on the SPE must be the same as that on the UPE connected to a dynamic PW.

To configure mixed PW switching, the following MTUs must be the same:

- Local MTU of the dynamic PW
- Peer MTU of the dynamic PW
- Local MTU of the static PW
- Peer MTU of the static PW

 **NOTE**

In mixed PW switching, the MTUs of the interfaces on the two ends must be the same and cannot be greater than 1500 bytes.

---End

Verifying the Configuration

- Run the **display mpls static-l2vc** [*vc-id* | **interface** *interface-type interface-number* | **state** { **down** | **up** }] command to check the information about static VCs.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number* | **remote-info** [*vc-id* | **verbose**]] | **state** { **down** | **up** }] command to check the information about virtual circuits in LDP mode.
- Run the **display mpls switch-l2vc** [*ip-address vc-id* **encapsulation** *encapsulation-type* | **state** { **down** | **up** }] command on the SPE to view information about PW switching.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.
- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check information about the specified tunnel policy.

12.8.4 Configuring TDM PWE3

You can configure TDM PWE3 to encapsulate TDM service data and transparently transmit the data through PWs to implement TDM service transmission through PWE3.

Pre-configuration Tasks

Before configuring TDM PWE3, complete the following tasks:

- Configuring an IGP protocol on the Ps and PEs on the MPLS backbone network to implement IP connectivity
 - Configuring basic MPLS capabilities on the backbone network
 - Installing a 8SA, 6E&M, 8E1T1-M, or 8E1T1-F interface card on the device and ensuring that it is registered successfully
 - Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs
- You also need to configure tunnel policies when PWE3 services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).

NOTE

Only the AR2220-S, AR2240-S, and AR3200-S (using SRU40 , SRU60, or SRU80) support this function.

Configuration Procedure

To configure TDM PWE3, you need to perform the following configurations, among which creating a PW template and setting attributes for the PW template is optional.

12.8.4.1 Configuring an AC Interface to Transparently Transmit TDM Cells

Context

You can configure an AC interface to transparently transmit TDM cells, so that after receiving TDM packets, the AC interface encapsulates the packets and transmits the packets through a PW.

Perform the following operations on the PEs.

Procedure

Step 1 Use either of the following methods to configure AC interface parameters based on the interface card type:

- 8SA interface card: Configure interface parameters such as a working mode for serial interfaces. Ensure that the parameters are the same as those of a CE's interface connected to the AC interface.

NOTE

For the detailed configuration of a serial interface, see *Configuring a Synchronous Serial Interface in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Interface Management*.

- 6E&M interface card: Configure interface parameters for E&M interfaces. Ensure that the parameters are the same as those of a CE's interface connected to the AC interface.

NOTE

For the detailed configuration of an E&M interface, see *Configuring Line Attributes for an E&M Interface in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Interface Management*.

- 8E1T1-M interface card: Configure interface parameters such as a working mode for CE1/PRI interfaces. Ensure that the parameters are the same as those of a CE's interface connected to the AC interface. To ensure that CEs exchange data successfully, the AC interfaces must work in clock synchronization state.

 **NOTE**

For the detailed configuration of a CE1/PRI interface, see *Configuring CE1/PRI Interface in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Interface Management*.

- 8E1T1-F interface card: Configure interface parameters such as a working mode for E1-F interfaces. Ensure that the parameters are the same as those of a CE's interface connected to the AC interface. To ensure that CEs exchange data successfully, the AC interfaces must work in clock synchronization state.

 **NOTE**

For the detailed configuration of an E1-F interface, see *Configuring E1-F Interface in Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Interface Management*.

Step 2 Configure the AC interface to transparently transmit TDM cells.

1. Run **system-view**

The system view is displayed.

2. Run **interface serial interface-number**

The serial interface view is displayed.

3. Run **link-protocol tdm**

The link layer protocol used for packet encapsulation on the serial interface is set to TDM.

---End

12.8.4.2 (Optional) Creating a PW Template and Setting Attributes for the PW Template

Context

A PW template defines common attributes for PWs, so it can be shared by different PWs. The PW template simplifies PW configuration. You can use the PW template to create a PW on an interface.

On the device, the PW can be bound to a PW template and can be reset.

You can set the attributes for a PW through commands or a PW template. The attributes include the peer, tunnel policy, and control word. Importing the PW template can simplify the configuration of PWs with similar attributes.

 **NOTE**

- Some PW attributes such as the MTU, PW type, and encapsulation type are obtained from the interface connecting the PE to the CE.
- If you specify a PW attribute through commands, the same PW attribute specified in the PW template does not function on the PW to which this PW template is applied.

Perform the following steps on the PE devices.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **pw-template** *pw-template-name*

A PW template is created and the PW template view is displayed.

Step 3 Run **peer-address** *ip-address*

The remote device IP address of the PW is specified.

Step 4 Run **control-word**

The control word function is enabled.

By default, the control word function is disabled.

To enable VCCV in control word mode, enable the control word function only on the UPE for dynamic single-hop PWs, dynamic multi-hop PWs, and static single-hop PWs; enable the control word function on both the UPE and SPE for static multi-hop PWs and mixed dynamic multi-hop PWs.

Step 5 Run **mtu** *mtu-value*

The MTU in the PW template is specified.

By default, the MTU in a PW template is 1500.

Step 6 Run **jitter-buffer depth** *depth*

The jitter buffer depth is configured.

By default, the jitter buffer depth is 8 ms.

The jitter buffer is used to reduce jitter at the network side. A larger jitter buffer depth indicates a stronger anti-jitter capability. However, when a long transmission delay is introduced when data flows are reconstructed, improper jitter buffer depth degrades transmission quality.

Step 7 Run **tdm-encapsulation-number** *number*

The number of encapsulated TDM frames in Circuit Emulation Services over Packet Switch Network (CESoPSN) or Structure-Agnostic Time Division Multiplexing over Packet (SAToP) packets in the TDMoPSN application is set.

By default, the number of encapsulated TDM frames in a CESoPSN or SAToP packet is 8. When the interface card where the AC interface locates is 8SA, 8E1T1-M or 8E1T1-F, the interface can encapsulate a maximum of 16 TDM frames in a packet. If the value is larger than 16, the interface only encapsulates 16 TDM frames in a packet.

You can determine the number of TDM frames encapsulated into each PW packet. The smaller the number of frames encapsulated into a packet, the shorter the encapsulation and transmission delay but the more the encapsulation overhead. The larger the number of frames encapsulated into a packet, the higher the bandwidth utilization but the longer the encapsulation delay.

Step 8 Run **idle-code** *idle-code-value*

The device is configured to fill idle codes when a jitter buffer underflow occurs.

By default, the system fills idle codes to FF.

A jitter buffer underflow occurs when the device needs to read packets but there are not sufficient packets in the buffer. The idle code has no significance, and can be set to any value.

Step 9 Run `tnl-policy policy-name`

A tunnel policy is configured for the PW.

Configure a tunnel policy before you can apply this policy. If no tunnel policy is configured, an LSP tunnel is used and load balancing is not implemented. For details on the tunnel policy configuration, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).

---End

Follow-up Procedure

After modifying the attributes of a PW template, run the `reset pw pw-template` command in the user view to make the modification take effect. This may cause PW disconnection and reconnection. If multiple PWs use this template simultaneously, system operation is affected.

12.8.4.3 Configuring PW

Context

When configuring TDM PWE3, you can create static PWs, dynamic PWs, or PW switching.

Procedure

Step 1 Run `system-view`

The system view is displayed.

Step 2 Run `interface serial interface-number`

The serial interface view is displayed.

Step 3 Configure PWE3 function.

Choose either of them based on the networking.

1. Configure a static PW.

a. To configure an active PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value [ tunnel-policy tnl-policy-name | [ control-word | no-control-word ] | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation number | tdm-sequence-number ] *
```

b. (Optional) To configure a standby PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value [ tunnel-policy tnl-policy-name | [ control-word | no-control-word ] | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation number | tdm-sequence-number ] * secondary
```

 **NOTE**

- You can configure a standby PW only after an active PW is configured.
- The combination of the PW ID and PW type must be unique on each node. The IDs of PWs on two ends can be the same. Active and standby PWs must have different VC IDs.
- Active and standby PWs must use the same control word configuration; otherwise, many packets may be lost during service switching.

2. Configure a dynamic PW

- a. To configure an active PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id [ tunnel-policy policy-name | [ control-word | no-control-word ] | mtu mtu-value | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation-number number | tdm-sequence-number ] *
```

- b. (Optional) To configure a standby PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id [ tunnel-policy policy-name | [ control-word | no-control-word ] | mtu mtu-value | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation-number number | tdm-sequence-number ] * secondary
```

 **NOTE**

- A dynamic PW requires that the VC ID of the same encapsulation type on a PE be unique. Changing the encapsulation type may cause a VC ID collision.
- You can configure a standby PW only after an active PW is configured.
- The combination of the PW ID and PW type must be unique on each node. The IDs of PWs on two ends can be the same. Active and standby PWs must have different VC IDs.
- Active and standby PWs must use the same control word configuration; otherwise, many packets may be lost during service switching.

3. Configuring PW Switching

Perform this operation on the SPE only. Select static PW or dynamic PW based on the PW switching mode on the UPE.

PW switching is classified into static mode, dynamic mode, and mixed mode:

- When static PWs are used between the SPE and two connected PEs, configure static PW switching.
- When dynamic PWs are used between the SPE and two connected PEs, configure dynamic PW switching.
- When a static PW and a dynamic PW are used between the SPE and two connected PEs, configure mixed PW switching.

Perform either of the following operations based on actual needs.

- To configuring a static PW switching, run:

```
mpls switch-l2vc ip-address vc-id trans trans-label recv received-label [ tunnel-policy policy-name ] between ip-address vc-id trans trans-label recv received-label [ tunnel-policy policy-name ] encapsulation encapsulation-type [ control-word [ cc { alert | cw } * cv lsp-ping ] | [ no-control-word ] [ cc alert cv lsp-ping ] ] [ control-word-transparent ]
```

 NOTE

When different PW encapsulation modes are selected, the parameters to be specified are as follows:

- If *encapsulation-type* is set to **satop-e1**, the following parameters can be specified: **[no-control-word] [cc alert cv lsp-ping]**.
- If *encapsulation-type* is set to **cesopns-basic**, the following parameters can be specified: **control-word [cc {alert | cw} * cv lsp-ping], [no-control-word] [cc alert cv lsp-ping]**.

The *timeslotnum timeslotnum* needs to be configured.

- To configuring a dynamic PW switching, run:

```
mpls switch-l2vc ip-address vc-id [ tunnel-policy policy-name ] between  
ip-address vc-id [ tunnel-policy policy-name ] encapsulation  
encapsulation-type [ control-word-transparent ]
```

- To configuring a mixed PW switching, run:

```
mpls switch-l2vc ip-address vc-id [ tunnel-policy policy-name ] between  
ip-address vc-id trans trans-label rcv received-label [ tunnel-policy  
policy-name ] encapsulation encapsulation-type [ mtu mtu-value ]  
[ control-word | no-control-word ] [ timeslotnum timeslotnum ] [ tdm-  
encapsulation number ] [ control-word-transparent ]
```

 NOTE

- When you configure mixed PW switching, *ip-address vc-id* before **between** specifies the VC ID of a dynamic PW and *ip-address vc-id* after **between** specifies the VC ID of a static PW. The two values cannot be interchanged.
- In mixed PW switching, the MTUs of the interfaces on the two ends must be the same and cannot be greater than 1500 bytes.
- When different PW encapsulation modes are selected, the parameters to be specified are as follows:
 - If *encapsulation-type* is set to **satop-e1**, the following parameters can be specified: **mtu mtu-value, control-word, no-control-word, and tdm-encapsulation number**.
 - If *encapsulation-type* is set to **cesopns-basic**, the following parameters can be specified: **mtu mtu-value, control-word, no-control-word, timeslotnum timeslotnum, and tdm-encapsulation number**.

The *timeslotnum timeslotnum* needs to be configured.

Step 4 (Optional) Run **mpls l2vpn pw performance disable**

The statistics collection function for PWs on the TDM interface is disabled.

After PW is configured on a TDM interface, the statistics collection function for PWs on the TDM interface is enabled by default. The system collects statistics on sent and received packets of the primary PW at an interval of 15 minutes. To improve statistics collection efficiency for system performance or other performance, you can perform this step to disable the statistics collection function for PWs on the TDM interface.

----End

12.8.4.4 Verifying the TDM PWE3 Configuration

Prerequisites

The configurations of the TDM PWE3 are complete.

 **NOTE**

After a PW with the encapsulation type being TDM is configured on the AC side, PW performance statistics collection is enabled on the related TDM interface by default. Run the **undo mpls l2vpn pw performance disable** command can enable the function of PW performance information collection on a TDM interface.

Procedure

- Run the **display pw-template** [*pw-template-name*] command to check information about the PW template.
- Run the **display mpls static-l2vc** [*vc-id* | **interface** *interface-type interface-number* | **state** { **down** | **up** }] command to check the information about static VCs.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number* | **remote-info** [*vc-id* | **verbose**] | **state** { **down** | **up** }] command to check the information about virtual circuits in LDP mode.
- Run the **display mpls switch-l2vc** [*ip-address vc-id encapsulation encapsulation-type* | **state** { **down** | **up** }] command on the SPE to view information about PW switching.
- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.
- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check information about the specified tunnel policy.
- Run the **display mpls l2vpn interface** *interface-type interface-number* **performance** command to check PW performance information on a TDM interface.

----End

12.8.5 Configuring Static BFD for PWs

Static BFD for PWs enables the device to fast detect faults on PWs and trigger switching of upper-layer applications. Static BFD applies to small-scale networks.

Pre-configuration Tasks

Before configuring static BFD for PWs, complete the following tasks:

- Setting IP parameters to make each node reachable
- Configuring a PW

Configuration Procedure

The following configurations are mandatory and must be performed in sequence.

12.8.5.1 Enabling BFD Globally

Context

Before configure BFD for PWs, enable BFD globally. Perform the following operations on the PEs at both ends of a PW.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bfd**

BFD is enabled globally on the local node and the BFD view is displayed.

----End

12.8.5.2 Configuring BFD for PWs

Context

You must configure or delete BFD for PWs on the two PEs of a PW simultaneously; otherwise, the PW status on the two PEs may be different. Perform the following operations on the two PEs of the PW to be detected.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bfd *cfg-name* bind pw interface *interface-type* *interface-number* [**secondary**]**

BFD for PWs is configured.

The outbound interface **interface *interface-type* *interface-number*** bound to a BFD session is the AC interface where a PW resides.

To detect a secondary PW, specify **secondary**.

Step 3 Configure BFD discriminators.

- Run **discriminator local *discr-value***
The local discriminator is set.
- Run **discriminator remote *discr-value***
The remote discriminator is set.

NOTE

The local discriminator at the local end must be the same as the remote discriminator at the peer end, and the remote discriminator at the local end must be the same as the local discriminator at the peer end.

Step 4 Run **commit**

The configuration is committed.

When the PW status is Down, a BFD session can be set up but cannot go Up.

 **NOTE**

- The local and remote BFD discriminators cannot be modified once being configured. To modify the local or remote BFD discriminator, run the **undo bfd bfd-name** command in the system view to delete the configuration of BFD for PWs, and then reconfigure the local or remote BFD discriminator.
- After the PW is deleted, the related BFD session and configuration are deleted.

Step 5 (Optional) Set the encapsulation type for BFD CV packets to be sent to remote peers.

1. Run **quit**

Return to the system view.

2. Run **mpls l2vpn**

The MPLS L2VPN view is displayed.

3. Run either of the following commands:

- To set the encapsulation type to 0x04 for BFD CV packets to be sent to all remote peers, run the **mpls l2vpn vccv bfd-cv-negotiation fault-detection-only** command.
- To set the encapsulation type 0x08 for BFD CV packets to be sent to all remote peers, run the **undo mpls l2vpn vccv bfd-cv-negotiation fault-detection-only** command.
- To set the encapsulation type to 0x04 for BFD CV packets to be sent to a specified remote peer, run the **mpls l2vpn vccv bfd-cv-negotiation fault-detection-only peer peer-address enable** command.
- To set the encapsulation type to 0x08 for BFD CV packets to be sent to a specified remote peer, run the **mpls l2vpn vccv bfd-cv-negotiation fault-detection-only peer peer-address disable** command.
- To restore the global encapsulation type of BFD CV packets to be sent to a specified remote peer, run the **undo mpls l2vpn vccv bfd-cv-negotiation fault-detection-only peer peer-address** command.

----End

12.8.5.3 Verifying the Configuration of Static BFD for PWs

Prerequisites

The configurations of static BFD for PWs are complete.

Procedure

- Run the **display bfd configuration pw interface interface-type interface-number [secondary] [verbose]** command to check the BFD configuration.
- Run the **display bfd session pw interface interface-type interface-number [secondary] [verbose]** command to check information about the BFD session.

----End

12.8.6 Configuring PWE3 FRR

After PWE3 FRR is configured, the L2VPN traffic is rapidly switched to the secondary path when a fault occurs on the primary path. After the fault on the primary path is rectified, the L2VPN traffic is switched back to the primary path based on a revertive switchover policy.

Context

On the network where CEs are asymmetrically connected to PEs, the secondary PW cannot transmit data when the primary path and secondary path work properly. If the AC interface of the secondary PW borrows the IP address of the AC interface of the primary PW, note the following points:

- The switching policy **No revertive switchover** cannot be configured.
- The local CE has two equal-cost and direct routes to the remote CE. The destination addresses and next hops of the two routes are the same. The route that passes through the secondary PW is unreachable.
- If the CEs exchange routing information using routing protocols, change the cost or metric value of the AC interface of the secondary path to a value greater than that of the AC interface of the primary path. The local CE may be unable to communicate with the remote CE, but can communicate with other remote user devices.
- If CEs use static routes and the AC links are Ethernet links, BFD for static routes needs to be configured on CEs.

Pre-configuration Tasks

Before configuring PWE3 FRR, complete the following tasks:

- Configuring primary and secondary PWs of the same type on the network where CEs are asymmetrically connected to PEs
 - Configuring CEs to exchange routing information using routing protocols or static routes
 - Setting up a tunnel (GRE tunnel, LSP tunnel, or TE tunnel) between the PEs
- You also need to configure tunnel policies when PWE3 services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources. For details, see step 1 in [12.8.8 Configuring and Applying a Tunnel Policy](#).

Configuration Procedure

Perform the operations in the following sequence. You can determine whether to perform optional operations based on site requirements.

12.8.6.1 Configuring Primary and Secondary PWs

Context

You can configure primary and secondary PWs to protect services on the PWs.

- On the network where CEs are symmetrically dual-homed to PEs, configure one primary PW for each of the primary and secondary paths. The primary and secondary paths can be configured with different types of PWs.
- On the network where CEs are asymmetrically connected to PEs, configure primary and secondary PWs for the primary and secondary paths respectively. The primary and secondary PWs must be of the same type.

Devices support only dynamic primary and secondary PWs.

Perform the following operations on the two PEs of a PW.

Procedure

Step 1 Configure dynamic primary and secondary PWs on the PEs. For details, see [12.8.2 Configuring a Dynamic PW](#).

 **NOTE**

- Primary and secondary PWs must have different VC IDs.
- Primary and secondary PWs must use the same control word; otherwise, many packets may be lost during service switching.

Step 2 (Optional) Configure other primary and secondary PW functions.

1. Run **interface** *interface-type interface-number*

The AC interface view is displayed.

2. Run **mpls l2vpn stream-dual-receiving**

The primary and secondary PWs are configured to receive packets simultaneously.

When PWE3 FRR is configured on a network, you must configure the primary and secondary PWs to receive packets simultaneously on the PE to which the PWs are single-homed, preventing packet loss during PW revertive switchover.

----End

12.8.6.2 (Optional) Configuring Fast Fault Notification - OAM Mapping

Context

OAM mapping expedites the fault detection and notification on the AC end. OAM mapping can be configured on various types of links. To configure OAM mapping on Ethernet links, the PE and CE devices must support the Ethernet OAM function.

Choose either of the following procedures to configure OAM mapping according to the AC types.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The view of the AC interface is displayed.

Step 3 Run **mpls l2vpn oam-mapping 3ah**

The fault mapping between the AC and the PW is enabled.

 NOTE

- The PW need be configured in homogeneous interworking mode when the AC is an Ethernet. Otherwise, the use device may learn a wrong outbound interface according to ARP.
- Before running the **mpls l2vpn oam-mapping 3ah** command, you need configure Ethernet OAM on the AC link. For details, refer to "EFM Configuration" in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Reliability*.
- If the **mpls l2vpn oam-mapping** command is configured, run the **display mpls l2vc interface** command to check the VC status. In the command output, "Local AC OAM State" indicates the status of the AC link; if the **mpls l2vpn oam-mapping** command is not configured, run the **display mpls l2vc interface** command to check the VC status. In the command output, "Local AC OAM State" is always Up, and has no relationship with the AC link status.

---End

12.8.6.3 (Optional) Configuring BFD for PW

Context

BFD for PW is recommended because it speeds up fault detection.

Procedure

For details, see the following topics.

- [12.8.5 Configuring Static BFD for PWs](#)

 NOTE

- BFD for PW on both PEs at the two ends must be configured or deleted simultaneously. Otherwise, the statuses of PWs on the PEs are inconsistent.
- To monitor statuses of tunnels that carry PWs, configure BFD for tunnel. For detailed configurations, see "MPLS LDP Configuration" and "MPLS TE Configuration" in *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Manual MPLS*.

12.8.6.4 (Optional) Configuring a Revertive Switchover Policy

Context

Revertive switching policies are classified into the following types:

- Immediate revertive switchover: When the primary PW recovers from a fault, the local PE switches traffic back to the primary PW immediately and notifies the peer PE on the secondary PW of the fault. In FRR mode, the local PE notifies the peer PE on the secondary PW of the recovery after a delay of *resume-time*. In PW redundancy master/slave mode, the parameter *resume-time* is not supported.
This revertive switchover applies to scenarios in which users hope traffic to be restored as soon as possible.
- Delayed revertive switchover: When the primary PW recovers from a fault, traffic is switched back to the primary PW after a period specified by *delay-time*. After traffic is switched back, the local device immediately notifies the peer device on the secondary PW of the fault. If *resume-time* is configured in FRR mode, the local device notifies the peer device on the secondary PW of the recovery after a delay of *resume-time*.

On a large-scale network, packet loss caused by incomplete route convergence may occur during the switchover. To prevent this problem, configure traffic to be switched back after a delay.

- None revertive switchover: When the primary PW recovers from a fault, traffic is not switched back to the primary PW until the secondary PW becomes faulty.

If you do not want traffic to be frequently switched between the primary and secondary PWs, you can use the non-revertive switchover.

By default, the delayed revertive switchover is performed.

A revertive switchover policy is configured on a PE. In asymmetric networking, if the active PW is faulty, the PE to which a CE is connected through a single link switches traffic. When the active PW is restored, configure a revertive switchover policy on this PE. The PE then processes traffic based on the configured revertive switchover policy.

Perform the following operations on the PE (where traffic is switched) to which the CE is connected through a single link.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface** *interface-type interface-number*

The AC interface view is displayed.

Step 3 Run **mpls l2vpn reroute** { { **delay** *delay-time* | **immediately** } [**resume** *resume-time*] | **never** }

The revertive switchover policy is configured.

For an asymmetric networking with ACs of the Ethernet type, if the Ethernet OAM function is configured on the PE interface connected to a CE, and a revertive switching policy is also configured, do not set *resume-time* to 0 seconds. Set *resume-time* to 1 second or longer.

NOTE

On the network where CEs are asymmetrically connected to PEs, the secondary PW cannot transmit data when the primary and secondary paths work normally. On the CE in the dual-homed site, if the interface of the secondary PW borrows the IP address of the interface of the primary PW, you cannot configure revertive switchover.

----End

12.8.6.5 Verifying the PWE3 FRR Configuration

Prerequisites

All configurations about PWE3 FRR are complete.

After PWE3 FRR is configured, you can view information about the local and remote PWs, BFD sessions, L2VPN forwarding, and OAM mapping. You can also run the **manual-set pw-ac-fault** command to simulate faults on a PW to verify whether the switchover between the primary and secondary PWs is normal.

Procedure

- Run the **manual-set pw-ac-fault** command on the interface of the primary PW to simulate faults on it to verify whether the switchover between the primary and secondary PWs is normal.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number*] command to check information about the local PWs.
- Run the **display mpls l2vc remote-info** [*vc-id*] command to check information about the remote PWs.
- Run the **display bfd session pw interface** *interface-type interface-number* [**secondary**] [**verbose**] command to check information about the BFD session.
- Run the **display mpls l2vpn forwarding-info** [*vc-label*] **interface** *interface-type interface-number* command to check the MPLS L2VPN forwarding information.

----End

12.8.7 Configuring Inter-AS PWE3

Inter-AS PWE3 allows the MPLS backbone network to transmit PWE3 services over multiple ASs.

Context

The devices support inter-AS PWE3 Option A. Option A is easy to implement; however, each ASBR must provide a dedicated interface for each inter-AS VC. The interface can be a sub-interface, physical interface, or logical interface. If the number of inter-AS VCs is small, this solution can be used.

Pre-configuration Tasks

Before configuring inter-AS PWE3, complete the following tasks:

- Configuring an IGP protocol for the MPLS backbone network in each AS to ensure IP connectivity of the backbone network within an AS
- Configuring basic MPLS functions on the MPLS backbone network in each AS
- Configuring MPLS LDP and establishing the LDP LSP for the MPLS backbone in each AS

Procedure

The configurations of inter-AS PWE3 Option A are as follows:

- Perform the operation of **Configuring a Dynamic PW** in each AS.
- Configure the ASBRs. Each ASBR considers the peer ASBR as its CE.

NOTE

You do not need to perform any additional configuration for inter-AS implementation on ASBRs and do not need to configure IP addresses for the directly connected interfaces between ASBRs.

The configuration details are not mentioned here.

Verifying the Configuration

- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number*] command on the PE to view information about the local PW.
- Run the **display mpls l2vc remote-info** [*vc-id*] command on the PE to view information about the remote PW.

12.8.8 Configuring and Applying a Tunnel Policy

You need to configure tunnel policies on PEs when PWE3 services need to be transmitted over TE tunnels or when PWE3 services need to be load balanced among multiple tunnels to fully use network resources.

Context

Service data on the PWE3 network is transmitted over tunnels. By default, LSP tunnels are used to transmit data, and each service is transmitted by only one LSP tunnel.

If the default tunnel configuration cannot meet PWE3 service requirements, apply tunnel policies to VPNs. You can configure either of the following types of tunnel policies based on service requirements:

- Tunnel type prioritization policy: This policy can change the type of tunnels selected for PWE3 data transmission or select multiple tunnels for load balancing.
- Tunnel binding policy: This policy can bind multiple TE tunnels to provide QoS guarantee for PWE3.

Pre-configuration Tasks

Before configuring and applying a tunnel policy, complete the following task:

- For details on how to create a GRE tunnel, see [GRE Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - VPN](#).
- For details on how to create an LSP tunnel, see [MPLS LDP Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS](#).
- For details on how to create a TE tunnel, see [MPLS TE Configuration in the Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - MPLS](#).

Perform the following operations on the PEs that need to use a tunnel policy.

Procedure

Step 1 Configure a tunnel policy.

Use either of the following methods to configure a tunnel policy.

Configure a tunnel type prioritization policy.

By default, no tunnel policy is configured. LSP tunnels are used to transmit PWE3 data and each VPN service is transmitted over one LSP tunnel.

1. Run **system-view**

The system view is displayed.

2. Run **tunnel-policy** *policy-name*

A tunnel policy is created, and tunnel policy view is displayed.

3. (Optional) Run **description** *description-information*

The description of the tunnel policy is configured.

4. Run **tunnel select-seq** { **cr-lsp** | **gre** | **lsp** } * **load-balance-number** *load-balance-number*

The sequence in which each type of tunnel is selected and the number of tunnels participating in load balancing are set.

Configure a tunnel binding policy.

1. Run **system-view**

The system view is displayed.

2. Run **interface tunnel** *interface-number*

The tunnel interface view of the MPLS TE tunnel is displayed.

3. Run **mpls te reserved-for-binding**

The binding capability of the TE tunnel is enabled.

4. Run **mpls te commit**

The MPLS TE configuration is committed for the configuration to take effect.

5. Run **quit**

Return to the system view.

6. Run **tunnel-policy** *policy-name*

A tunnel policy is created.

7. (Optional) Run **description** *description-information*

The description of the tunnel policy is configured.

8. Run **tunnel binding destination** *dest-ip-address* **te** { **tunnel** *interface-number* } &<1-16> [**ignore-destination-check**] [**down-switch**]

The TE tunnel is bound to a specified tunnel policy.

NOTE

- If the PE has multiple peers, you can run the **tunnel binding** command multiple times to specify different destination IP addresses in a tunnel policy.
- If **down-switch** is specified in the command, the system selects available tunnels in an order of LSP, CR-LSP, and GRE when the bound tunnels are unavailable.

Step 2 Apply the tunnel policy.

Perform the following operations on AC interfaces on the PEs.

1. Run **system-view**

The system view is displayed.

2. Run **interface** *interface-type interface-number*

The interface view is displayed.

3. Use either of the following methods to create a static PW, a dynamic PW, or PW switching.

- To create a static PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value tunnel-policy tnl-policy-name [ [ control-word | no-control-word ] | [ raw | tagged ] | secondary ] *
```

 NOTE

When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.

- To create a static PW, run:

```
mpls static-l2vc { { destination ip-address | pw-template pw-template-name vc-id } * | destination ip-address [ vc-id ] } transmit-vpn-label transmit-label-value receive-vpn-label receive-label-value tunnel-policy tnl-policy-name [ [ control-word | no-control-word ] | [ raw | tagged ] | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation number | tdm-sequence-number | secondary ] *
```

 NOTE

- When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.
- When the AC interfaces are serial interfaces, CE1/PRI interfaces, or E1-F interfaces, you can specify the parameters **idle-code**, **jitter-buffer**, **tdm-encapsulation**, and **tdm-sequence-number**.

- To create a dynamic PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id tunnel-policy policy-name [ [ control-word | no-control-word ] | [ raw | tagged ] | mtu mtu-value | secondary ] *
```

 NOTE

When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.

- To create a dynamic PW, run:

```
mpls l2vc { ip-address | pw-template pw-template-name } * vc-id tunnel-policy policy-name [ [ control-word | no-control-word ] | [ raw | tagged ] | mtu mtu-value | idle-code idle-code-value | jitter-buffer depth | tdm-encapsulation-number number | tdm-sequence-number | secondary ] *
```

 NOTE

- When the AC interfaces are Ethernet interfaces, you can specify the parameters **raw** and **tagged**.
- When the AC interfaces are serial interfaces, CE1/PRI interfaces, or E1-F interfaces, you can specify the parameters **idle-code**, **jitter-buffer**, **tdm-encapsulation-number**, and **tdm-sequence-number**.

- To create static PW switching, run:

```
mpls switch-l2vc ip-address vc-id trans trans-label rcv received-label [ tunnel-policy policy-name ] between ip-address vc-id trans trans-label rcv received-label [ tunnel-policy policy-name ] encapsulation encapsulation-type [ control-word [ cc { alert | cw } * cv lsp-ping ] | [ no-control-word ] [ cc alert cv lsp-ping ] ] [ control-word-transparent ]
```

- To create dynamic PW switching, run:

```
mpls switch-l2vc ip-address vc-id [ tunnel-policy policy-name ] between ip-address vc-id [ tunnel-policy policy-name ] encapsulation encapsulation-type [ control-word-transparent ]
```

- To create mixed PW switching, run:

```
mpls switch-l2vc ip-address vc-id [ tunnel-policy policy-name ] between
ip-address vc-id trans trans-label recv received-label [ tunnel-policy
policy-name ] encapsulation encapsulation-type [ mtu mtu-value ]
[ control-word [ cc { alert | cw } * cv lsp-ping ] | [ no-control-word ]
[ cc alert cv lsp-ping ] ] [ timeslotnum timeslotnum ] [ tdm-
encapsulation number ] [ control-word-transparent ]
```

- To create mixed PW switching, run:

```
mpls switch-l2vc ip-address vc-id [ tunnel-policy policy-name ] between
ip-address vc-id trans trans-label recv received-label [ tunnel-policy
policy-name ] encapsulation encapsulation-type [ mtu mtu-value ]
[ control-word [ cc { alert | cw } * cv lsp-ping ] | [ no-control-word ]
[ cc alert cv lsp-ping ] ] [ control-word-transparent ]
```

---End

Verifying the Configuration

After configuring a tunnel policy and applying it to PWE3, you can check information about the tunnel policy applied to the PWE3 and tunnels in the system.

- Run the **display tunnel-info** { **tunnel-id** *tunnel-id* | **all** | **statistics** [**slots**] } command to check information about tunnels in the system.
- Run the **display tunnel-policy** [*tunnel-policy-name*] command to check the configurations of tunnel policies.
- Run the **display mpls static-l2vc** [*vc-id* | **interface** *interface-type interface-number* | **state** { **down** | **up** }] command to check the information about static VCs.
- Run the **display mpls l2vc** [*vc-id* | **interface** *interface-type interface-number* | **remote-info** [*vc-id* | **verbose**] | **state** { **down** | **up** }] command to check the information about virtual circuits in LDP mode.

12.9 Maintaining PWE3

This section describes how to maintain PWE3, including verifying connectivity of a PW and locating a fault on a PW.

12.9.1 Verifying Connectivity of a PW

Context

Before using the **ping vc** and **tracert vc** commands to check connectivity of a PW, ensure that the PWE3 network is correctly configured.

- VCCV ping can be performed in control word channel mode or Label alert channel mode:
 - Control word channel: supports detection between UPEs.
 - Label alert channel: supports detection between CEs and per-hop detection between the UPE and SPE.

By default, VCCV in Label Alert mode is enabled. Before using the control word channel, run the **control-word** command to enable the control word function. VCCV in control word channel mode is then enabled.

When locating faults on the PW, you can use either VCCV in control word channel mode or normal mode.

Checking connectivity of the PW is not supported in the following situations:

- SPEs do not support the **ping vc** and **tracert vc** command (these commands are supported only by UPEs).
- Multiple users cannot run the command simultaneously. That is, the devices on the two ends cannot ping a VC at the same time. On a device serving as both a UPE and an SPE, if the PW serving as an SPE is performing VCCV ping, the PW serving as a UPE will be unable to perform VCCV ping. That is, two VCCV pings cannot be performed on a same device at the same time.
- The MTU check of the VC is not supported.

For an MH-PW, the local VC ID and VC type needs to be specified.

In the control word mode, if VC IDs are different, the VC ID of the remote UPE needs to be specified. In the MPLS Label Alert mode, the addresses of the remote peer SPEs or UPEs need to be specified.

Because a static PW does not support signaling negotiation, configurations of the UPE control word on both ends of the PW are different, with the control word being enabled on one end, but disabled on the other. When the MPLS Label Alert mode is enabled on both ends, the PW can be Up and the **ping vc** command can work. CEs, however, cannot communicate with each other because the control words are different.

Procedure

- Check the connectivity of the PW.
 - Control word channel
ping vc *pw-type pw-id* [**-c** *echo-number* | **-m** *time-value* | **-s** *data-bytes* | **-t** *timeout-value* | **-exp** *exp-value* | **-r** *reply-mode* | **-v**] * **control-word** [**remote** *remote-ip-address peer-pw-id* | **draft6**] * [**ttl** *tll-value*] [**uniform**]
To check the connectivity of a PW switching, run the following commands:
ping vc *pw-type pw-id* [**-c** *echo-number* | **-m** *time-value* | **-s** *data-bytes* | **-t** *timeout-value* | **-exp** *exp-value* | **-r** *reply-mode* | **-v**] * **control-word remote** *remote-ip-address peer-pw-id sender sender-address* [**tll** *tll-value*] [**uniform**]
 - Label Alert channel
ping vc *pw-type pw-id* [**-c** *echo-number* | **-m** *time-value* | **-s** *data-bytes* | **-t** *timeout-value* | **-exp** *exp-value* | **-r** *reply-mode* | **-v**] * **label-alert** [**no-control-word**] [**remote** *remote-ip-address* | **draft6**] * [**uniform**]
 - Normal mode
ping vc *pw-type pw-id* [**-c** *echo-number* | **-m** *time-value* | **-s** *data-bytes* | **-t** *timeout-value* | **-exp** *exp-value* | **-r** *reply-mode* | **-v**] * **normal** [**no-control-word**] [**remote** *remote-ip-address peer-pw-id*] [**tll** *tll-value*] [**uniform**]
- Locate a fault on the PW.
 - Control word channel
tracert vc *pw-type pw-id* [**-exp** *exp-value* | **-f** *first-ttl* | **-m** *max-ttl* | **-r** *reply-mode* | **-t** *timeout-value*] * **control-word** [**draft6**] [**full-lsp-path**] [**uniform**]

- ```
tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * control-word remote remote-ip-address [ptn-mode | full-lsp-path] [uniform]
```
- ```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * control-word remote remote-pw-id draft6 [ full-lsp-path ] [ uniform ]
```
- Label Alert channel

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * label-alert [ no-control-word ] [ remote remote-ip-address ] [ full-lsp-path ] [ draft6 ] [ uniform ]
```
 - Normal mode

```
tracert vc pw-type pw-id [ -exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value ] * normal [ no-control-word ] [ remote remote-ip-address ] [ full-lsp-path ] [ draft6 ] [ uniform ]
```

---End

12.9.2 Locating a Fault on a PW

Context

After PWE3 is configured, you can locate any PW faults. To locate a fault on a PW, configure basic PWE3 functions using a PW template, and then run the following commands on U-PEs.

Procedure

- Step 1** Run the **system-view** command to enter the system view of the U-PE.
- Step 2** Run the **pw-template pw-template-name** command to enter the PW template view.
- Step 3** Run the **control-word** command to enable the control word function.
- Step 4** Run either of the following commands to collect information about each LSR along the PW and information about the egress PE.
 - **tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * control-word [draft6] [full-lsp-path] [uniform]**
 - **tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * control-word remote remote-ip-address [ptn-mode | full-lsp-path] [uniform]**
 - **tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * control-word remote remote-pw-id draft6 [full-lsp-path] [uniform]**
 - **tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * label-alert [remote remote-ip-address] [full-lsp-path] [draft6]**
 - **tracert vc pw-type pw-id [-exp exp-value | -f first-ttl | -m max-ttl | -r reply-mode | -t timeout-value] * normal [remote remote-ip-address] [full-lsp-path] [draft6]**

When running the **tracert vc** command to locate a PW fault, pay attention to the following points:

- The SPEs do not support this command. You can run this command on UPEs only.
- You can use this command to tracet a single-segment PW and a multi-segment PW created using LDP.
- When tracet a multi-segment PW, specify the remote PW ID in addition to the local PW ID and PW type.

When the **tracert vc** command is run, the tracert operation is terminated in the following cases:

- The PE that initiates tracert receives an MPLS Echo Reply packet from the egress PE.
- The TTL in the label of the previous MPLS Echo Request packet sent by the PE that initiates tracert reaches the configured or default maximum number of hops.
- A user presses **Ctrl+C** on the PE that initiates tracert.

----End

12.10 Configuration Examples for PWE3

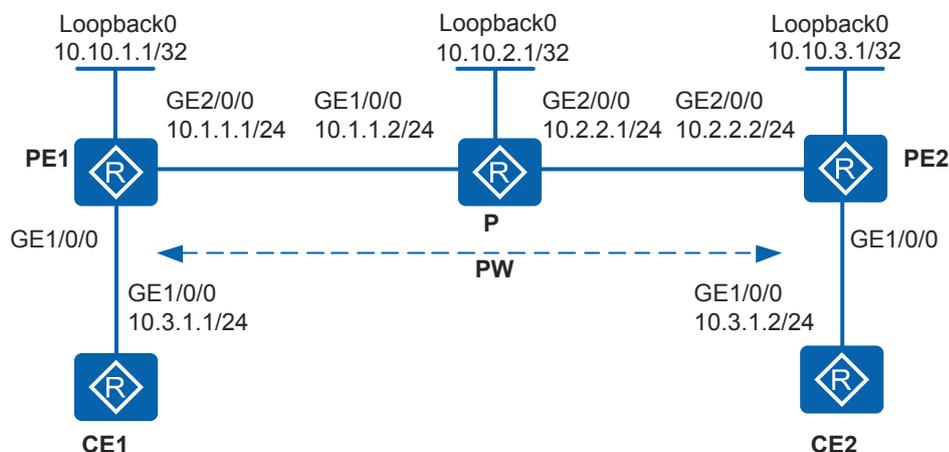
This section describes PWE3 configuration examples including the networking requirements, and configuration roadmap, configuration procedure, and configuration files.

12.10.1 Example for Configuring a Dynamic Single-Segment PW

Networking Requirements

As shown in **Figure 12-12**, the MPLS network of an ISP provides the L2VPN service for users. Many users connect to the MPLS network through PE1 and PE2, and users on the PEs change frequently. A proper VPN solution is required to provide secure VPN services for users, save network resources, and simplify configuration when new users connect to the network.

Figure 12-12 Networking diagram for configuring a dynamic single-segment PW (using an LSP tunnel)



Configuration Roadmap

Because users on the two PEs change frequently, manual configuration is inefficient and may cause configuration errors. In this scenario, the two PEs can set up a remote LDP session and use the LDP protocol to synchronize user information through a dynamic PW. Compared with Martini, PWE3 reduces the signaling cost and defines the multi-segment negotiation mode, making networking more flexible. PWE3 is recommended if network resources need to be saved.

The configuration roadmap is as follows:

1. Configure an IGP protocol on the backbone network so that backbone network devices can communicate.
2. Configure basic MPLS functions and establish LSP tunnels on the backbone network. Then establish the remote MPLS LDP peer relationship between the PEs at both ends of the PW.
3. Create MPLS L2VC connections on the PEs.

Procedure

- Step 1** Configure an IP address for each interface on the CEs, PEs, and the P according to [Figure 12-12](#).

Configure CE1. The configuration on PE1, P, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 10.3.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

- Step 2** Configure an IGP protocol and Loopback address on the MPLS backbone network.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 10.10.1.1 255.255.255.255
[PE1-LoopBack0] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 10.10.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command. The command output shows that PE1 and PE2 have learnt the routes to each other's Loopback0 interface through OSPF, and that PE1 and PE2 can ping each other.

- Step 3** Enable MPLS, and set up tunnels and remote LDP sessions.

Enable MPLS on the MPLS backbone network, and set up an LSP tunnel and remote LDP sessions between the PEs.

Configure PE1.

```
[PE1] mpls
[PE1-mpls] mpls ldp
[PE1-mpls-ldp] quit
```

```
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
[PE1] mpls ldp remote-peer 10.10.3.1
[PE1-mpls-ldp-remote-10.10.3.1] remote-ip 10.10.3.1
[PE1-mpls-ldp-remote-10.10.3.1] quit
```

Configure P.

```
[P] mpls
[P-mpls] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] ip address 10.1.1.2 255.255.255.0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] ip address 10.2.2.1 255.255.255.0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls
[PE2-mpls] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] ip address 10.2.2.2 255.255.255.0
[PE2-GigabitEthernet2/0/0] mpls
[PE2-GigabitEthernet2/0/0] mpls ldp
[PE2-GigabitEthernet2/0/0] quit
[PE2] mpls ldp remote-peer 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] remote-ip 10.10.1.1
[PE2-mpls-ldp-remote-10.10.1.1] quit
```

After the configuration is complete, run the **display mpls ldp session** command on the devices. The command output shows that LDP sessions are established between the PEs and between the P and PEs, and the session status is **Operational**.

Step 4 Create VCs.

Enable MPLS L2VPN on PE1 and PE2, and create a VC on each PE.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 10.10.3.1 100
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls l2vc 10.10.1.1 100
[PE2-GigabitEthernet1/0/0] quit
```

Step 5 Verify the configuration.

Run the following command on the PEs to check the L2VPN connections. The command output shows that an L2VC connection is set up and is in Up state.

The display on PE1 is used as an example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 100
VC type               : Ethernet
destination           : 10.10.3.1
local group ID        : 0          remote group ID      : 0
local VC label        : 1031       remote VC label       : 1030
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state : forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW           : unavailable
VCCV State            : up
manual fault          : not set
active state          : active
forwarding entry      : exist
link state            : up
local VC MTU          : 1500       remote VC MTU         : 1500
local VCCV            : alert ttl lsp-ping bfd
remote VCCV           : alert ttl lsp-ping bfd
local control word    : disable     remote control word   : disable
tunnel policy name    : --
PW template name      : --
primary or secondary  : primary
load balance type     : flow
Access-port           : false
Switchover Flag       : false
VC tunnel/token info  : 1 tunnels/tokens
  NO.0 TNL type       : lsp , TNL ID : 0x8
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 13 hours, 41 minutes, 24 seconds
up time               : 0 days, 0 hours, 46 minutes, 55 seconds
last change time      : 0 days, 0 hours, 46 minutes, 55 seconds
VC last up time       : 2013/12/02 00:16:31
VC total up time      : 0 days, 0 hours, 46 minutes, 55 seconds
CKey                  : 8
NKey                  : 7
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --
```

CE1 and CE2 can ping each other.

The display on CE1 is used as an example.

```
[CE1] ping 10.3.1.2
PING 10.3.1.2: 56 data bytes, press CTRL_C to break
  Reply from 10.3.1.2: bytes=56 Sequence=1 ttl=255 time=31 ms
  Reply from 10.3.1.2: bytes=56 Sequence=2 ttl=255 time=10 ms
  Reply from 10.3.1.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.3.1.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.3.1.2: bytes=56 Sequence=5 ttl=255 time=28 ms
--- 10.3.1.2 ping statistics ---
  5 packet(s) transmitted
```

```
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 2/15/31 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 10.3.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 10.10.1.1
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 10.10.3.1
 remote-ip 10.10.3.1
#
interface GigabitEthernet1/0/0
 mpls l2vc 10.10.3.1 100
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 10.10.1.1 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 10.10.1.1 0.0.0.0
  network 10.1.1.0 0.0.0.255
#
return
```

- Configuration file of the P

```
#
sysname P
#
mpls lsr-id 10.10.2.1
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 10.2.2.1 255.255.255.0
 mpls
 mpls ldp
#
```

```
interface LoopBack0
ip address 10.10.2.1 255.255.255.255
#
ospf 1
area 0.0.0.0
network 10.10.2.1 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 10.10.3.1
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 10.10.1.1
remote-ip 10.10.1.1
#
interface GigabitEthernet1/0/0
mpls l2vc 10.10.1.1 100
#
interface GigabitEthernet2/0/0
ip address 10.2.2.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 10.10.3.1 255.255.255.255
#
ospf 1
area 0.0.0.0
network 10.10.3.1 0.0.0.0
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of CE2

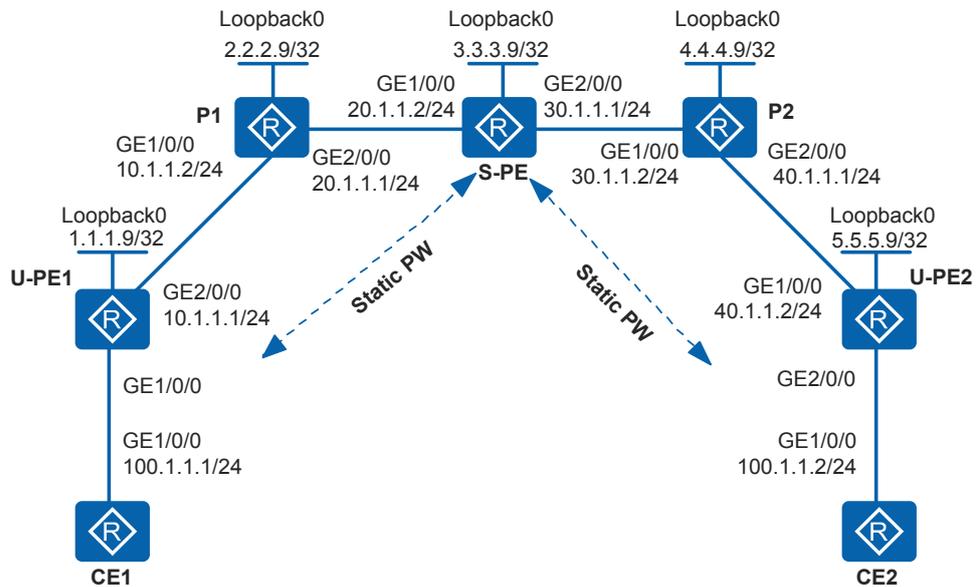
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 10.3.1.2 255.255.255.0
#
return
```

12.10.2 Example for Configuring a Static Multi-Segment PW

Networking Requirements

As shown in [Figure 12-13](#), sites of an enterprise at different geographical locations connect to the MPLS network of an ISP through CE1 and CE2. The S-PE has powerful functions, and U-PE1 and U-PE2 function as access devices and cannot directly establish remote LDP sessions. To simplify configuration, the enterprise hopes that the two CEs communicate with each other like on a LAN. That is, data packets of users traverse the ISP network without being modified by the PEs. The enterprise will not increase sites in the future and wants to use exclusive VPN resources of the ISP to protect user data security.

Figure 12-13 Networking diagram for configuring a static multi-segment PW



Configuration Roadmap

Because the enterprise will not increase sites in the future and wants to use exclusive VPN resources, you can configure a static PW to meet the customer requirements. To use hierarchical networking, configure a static multi-segment PW.

The configuration roadmap is as follows:

1. Configure a common routing protocol on the backbone network so that backbone network devices can communicate.
2. Configure basic MPLS functions and establish LSPs on the backbone network.
3. Establish static MPLS L2VC connections on U-PEs.
4. Configure PW switching on the S-PE for a multi-segment PW.

Procedure

Step 1 Configure an IP address for each interface on the devices according to [Figure 12-13](#).

Configure CE1. The configuration on U-PE1, P1, S-PE, P2, U-PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol and Loopback address on the MPLS backbone network.

Configure U-PE1. The configuration on P1, S-PE, P2, and U-PE2 is similar to the configuration on U-PE1 and is not mentioned here.

```
[U-PE1] interface loopback 0
[U-PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[U-PE1-LoopBack0] quit
[U-PE1] ospf 1
[U-PE1-ospf-1] area 0
[U-PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[U-PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[U-PE1-ospf-1-area-0.0.0.0] quit
[U-PE1-ospf-1] quit
```

Step 3 Configure basic MPLS functions and set up LSP tunnels.

Configure basic MPLS functions on the MPLS backbone network, and set up LSP tunnels between U-PE1 and S-PE, and between SPE and U-PE2. U-PE1 is used as an example. The configurations of other devices are similar to the configuration of U-PE1 and are not mentioned here.

Configure U-PE1.

```
[U-PE1] mpls lsr-id 1.1.1.9
[U-PE1] mpls
[U-PE1-mpls] mpls ldp
[U-PE1-mpls-ldp] quit
[U-PE1] interface gigabitEthernet 2/0/0
[U-PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[U-PE1-GigabitEthernet2/0/0] mpls
[U-PE1-GigabitEthernet2/0/0] mpls ldp
[U-PE1-GigabitEthernet2/0/0] quit
```

Step 4 Create VCs.

Enable MPLS L2VPN on U-PE1, U-PE2, and S-PE, and set up VCs on U-PE1 and U-PE2.

Configure U-PE1.

```
[U-PE1] mpls l2vpn
[U-PE1-l2vpn] quit
[U-PE1] pw-template pwt
[U-PE1-pw-template-pwt] peer-address 3.3.3.9
[U-PE1-pw-template-pwt] quit
[U-PE1] interface gigabitEthernet 1/0/0
[U-PE1-GigabitEthernet1/0/0] mpls static-l2vc pw-template pwt 100 transmit-vpn-label 100 receive-vpn-label 100
[U-PE1-GigabitEthernet1/0/0] quit
```

Configure S-PE.

```
[S-PE] mpls l2vpn
[S-PE-l2vpn] quit
[S-PE] mpls switch-l2vc 5.5.5.9 100 trans 200 recv 200 between 1.1.1.9 100 trans 100 recv 100 encapsulation ethernet
```

Configure U-PE2.

```
[U-PE2] mpls l2vpn
[U-PE2-l2vpn] quit
[U-PE2] pw-template pwt
[U-PE2-pw-template-pwt] peer-address 3.3.3.9
[U-PE2-pw-template-pwt] quit
[U-PE2] interface gigabitEthernet 1/0/0
[U-PE2-GigabitEthernet1/0/0] mpls static-l2vc pw-template pwt 100 transmit-vpn-label 200 receive-vpn-label 200
[U-PE2-GigabitEthernet1/0/0] quit
```

NOTE

The *transmit-vpn-label* configured on the U-PE must be the same as the *recv* label on the S-PE, and the *receive-vpn-label* configured on the U-PE must be the same as the *trans* label on the S-PE. Otherwise, CEs cannot communicate.

Step 5 Verify the configuration.

Run the following command on the PEs to check the L2VPN connections. The command output shows that an L2VC connection is set up and is in Up state.

The display on U-PE1 and the S-PE is used as an example.

```
[U-PE1] display mpls static-l2vc interface gigabitethernet 1/0/0
*Client Interface      : GigabitEthernet1/0/0 is up
AC Status              : up
VC State               : up
VC ID                  : 100
VC Type                : Ethernet
Destination            : 3.3.3.9
Transmit VC Label     : 100
Receive VC Label      : 100
Label Status          : 0
Token Status          : 0
Control Word           : Disable
VCCV Capabilty        : alert ttl lsp-ping bfd
active state          : active
Link State             : up
Tunnel Policy         : --
PW Template Name      : pwt
Main or Secondary     : Main
load balance type     : flow
Access-port           : false
VC tunnel/token info  : 1 tunnels/tokens
NO.0 TNL Type         : lsp , TNL ID : 0x4
Backup TNL Type       : lsp , TNL ID : 0x0
Create time           : 0 days, 4 hours, 38 minutes, 4 seconds
UP time               : 0 days, 0 hours, 12 minutes, 6 seconds
Last change time     : 0 days, 0 hours, 12 minutes, 6 seconds
VC last up time      : 2013/12/04 15:29:44
VC total up time     : 0 days, 0 hours, 12 minutes, 6 seconds
CKey                  : 2
NKey                  : 1
Diffserv Mode        : uniform
Service Class         : --
Color                 : --
DomainId              : --
Domain Name           : --
BFD for PW           : unavailable
[S-PE] display mpls switch-l2vc
Total Switch VC : 1, 1 up, 0 down

*Switch-l2vc type      : SVC<---->SVC
Peer IP Address       : 5.5.5.9, 1.1.1.9
VC ID                 : 100, 100
VC Type               : Ethernet
VC State              : up
In/Out Label          : 200/200, 100/100
InLabel Status        : 0 , 0
Control Word          : Disable, Disable
VCCV Capability        : alert ttl lsp-ping bfd , alert ttl lsp-ping bfd
Switch-l2vc tunnel info :
                        : 1 tunnels for peer 5.5.5.9
                        : NO.0 TNL Type : lsp , TNL ID : 0x10
                        : 1 tunnels for peer 1.1.1.9
                        : NO.0 TNL Type : lsp , TNL ID : 0xe
CKey                  : 8, 10
NKey                  : 7, 9
Tunnel policy         : --, --
Create time           : 0 days, 0 hours, 7 minutes, 19 seconds
UP time               : 0 days, 0 hours, 0 minutes, 34 seconds
Last change time     : 0 days, 0 hours, 0 minutes, 34 seconds
VC last up time      : 2013/12/01 22:31:43
VC total up time     : 0 days, 0 hours, 0 minutes, 34 seconds
```

CE1 and CE2 can ping each other successfully.

The display on CE1 is used as an example.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=188 ms
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=187 ms
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=187 ms
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=188 ms
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=188 ms

--- 100.1.1.2 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 187/187/188 ms
```

---End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of U-PE1

```
#
sysname U-PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
pw-template pwt
 peer-address 3.3.3.9
#
mpls ldp
#
interface GigabitEthernet1/0/0
 mpls static-l2vc pw-template pwt 100 transmit-vpn-label 100 receive-vpn-label 100
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 1.1.1.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
 network 10.1.1.0 0.0.0.255
 network 1.1.1.9 0.0.0.0
#
return
```

- Configuration file of P1

```
#
sysname P1
#
```

```
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 20.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 10.1.1.0 0.0.0.255
  network 20.1.1.0 0.0.0.255
  network 2.2.2.9 0.0.0.0
#
return
```

● Configuration file of S-PE

```
#
sysname S-PE
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls switch-l2vc 5.5.5.9 100 trans 200 recv 200 between 1.1.1.9 100 trans 100
recv 100 encapsulation ethernet
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 20.1.1.0 0.0.0.255
  network 30.1.1.0 0.0.0.255
  network 3.3.3.9 0.0.0.0
#
return
```

● Configuration file of P2

```
#
sysname P2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
```

```
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 40.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 4.4.4.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 4.4.4.9 0.0.0.0
  network 30.1.1.0 0.0.0.255
  network 40.1.1.0 0.0.0.255
#
return
```

- Configuration file of U-PE2

```
#
sysname U-PE2
#
mpls lsr-id 5.5.5.9
mpls
#
mpls l2vpn
#
pw-template pwt
 peer-address 3.3.3.9
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 40.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 mpls static-l2vc pw-template pwt 100 transmit-vpn-label 200 receive-vpn-label 200
#
interface LoopBack0
 ip address 5.5.5.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 5.5.5.9 0.0.0.0
  network 40.1.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

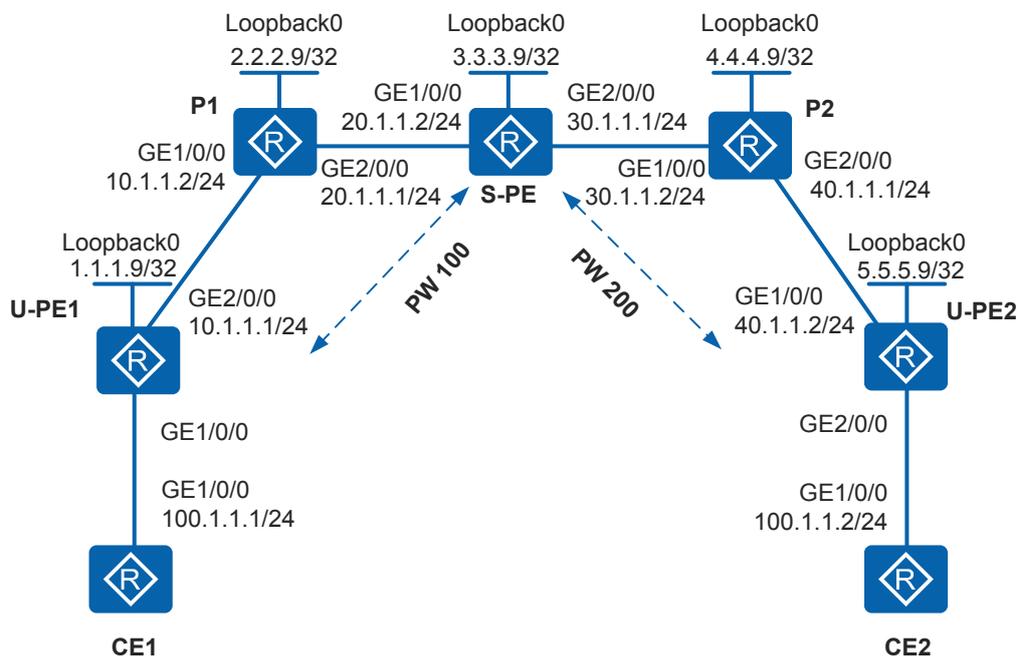
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.2 255.255.255.0
#
return
```

12.10.3 Example for Configuring a Dynamic Multi-Segment PW

Networking Requirements

As shown in [Figure 12-14](#), the MPLS network of an ISP provides the L2VPN service for users. The S-PE has powerful functions, and U-PE1 and U-PE2 function as access devices and cannot directly establish remote LDP sessions. Many users connect to the MPLS network through U-PE1 and U-PE2, and users on the U-PEs change frequently. A proper VPN solution is required to provide secure VPN services for users and simplify configuration and maintenance when new users connect to the network.

Figure 12-14 Networking diagram for configuring a dynamic multi-segment PW



Configuration Roadmap

Because the S-PE has powerful functions, and U-PE1 and U-PE2 cannot directly establish remote LDP sessions, you can configure a multi-segment PW and PW switching on the S-PE to meet the customer requirements. To simplify maintenance, configure a dynamic multi-segment PW.

The configuration roadmap is as follows:

1. Configure an IGP protocol on the backbone network so that backbone network devices can communicate.
2. Configure basic MPLS functions and establish LSPs on the backbone network. Establish remote MPLS LDP peer relationships between U-PE1 and the S-PE, and between U-PE2 and the S-PE.
3. Create PW templates and enable the control word function and LSP ping.
4. Configure a dynamic PW on the S-PE.

5. Configure PW switching on the S-PE.

Procedure

Step 1 Configure an IP address for each interface on the devices according to [Figure 12-14](#).

Configure CE1. The configuration on U-PE1, P1, S-PE, P2, U-PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol and Loopback address on the MPLS backbone network.

Configure U-PE1. The configuration on P1, S-PE, P2, and U-PE2 is similar to the configuration on U-PE1 and is not mentioned here.

```
[U-PE1] interface loopback 0
[U-PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[U-PE1-LoopBack0] quit
[U-PE1] ospf 1
[U-PE1-ospf-1] area 0
[U-PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[U-PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[U-PE1-ospf-1-area-0.0.0.0] quit
[U-PE1-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on the U-PEs, Ps or S-PE. The command output shows that these devices have learnt the routes of each other.

Step 3 Enable MPLS and set up LSP tunnels and remote LDP sessions.

Configure basic MPLS functions on the MPLS backbone network, and set up LSP tunnels and remote LDP sessions between U-PE1 and the S-PE, and between the S-PE and U-PE2.

Configure U-PE1.

```
[U-PE1] mpls lsr-id 1.1.1.9
[U-PE1] mpls
[U-PE1-mpls] quit
[U-PE1] mpls ldp
[U-PE1-mpls-ldp] quit
[U-PE1] interface gigabitethernet 2/0/0
[U-PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[U-PE1-GigabitEthernet2/0/0] mpls
[U-PE1-GigabitEthernet2/0/0] mpls ldp
[U-PE1-GigabitEthernet2/0/0] quit
[U-PE1] mpls ldp remote-peer 3.3.3.9
[U-PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[U-PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure P1.

```
[P1] mpls lsr-id 2.2.2.9
[P1] mpls
[P1-mpls] quit
[P1] mpls ldp
[P1-mpls-ldp] quit
[P1] interface gigabitethernet 1/0/0
[P1-GigabitEthernet1/0/0] mpls
[P1-GigabitEthernet1/0/0] mpls ldp
[P1-GigabitEthernet1/0/0] quit
```

```
[P1] interface gigabitethernet 2/0/0
[P1-GigabitEthernet2/0/0] mpls
[P1-GigabitEthernet2/0/0] mpls ldp
[P1-GigabitEthernet2/0/0] quit
```

Configure the S-PE.

```
[S-PE] mpls lsr-id 3.3.3.9
[S-PE] mpls
[S-PE-mpls] quit
[S-PE] mpls ldp
[S-PE-mpls-ldp] quit
[S-PE] interface gigabitethernet 1/0/0
[S-PE-GigabitEthernet1/0/0] mpls
[S-PE-GigabitEthernet1/0/0] mpls ldp
[S-PE-GigabitEthernet1/0/0] quit
[S-PE] interface gigabitethernet 2/0/0
[S-PE-GigabitEthernet2/0/0] mpls
[S-PE-GigabitEthernet2/0/0] mpls ldp
[S-PE-GigabitEthernet2/0/0] quit
[S-PE] mpls ldp remote-peer 1.1.1.9
[S-PE-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[S-PE-mpls-ldp-remote-1.1.1.9] quit
[S-PE] mpls ldp remote-peer 5.5.5.9
[S-PE-mpls-ldp-remote-5.5.5.9] remote-ip 5.5.5.9
[S-PE-mpls-ldp-remote-5.5.5.9] quit
```

Configure P2.

```
[P2] mpls lsr-id 4.4.4.9
[P2] mpls
[P2-mpls] quit
[P2] mpls ldp
[P2-mpls-ldp] quit
[P2] interface gigabitethernet 1/0/0
[P2-GigabitEthernet1/0/0] mpls
[P2-GigabitEthernet1/0/0] mpls ldp
[P2-GigabitEthernet1/0/0] quit
[P2] interface gigabitethernet 2/0/0
[P2-GigabitEthernet2/0/0] mpls
[P2-GigabitEthernet2/0/0] mpls ldp
[P2-GigabitEthernet2/0/0] quit
```

Configure U-PE2.

```
[U-PE2] mpls lsr-id 5.5.5.9
[U-PE2] mpls
[U-PE2-mpls] quit
[U-PE2] mpls ldp
[U-PE2-mpls-ldp] quit
[U-PE2] interface gigabitethernet 1/0/0
[U-PE2-GigabitEthernet1/0/0] mpls
[U-PE2-GigabitEthernet1/0/0] mpls ldp
[U-PE2-GigabitEthernet1/0/0] quit
[U-PE2] mpls ldp remote-peer 3.3.3.9
[U-PE2-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[U-PE2-mpls-ldp-remote-3.3.3.9] quit
```

After the configuration is complete, run the **display mpls ldp session** command on the U-PEs, Ps, or S-PE. The command output shows that the LDP sessions are established and the status is **Operational**. Run the **display mpls ldp peer** command. The command output shows that LDP peer relationships are established. Run the **display mpls lsp** command. The command output shows that LSPs are established.

Step 4 Create and configure PW templates.

Create PW templates on the U-PEs, and enable the control word function.

Configure U-PE1.

```
[U-PE1] mpls l2vpn
[U-PE1-l2vpn] quit
[U-PE1] pw-template pwt
[U-PE1-pw-template-pwt] peer-address 3.3.3.9
[U-PE1-pw-template-pwt] control-word
[U-PE1-pw-template-pwt] quit
```

Configure U-PE2.

```
[U-PE2] mpls l2vpn
[U-PE2-l2vpn] quit
[U-PE2] pw-template pwt
[U-PE2-pw-template-pwt] peer-address 3.3.3.9
[U-PE2-pw-template-pwt] control-word
[U-PE2-pw-template-pwt] quit
```

NOTE

You can also configure a dynamic PW without using the PW template. If the PW template is not used, PW connectivity cannot be verified and path information of the PW cannot be collected. That is, you cannot run the **ping vc** or **tracert vc** command.

Step 5 Create VCs.

Enable MPLS L2VPN on U-PE1, U-PE2, and the S-PE.

Configure dynamic PWs on U-PEs, and configure PW switching on the S-PE.

Configure U-PE1.

```
[U-PE1] interface gigabitethernet 1/0/0
[U-PE1-GigabitEthernet1/0/0] mpls l2vc pw-template pwt 100
[U-PE1-GigabitEthernet1/0/0] quit
```

Configure the S-PE.

```
[S-PE] mpls l2vpn
[S-PE-l2vpn] quit
[S-PE] mpls switch-l2vc 1.1.1.9 100 between 5.5.5.9 200 encapsulation ethernet
```

Configure U-PE2.

```
[U-PE2] interface gigabitethernet 2/0/0
[U-PE2-GigabitEthernet2/0/0] mpls l2vc pw-template pwt 200
[U-PE2-GigabitEthernet2/0/0] quit
```

Step 6 Verify the configuration.

1. View the PWE3 connection.

View the L2VPN connection on the U-PEs and S-PE. The command output shows that an L2VC is set up and the VC status is Up.

The display on U-PE1 is used as an example.

```
[U-PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 100
VC type               : Ethernet
destination           : 3.3.3.9
local group ID        : 0          remote group ID      : 0
local VC label        : 1028       remote VC label      : 1032
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : forwarding
```

```

local status code      : 0x0
remote AC OAM state    : up
remote PSN OAM state   : up
remote forwarding state: forwarding
remote status code     : 0x0
ignore standby state   : no
BFD for PW             : unavailable
VCCV State             : up
manual fault           : not set
active state           : active
forwarding entry       : exist
link state              : up
local VC MTU           : 1500          remote VC MTU           : 1500
local VCCV              : cw alert ttl lsp-ping bfd
remote VCCV            : cw alert ttl lsp-ping bfd
local control word     : enable       remote control word    : enable
tunnel policy name     : --
PW template name       : pwt
primary or secondary   : primary
load balance type      : flow
Access-port            : false
Switchover Flag        : false
VC tunnel/token info   : 1 tunnels/tokens
  NO.0 TNL type        : lsp , TNL ID : 0x4
  Backup TNL type      : lsp , TNL ID : 0x0
create time            : 0 days, 0 hours, 9 minutes, 38 seconds
up time                : 0 days, 0 hours, 0 minutes, 50 seconds
last change time       : 0 days, 0 hours, 0 minutes, 50 seconds
VC last up time        : 2013/12/04 16:09:45
VC total up time       : 0 days, 0 hours, 0 minutes, 50 seconds
CKey                   : 4
NKey                   : 3
PW redundancy mode     : frr
AdminPw interface      : --
AdminPw link state     : --
Diffserv Mode          : uniform
Service Class          : --
Color                  : --
DomainId               : --
Domain Name            : --
  
```

Check the L2VC status on the S-PE.

```

[S-PE] display mpls switch-l2vc
Total Switch VC : 1, 1 up, 0 down

*Switch-l2vc type      : LDP<---->LDP
Peer IP Address        : 5.5.5.9, 1.1.1.9
VC ID                  : 200, 100
VC Type                : Ethernet
VC State               : up
VC StatusCode          :
  PSN | OAM | FW |      | PSN | OAM | FW |
-Local VC : | UP | UP | UP |      | UP | UP | UP |
-Remote VC : | UP | UP | UP |      | UP | UP | UP |
Session State          : up, up
Local/Remote Label     : 1031/1028, 1032/1028
InLabel Status         : 0 , 0
Local/Remote MTU       : 1500/1500, 1500/1500
Local/Remote Control Word : Enable/Enable, Enable/Enable
Local/Remote VCCV Capability : cw alert ttl lsp-ping bfd /cw alert ttl
lsp-ping bfd , cw alert ttl lsp-ping bfd /cw alert ttl lsp-ping bfd
Switch-l2vc tunnel info :
  1 tunnels for peer 5.5.5.9
  NO.0 TNL Type : lsp , TNL ID : 0x10
  1 tunnels for peer 1.1.1.9
  NO.0 TNL Type : lsp , TNL ID : 0xe
CKey                   : 14, 16
NKey                   : 13, 15
Tunnel policy          : --, --
Control-Word transparent : NO
Create time            : 0 days, 0 hours, 6 minutes, 39 seconds
  
```

```
UP time : 0 days, 0 hours, 5 minutes, 16 seconds
Last change time : 0 days, 0 hours, 5 minutes, 16 seconds
VC last up time : 2013/12/01 23:02:39
VC total up time : 0 days, 0 hours, 5 minutes, 16 seconds
```

2. Detect connectivity of the PW.

Run the **ping vc** command on the U-PEs. The command output shows that connectivity of the PW is normal. The display on U-PE1 is used as an example.

```
[U-PE1] ping vc ethernet 100 control-word remote 5.5.5.9 200
Reply from 5.5.5.9: bytes=100 Sequence=1 time = 740 ms
Reply from 5.5.5.9: bytes=100 Sequence=2 time = 90 ms
Reply from 5.5.5.9: bytes=100 Sequence=3 time = 160 ms
Reply from 5.5.5.9: bytes=100 Sequence=4 time = 130 ms
Reply from 5.5.5.9: bytes=100 Sequence=5 time = 160 ms

--- FEC: FEC 128 PSEUDOWIRE (NEW). Type = vlan, ID = 100 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 90/256/740 ms
```

3. Check connectivity between the CEs and information about the paths between the CEs.

CE1 and CE2 can ping each other.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=180 ms
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=120 ms
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=160 ms
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=160 ms
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=130 ms

--- 100.1.1.2 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 120/150/180 ms
```

On CE1, perform the **tracert** operation.

```
[CE1] tracert 100.1.1.2
traceroute to 100.1.1.2(100.1.1.2) max hops: 30 ,packet length: 40,press
CTRL_C to break
1 100.1.1.2 250 ms 220 ms 130 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of U-PE1

```
#
sysname U-PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
pw-template pwt
```

```
peer-address 3.3.3.9
control-word
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
mpls l2vc pw-template pwt 100
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 1.1.1.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 10.1.1.0 0.0.0.255
network 1.1.1.9 0.0.0.0
#
return
```

● Configuration file of P1

```
#
sysname P1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 10.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 20.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 10.1.1.0 0.0.0.255
network 20.1.1.0 0.0.0.255
#
return
```

● Configuration file of the S-PE

```
#
sysname S-PE
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls switch-l2vc 1.1.1.9 100 between 5.5.5.9 200 encapsulation ethernet
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
```

```
remote-ip 1.1.1.9
#
mpls ldp remote-peer 5.5.5.9
remote-ip 5.5.5.9
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 3.3.3.9 0.0.0.0
  network 20.1.1.0 0.0.0.255
  network 30.1.1.0 0.0.0.255
#
return
```

● Configuration file of P2

```
#
sysname P2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 40.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 4.4.4.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 4.4.4.9 0.0.0.0
  network 30.1.1.0 0.0.0.255
  network 40.1.1.0 0.0.0.255
#
return
```

● Configuration file of U-PE2

```
#
sysname U-PE2
#
mpls lsr-id 5.5.5.9
mpls
#
mpls l2vpn
#
pw-template pwt
 peer-address 3.3.3.9
 control-word
#
mpls ldp
```

```
#
mpls ldp remote-peer 3.3.3.9
remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
ip address 40.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
mpls l2vc pw-template pwt 200
#
interface LoopBack0
ip address 5.5.5.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 5.5.5.9 0.0.0.0
network 40.1.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
```

12.10.4 Example for Configuring a Mixed Multi-Segment PW

Networking Requirements

As shown in [Figure 12-15](#), the MPLS network of an ISP provides the L2VPN service for users. The S-PE has powerful functions, and U-PE1 and U-PE2 (U-PE2 supports only static PWs) function as access devices and cannot directly establish remote LDP session. Many users connect to the MPLS network through U-PE1 and U-PE2, and users on the U-PEs change frequently. A proper VPN solution is required to provide secure VPN services for users and simplify configuration and maintenance when new users connect to the network.

Step 2 Configure an IGP protocol and Loopback address on the MPLS backbone network.

Configure U-PE1. The configuration on P1, S-PE, P2, and U-PE2 is similar to the configuration on U-PE1 and is not mentioned here.

```
[U-PE1] interface loopback 0
[U-PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[U-PE1-LoopBack0] quit
[U-PE1] ospf 1
[U-PE1-ospf-1] area 0
[U-PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[U-PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[U-PE1-ospf-1-area-0.0.0.0] quit
[U-PE1-ospf-1] quit
```

Step 3 Enable MPLS, set up tunnels, and set up a remote LDP session between U-PE1 and the S-PE.

Configure basic MPLS functions and set up tunnels on the MPLS backbone network. In this example, the LSP tunnel is used.

You need to set up a remote LDP session between U-PE1 and the S-PE. U-PE1 is used as an example.

Configure U-PE1.

```
[U-PE1] mpls lsr-id 1.1.1.9
[U-PE1] mpls
[U-PE1-mpls] quit
[U-PE1] mpls ldp
[U-PE1-mpls-ldp] quit
[U-PE1] interface gigabitethernet 2/0/0
[U-PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[U-PE1-GigabitEthernet2/0/0] mpls
[U-PE1-GigabitEthernet2/0/0] mpls ldp
[U-PE1-GigabitEthernet2/0/0] quit
[U-PE1] mpls ldp remote-peer 3.3.3.9
[U-PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[U-PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure P1

```
[P1] mpls lsr-id 2.2.2.9
[P1] mpls
[P1-mpls] quit
[P1] mpls ldp
[P1-mpls-ldp] quit
[P1] interface gigabitethernet 1/0/0
[P1-GigabitEthernet1/0/0] mpls
[P1-GigabitEthernet1/0/0] mpls ldp
[P1-GigabitEthernet1/0/0] quit
[P1] interface gigabitethernet 2/0/0
[P1-GigabitEthernet2/0/0] mpls
[P1-GigabitEthernet2/0/0] mpls ldp
[P1-GigabitEthernet2/0/0] quit
```

Configure the S-PE.

```
[S-PE] mpls lsr-id 3.3.3.9
[S-PE] mpls
[S-PE-mpls] quit
[S-PE] mpls ldp
[S-PE-mpls-ldp] quit
[S-PE] interface gigabitethernet 1/0/0
[S-PE-GigabitEthernet1/0/0] mpls
[S-PE-GigabitEthernet1/0/0] mpls ldp
[S-PE-GigabitEthernet1/0/0] quit
[S-PE] interface gigabitethernet 2/0/0
[S-PE-GigabitEthernet2/0/0] mpls
```

```
[S-PE-GigabitEthernet2/0/0] mpls ldp
[S-PE-GigabitEthernet2/0/0] quit
[S-PE] mpls ldp remote-peer 1.1.1.9
[S-PE-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[S-PE-mpls-ldp-remote-1.1.1.9] quit
[S-PE] mpls ldp remote-peer 5.5.5.9
[S-PE-mpls-ldp-remote-5.5.5.9] remote-ip 5.5.5.9
[S-PE-mpls-ldp-remote-5.5.5.9] quit
```

Configure P2

```
[P2] mpls lsr-id 4.4.4.9
[P2] mpls
[P2-mpls] quit
[P2] mpls ldp
[P2-mpls-ldp] quit
[P2] interface gigabitethernet 1/0/0
[P2-GigabitEthernet1/0/0] mpls
[P2-GigabitEthernet1/0/0] mpls ldp
[P2-GigabitEthernet1/0/0] quit
[P2] interface gigabitethernet 2/0/0
[P2-GigabitEthernet2/0/0] mpls
[P2-GigabitEthernet2/0/0] mpls ldp
[P2-GigabitEthernet2/0/0] quit
```

Configure U-PE2

```
[U-PE2] mpls lsr-id 5.5.5.9
[U-PE2] mpls
[U-PE2-mpls] quit
[U-PE2] mpls ldp
[U-PE2-mpls-ldp] quit
[U-PE2] interface gigabitethernet 1/0/0
[U-PE2-GigabitEthernet1/0/0] mpls
[U-PE2-GigabitEthernet1/0/0] mpls ldp
[U-PE2-GigabitEthernet1/0/0] quit
[U-PE2] mpls ldp remote-peer 3.3.3.9
[U-PE2-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[U-PE2-mpls-ldp-remote-3.3.3.9] quit
```

Step 4 Create VCs.

Enable MPLS L2VPN on U-PE1, U-PE2, and the S-PE.

Configure a dynamic VC on U-PE1 and a static VC on U-PE2, and configure mixed PW switching on the S-PE.

Configure U-PE1.

```
[U-PE1] mpls l2vpn
[U-PE1-l2vpn] quit
[U-PE1] interface gigabitethernet 1/0/0
[U-PE1-GigabitEthernet1/0/0] mpls l2vc 3.3.3.9 100
[U-PE1-GigabitEthernet1/0/0] quit
```

NOTE

When you configure mixed PW switching, *ip-address vc-id* before **between** specifies the VC ID of a dynamic PW and *ip-address vc-id* after **between** specifies the VC ID of a static PW. The two values cannot be interchanged.

Configure the S-PE.

```
[S-PE] mpls l2vpn
[S-PE-l2vpn] quit
[S-PE] mpls switch-l2vc 1.1.1.9 100 between 5.5.5.9 200 trans 200 recv 100
encapsulation ethernet
```

Configure U-PE2.

```
[U-PE2] mpls l2vpn
[U-PE2-l2vpn] quit
[U-PE2] pw-template pwt
[U-PE2-pw-template-pwt] peer-address 3.3.3.9
[U-PE2-pw-template-pwt] quit
[U-PE2] interface gigabitethernet 2/0/0
[U-PE2-GigabitEthernet2/0/0] mpls static-l2vc pw-template pwt 200 transmit-vpn-label 100 receive-vpn-label 200
[U-PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

View information about L2VPN connections on the PEs. The command output shows that an L2VC is set up and the VC status is Up.

The display on U-PE1 and the S-PE is used as an example.

```
[U-PE1] display mpls l2vc interface gigabitethernet 1/0/0
*client interface      : GigabitEthernet1/0/0 is up
Administrator PW      : no
session state         : up
AC status             : up
VC state              : up
Label state           : 0
Token state           : 0
VC ID                 : 100
VC type               : Ethernet
destination           : 3.3.3.9
local group ID        : 0          remote group ID      : 0
local VC label        : 1029       remote VC label       : 1033
local AC OAM State    : up
local PSN OAM State   : up
local forwarding state : forwarding
local status code     : 0x0
remote AC OAM state   : up
remote PSN OAM state  : up
remote forwarding state : forwarding
remote status code    : 0x0
ignore standby state  : no
BFD for PW            : unavailable
VCCV State            : up
manual fault          : not set
active state          : active
forwarding entry      : exist
link state             : up
local VC MTU          : 1500       remote VC MTU        : 1500
local VCCV             : alert ttl lsp-ping bfd
remote VCCV           : alert ttl lsp-ping bfd
local control word    : disable    remote control word   : disable
tunnel policy name    : --
PW template name      : --
primary or secondary  : primary
load balance type     : flow
Access-port           : false
Switchover Flag       : false
VC tunnel/token info  : 1 tunnels/tokens
  NO.0 TNL type       : lsp , TNL ID : 0x4
  Backup TNL type     : lsp , TNL ID : 0x0
create time           : 0 days, 0 hours, 3 minutes, 32 seconds
up time               : 0 days, 0 hours, 2 minutes, 36 seconds
last change time      : 0 days, 0 hours, 2 minutes, 36 seconds
VC last up time       : 2013/12/04 16:32:08
VC total up time      : 0 days, 0 hours, 2 minutes, 36 seconds
CKey                  : 6
NKey                  : 5
PW redundancy mode    : frr
AdminPw interface     : --
AdminPw link state    : --
Diffserv Mode         : uniform
```

```

Service Class      : --
Color             : --
DomainId         : --
Domain Name      : --
BFD for PW       : unavailable
[S-PE] display mpls switch-l2vc
Total Switch VC : 1, 1 up, 0 down

*Switch-l2vc type      : LDP<---->SVC
Peer IP Address      : 1.1.1.9, 5.5.5.9
VC ID                : 100, 200
VC Type              : Ethernet
VC State             : up
Session State        : up, None
Local(In)/Remote(Out) Label : 1033/1029, 100/200
InLabel Status       : 0, 0
Local/Remote MTU     : 1500/1500, 1500
Local/Remote Control Word : Disable/Disable, Disable
Local/Remote VCCV Capability : alert ttl lsp-ping bfd /alert ttl lsp-ping bfd ,
alert ttl lsp-ping bfd
Switch-l2vc tunnel info :
                        : 1 tunnels for peer 1.1.1.9
                        : NO.0 TNL Type : lsp , TNL ID : 0xe
                        : 1 tunnels for peer 5.5.5.9
                        : NO.0 TNL Type : lsp , TNL ID : 0x10
CKey                 : 18, 20
NKey                  : 17, 19
Tunnel policy        : --, --
Create time          : 0 days, 0 hours, 6 minutes, 8 seconds
UP time              : 0 days, 0 hours, 6 minutes, 7 seconds
Last change time     : 0 days, 0 hours, 6 minutes, 7 seconds
VC last up time      : 2013/12/01 23:25:03
VC total up time     : 0 days, 0 hours, 6 minutes, 7 seconds
  
```

CE1 and CE2 can ping each other successfully.

The display on CE1 is used as an example.

```

[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=270 ms
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=220 ms
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=190 ms
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=190 ms
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=160 ms

--- 100.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 160/206/270 ms
  
```

----End

Configuration Files

- Configuration file of CE1

```

#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
  
```

- Configuration file of U-PE1

```

#
sysname U-PE1
  
```

```
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
  remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
  mpls l2vc 3.3.3.9 100
#
interface GigabitEthernet2/0/0
  ip address 10.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack0
  ip address 1.1.1.9 255.255.255.255
#
ospf 1
  area 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 1.1.1.9 0.0.0.0
#
return
```

● Configuration file of P1

```
#
  sysname P1
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
  ip address 10.1.1.2 255.255.255.0
  mpls
  mpls ldp
#
interface GigabitEthernet2/0/0
  ip address 20.1.1.1 255.255.255.0
  mpls
  mpls ldp
#
interface LoopBack0
  ip address 2.2.2.9 255.255.255.255
#
ospf 1
  area 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 20.1.1.0 0.0.0.255
    network 2.2.2.9 0.0.0.0
#
return
```

● Configuration file of the S-PE

```
#
  sysname S-PE
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
  mpls switch-l2vc 1.1.1.9 100 between 5.5.5.9 200 trans 200 recv 100
  encapsulation ethernet
```

```
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
remote-ip 1.1.1.9
#
mpls ldp remote-peer 5.5.5.9
remote-ip 5.5.5.9
#
interface GigabitEthernet1/0/0
ip address 20.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 30.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 20.1.1.0 0.0.0.255
network 30.1.1.0 0.0.0.255
network 3.3.3.9 0.0.0.0
#
return
```

● Configuration file of P2

```
#
sysname P2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 30.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 40.1.1.1 255.255.255.0
mpls
mpls ldp
#
interface LoopBack0
ip address 4.4.4.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 4.4.4.9 0.0.0.0
network 30.1.1.0 0.0.0.255
network 40.1.1.0 0.0.0.255
#
return
```

● Configuration file of U-PE2

```
#
sysname U-PE2
#
mpls lsr-id 5.5.5.9
mpls
#
mpls l2vpn
#
pw-template pwt
```

```
peer-address 3.3.3.9
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
ip address 40.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
mpls static-l2vc pw-template pwt 200 transmit-vpn-label 100 receive-vpn-label 200
#
interface LoopBack0
ip address 5.5.5.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 5.5.5.9 0.0.0.0
network 40.1.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

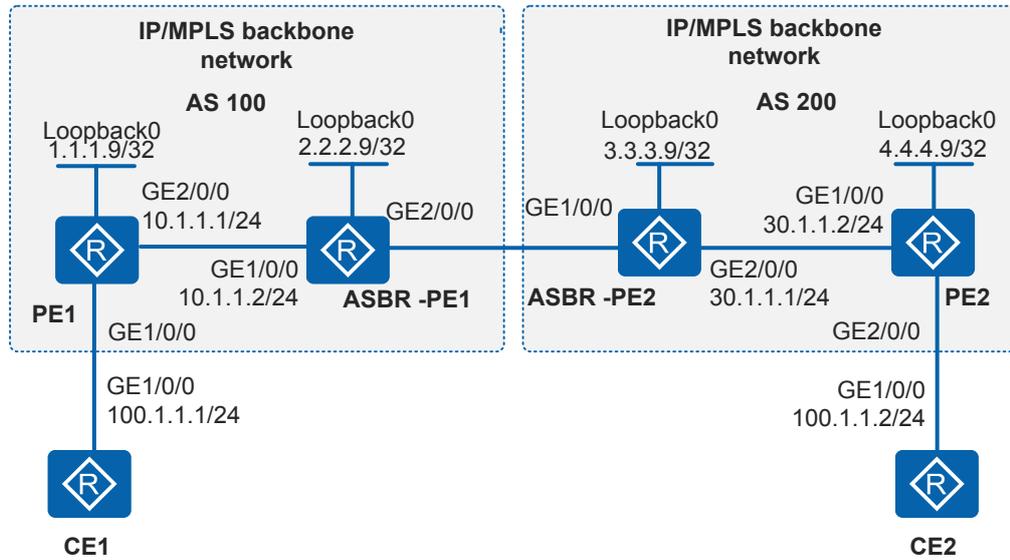
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
```

12.10.5 Example for Configuring Inter-AS PWE3 Option A

Networking Requirements

As shown in [Figure 12-16](#), the MPLS network of an ISP provides the L2VPN service for users. PE1 becomes to AS 100 and PE2 belongs to AS 200. Many users connect to the MPLS network through PE1 and PE2, and many new users will connect to the PEs in the future. A proper VPN solution is required to provide secure VPN services for users, save network resources, and simplify configuration when new users connect to the network.

Figure 12-16 Networking diagram for configuring inter-AS PWE3 Option A



MPLS backbone networks in the same AS use IS-IS as the IGP protocol.

Configuration Roadmap

The PEs connect to different ASs (AS 100 and AS 200) of the ISP, so an inter-AS VPN solution is required. To simplify configuration when new users connect to the network and save network resources, PWE3 Option A is recommended to meet the customer requirements.

The configuration roadmap is as follows:

1. Run an IGP protocol on the backbone network so that devices in an AS can communicate.
2. Configure basic MPLS functions on the backbone network and establish a dynamic LSP between the PE and ASBR-PE in the same AS. Establish a remote LDP session if the PE and ASBR-PE are not directly connected.
3. Establish an MPLS L2VC between the PE and ASBR-PE in the same AS.

Procedure

Step 1 Configure an IP address for each interface on the devices according to [Figure 12-16](#). CE1 is used as an example.

Configure CE1. The configuration on PE1, ASBR-PE1, ASBR-PE2, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol and Loopback address on the MPLS backbone network.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack0] quit
[PE1] isis 1
[PE1-isis-1] network-entity 10.0000.0000.0001.00
[PE1-isis-1] quit
```

After the configuration is complete, the IS-IS neighbor relationship can be established between the ASBR-PE and PE in the same AS. Run the **display isis peer** command. The command output shows that the neighbor relationship is Up.

Run the **display ip routing-table** command. The command output shows that the PE and ASBR-PE in the same AS can learn the routes to the loopback interface of each other.

The ASBR-PE and PE in the same AS can ping each other successfully.

Step 3 Enable MPLS and configure a dynamic LSP.

Configure basic MPLS functions on the MPLS backbone network. Establish a dynamic LDP LSP between the PE and ASBR-PE in the same AS.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

After this step is performed, an LSP tunnel is established between the PE and ASBR-PE in the same AS.

Step 4 Configure MPLS L2VCs.

Configure the L2VC on the PE and ASBR-PE and connect the PE to the CE.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls l2vc 2.2.2.9 100
[PE1-GigabitEthernet1/0/0] quit
```

Configure ASBR-PE1.

```
[ASBR-PE1] mpls l2vpn
[ASBR-PE1-l2vpn] quit
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] mpls l2vc 1.1.1.9 100
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

Configure ASBR-PE2.

```
[ASBR-PE2] mpls l2vpn
[ASBR-PE2-l2vpn] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
```

```
[ASBR-PE2-GigabitEthernet1/0/0] mpls l2vc 4.4.4.9 100  
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn  
[PE2-l2vpn] quit  
[PE2] interface gigabitethernet 2/0/0  
[PE2-GigabitEthernet2/0/0] mpls l2vc 3.3.3.9 100  
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

Run the following command to check information about the L2VPN connection on the PEs. The command output shows that an L2VC is set up and the VC status is Up.

The display on PE1 is used as an example.

```
[PE1] display mpls l2vc interface gigabitethernet 1/0/0  
*client interface      : GigabitEthernet1/0/0 is up  
  Administrator PW    : no  
  session state       : up  
  AC status           : up  
  VC state            : up  
  Label state         : 0  
  Token state         : 0  
  VC ID               : 100  
  VC type             : Ethernet  
  destination         : 2.2.2.9  
  local group ID     : 0          remote group ID      : 0  
  local VC label     : 21505     remote VC label      : 21506  
  local AC OAM State : up  
  local PSN OAM State : up  
  local forwarding state : forwarding  
  local status code  : 0x0  
  remote AC OAM state : up  
  remote PSN OAM state : up  
  remote forwarding state: forwarding  
  remote status code : 0x0  
  ignore standby state : no  
  BFD for PW         : unavailable  
  VCCV State         : up  
  manual fault       : not set  
  active state       : active  
  forwarding entry   : exist  
  link state         : up  
  local VC MTU       : 1500     remote VC MTU        : 1500  
  local VCCV         : alert ttl lsp-ping bfd  
  remote VCCV        : alert ttl lsp-ping bfd  
  local control word : disable   remote control word  : disable  
  tunnel policy name : --  
  PW template name   : --  
  primary or secondary : primary  
  load balance type  : flow  
  Access-port        : false  
  Switchover Flag    : false  
  VC tunnel/token info : 1 tunnels/tokens  
    NO.0 TNL type    : lsp , TNL ID : 0x20021  
    Backup TNL type  : lsp , TNL ID : 0x0  
  create time        : 0 days, 0 hours, 8 minutes, 8 seconds  
  up time            : 0 days, 0 hours, 7 minutes, 26 seconds  
  last change time   : 0 days, 0 hours, 7 minutes, 26 seconds  
  VC last up time    : 2013/12/04 17:17:07  
  VC total up time   : 0 days, 2 hours, 12 minutes, 51 seconds  
  CKey               : 8  
  NKey               : 7  
  PW redundancy mode : frr  
  AdminPw interface  : --  
  AdminPw link state : --
```

```
Diffserv Mode      : uniform
Service Class      : --
Color              : --
DomainId           : --
Domain Name        : --
```

CE1 and CE2 can ping each other successfully.

The display on CE1 is used as an example.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=430 ms
Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=220 ms
Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=190 ms
Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=190 ms
Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=190 ms

--- 100.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 190/244/430 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 mpls l2vc 2.2.2.9 100
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 1.1.1.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of ASBR-PE1

```
#
sysname ASBR-PE1
```

```
#
mpls lsr-id 2.2.2.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 mpls l2vc 1.1.1.9 100
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of ASBR-PE2

```
#
sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 mpls l2vc 4.4.4.9 100
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls l2vpn
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
```

```

ip address 30.1.1.2 255.255.255.0
isis enable 1
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
mpls l2vc 3.3.3.9 100
#
interface LoopBack0
ip address 4.4.4.9 255.255.255.255
isis enable 1
#
return
    
```

- Configuration file of CE2

```

#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
    
```

12.10.6 Example for Configuring TDM PWE3 (Using the 8E1T1-M Interface Card)

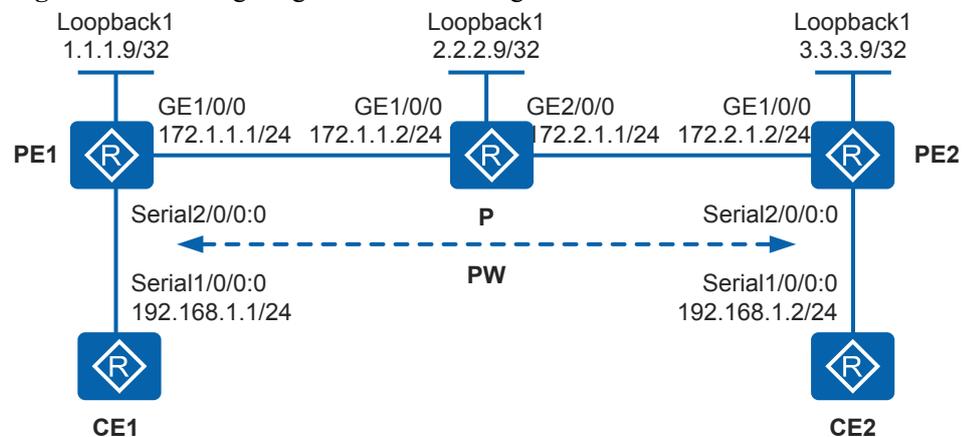
Networking Requirements

NOTE

Only the AR2220-S, AR2240-S, and AR3200-S (using SRU40) can be used in this scenario.

As shown in [Figure 12-17](#), the carrier MPLS network provides the L2VPN service for users who access the network through low-speed TDM links. The backbone devices are connected through the 4GECS interface cards on which the Combo interfaces work as electrical interfaces with a rate of 1000 Mbit/s. Many users connect to the network through PE1 and PE2, and users on the PEs change frequently. (This example lists only two user devices CE1 and CE2, and they are connected to the PEs which have 8E1T1-M interface cards installed.) A proper VPN solution is required to provide secure VPN services for users, save network resources, and simplify configuration when new users connect to the network.

Figure 12-17 Configuring TDM PWE3 using the 8E1T1-M interface card



Configuration Roadmap

Because users on the PEs change frequently, manual configuration is inefficient and may cause configuration errors. In this scenario, the two PEs can set up a remote LDP session and use the LDP protocol to synchronize user information through a dynamic PW. Compared with Martini, PWE3 reduces signaling costs and defines the multi-hop negotiation mode, making networking more flexible. PWE3 is recommended if network resources need to be saved. TDM PWE3 can be used to meet user requirements based on users' access modes.

The configuration roadmap is as follows:

1. Run an IGP protocol on the backbone network so that backbone devices can communicate.
2. Enable basic MPLS capabilities, set up an LSP tunnel on the backbone network, and establish a remote MPLS LDP peer relationship between the PEs at two ends of the PW.
3. Create an MPLS L2VC connection between CE1/PRI interfaces on the PEs to implement TDM PWE3, so that users can communicate with each other.
4. Configure all the devices to work in clock synchronization state to ensure that CEs can accurately exchange data with each other. In this example, the system clock of PE1 is used as the clock source.

Procedure

Step 1 Configure IP addresses for the interfaces on the MPLS backbone network.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol on the MPLS backbone network.

Configure an IGP protocol on the MPLS backbone network. In this example, OSPF is used.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command. You can view that the devices have learnt routes to Loopback1 of each other.

Step 3 Enable MPLS, and set up LSPs and remote LDP sessions.

Enable MPLS on the MPLS backbone network and set up a remote MPLS peer relationship between the PEs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
[PE2] mpls ldp remote-peer 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] quit
```

After the configuration is complete, run the **display mpls ldp session** command. You can view that LDP sessions are established between PEs and between PEs and P, and the session status is **Operational**.

Step 4 Configure user devices to access the PEs.

Configure interface parameters on the CEs and PEs because user devices access the PEs through low-speed TDM links.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] controller e1 1/0/0
[CE1-E1 1/0/0] using e1
[CE1-E1 1/0/0] quit
[CE1] interface serial 1/0/0:0
[CE1-Serial1/0/0:0] link-protocol ppp
[CE1-Serial1/0/0:0] ip address 192.168.1.1 255.255.255.0
[CE1-Serial1/0/0:0] quit
```

Configure PE1.

```
[PE1] controller e1 2/0/0
[PE1-E1 2/0/0] using e1
```

```
[PE1-E1 2/0/0] quit
[PE1] interface serial 2/0/0:0
[PE1-Serial2/0/0:0] link-protocol tdm
[PE1-Serial2/0/0:0] quit
```

Configure PE2.

```
[PE2] controller e1 2/0/0
[PE2-E1 2/0/0] using e1
[PE2-E1 2/0/0] quit
[PE2] interface serial 2/0/0:0
[PE2-Serial2/0/0:0] link-protocol tdm
[PE2-Serial2/0/0:0] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] controller e1 1/0/0
[CE2-E1 1/0/0] using e1
[CE2-E1 1/0/0] quit
[CE2] interface serial 1/0/0:0
[CE2-Serial1/0/0:0] link-protocol ppp
[CE2-Serial1/0/0:0] ip address 192.168.1.2 255.255.255.0
[CE2-Serial1/0/0:0] quit
```

Step 5 Create a VC connection.

Enable MPLS L2VPN on PE1 and PE2, and create a VC connection between them.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] pw-template pe2pe
[PE1-pw-template-pe2pe] peer-address 3.3.3.9
[PE1-pw-template-pe2pe] jitter-buffer depth 8
[PE1-pw-template-pe2pe] tdm-encapsulation-number 8
[PE1-pw-template-pe2pe] quit
[PE1] interface serial 2/0/0:0
[PE1-Serial2/0/0:0] mpls l2vc pw-template pe2pe 100
[PE1-Serial2/0/0:0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] pw-template pe2pe
[PE2-pw-template-pe2pe] peer-address 1.1.1.9
[PE2-pw-template-pe2pe] jitter-buffer depth 8
[PE2-pw-template-pe2pe] tdm-encapsulation-number 8
[PE2-pw-template-pe2pe] quit
[PE2] interface serial 2/0/0:0
[PE2-Serial2/0/0:0] mpls l2vc pw-template pe2pe 100
[PE2-Serial2/0/0:0] quit
```

Step 6 Configure the clock synchronization function.

Configure all the devices to work in clock synchronization state; otherwise, CEs cannot accurately exchange data with each other. The system clock of PE1 is used as the clock source for all the devices.

Configure PE1.

```
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] clock master
[PE1-GigabitEthernet1/0/0] quit
[PE1] controller e1 2/0/0
```

```
[PE1-E1 2/0/0] clock system  
[PE1-E1 2/0/0] quit
```

Configure CE1.

```
[CE1] controller e1 1/0/0  
[CE1-E1 1/0/0] clock slave  
[CE1-E1 1/0/0] quit
```

Configure the P.

```
[P] interface gigabitethernet 1/0/0  
[P-GigabitEthernet1/0/0] clock slave  
[P-GigabitEthernet1/0/0] quit  
[P] clock source 0 1/0/0  
[P] interface gigabitethernet 2/0/0  
[P-GigabitEthernet2/0/0] clock master  
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] interface gigabitethernet 1/0/0  
[PE2-GigabitEthernet1/0/0] clock slave  
[PE2-GigabitEthernet1/0/0] quit  
[PE2] clock source 0 1/0/0  
[PE2] controller e1 2/0/0  
[PE2-E1 2/0/0] clock system  
[PE2-E1 2/0/0] quit
```

Configure CE2.

```
[CE2] controller e1 1/0/0  
[CE2-E1 1/0/0] clock slave  
[CE2-E1 1/0/0] quit
```

Step 7 Verify the configuration.

Check the L2VPN connections on PEs. You can see that an L2VC connection has been set up and is in the Up state.

The display on PE1 is used as an example:

```
[PE1] display mpls l2vc interface serial 2/0/0:0  
*client interface      : Serial2/0/0:0 is up  
Administrator PW      : no  
session state         : up  
AC status              : up  
VC state               : up  
Label state           : 0  
Token state            : 0  
VC ID                  : 100  
VC type                : SAT E1 over Packet  
destination            : 3.3.3.9  
local group ID         : 0          remote group ID      : 0  
local VC label         : 1039       remote VC label      : 1045  
local TDM Encap Num    : 8          remote TDM Encap Num : 8  
jitter-buffer         : 8  
idle-code              : ff  
local rtp-header       : disable    remote rtp-header    : disable  
local bit-rate         : 32         remote bit-rate      : 32  
local AC OAM State     : up  
local PSN OAM State    : up  
local forwarding state : forwarding  
local status code      : 0x0  
remote AC OAM state    : up  
remote PSN OAM state   : up  
remote forwarding state: forwarding  
remote status code     : 0x0  
ignore standby state   : no
```

```
BFD for PW          : unavailable
VCCV State          : up
manual fault        : not set
active state        : active
forwarding entry    : exist
link state          : up
local VC MTU        : --          remote VC MTU          : --
local VCCV          : alert ttl lsp-ping bfd
remote VCCV         : alert ttl lsp-ping bfd
local control word  : disable     remote control word : disable
tunnel policy name  : --
PW template name    : pe2pe
primary or secondary : primary
load balance type   : flow
Access-port         : false
Switchover Flag     : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type     : lsp , TNL ID : 0x5
  Backup TNL type   : lsp , TNL ID : 0x0
create time         : 0 days, 0 hours, 1 minutes, 36 seconds
up time             : 0 days, 0 hours, 1 minutes, 36 seconds
last change time    : 0 days, 0 hours, 1 minutes, 36 seconds
VC last up time     : 2013/11/02 09:30:04
VC total up time    : 0 days, 0 hours, 1 minutes, 36 seconds
CKey                : 9
NKey                 : 8
PW redundancy mode  : frr
AdminPw interface   : --
AdminPw link state  : --
Diffserv Mode       : pipe
Service Class       : ef
Color                : green
DomainId             : --
Domain Name         : --
```

CE1 and CE2 can ping each other.

The display on CE1 is used as an example:

```
[CE1] ping 192.168.1.2
PING 192.168.1.2: 56 data bytes, press CTRL_C to break
  Reply from 192.168.1.2: bytes=56 Sequence=1 ttl=255 time=16 ms
  Reply from 192.168.1.2: bytes=56 Sequence=2 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=3 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=4 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=5 ttl=255 time=14 ms

--- 192.168.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 14/15/16 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
controller E1 1/0/0
  using e1
#
interface Serial1/0/0:0
  link-protocol ppp
  ip address 192.168.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
pw-template pe2pe
peer-address 3.3.3.9
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
remote-ip 3.3.3.9
#
controller E1 2/0/0
using e1
clock system
#
interface Serial2/0/0:0
link-protocol tdm
mpls l2vc pw-template pe2pe 100
#
interface GigabitEthernet1/0/0
ip address 172.1.1.1 255.255.255.0
mpls
mpls ldp
clock master
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 172.1.1.0 0.0.0.255
#
return
```

- Configuration file of the P device

```
#
sysname P
#
clock source 0 1/0/0 priority 9
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 172.1.1.2 255.255.255.0
mpls
mpls ldp
clock slave
#
interface GigabitEthernet2/0/0
ip address 172.2.1.1 255.255.255.0
mpls
mpls ldp
clock master
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
```

```
network 2.2.2.9 0.0.0.0
network 172.1.1.0 0.0.0.255
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
clock source 0 1/0/0 priority 9
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
pw-template pe2pe
peer-address 1.1.1.9
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
remote-ip 1.1.1.9
#
controller E1 2/0/0
using e1
clock system
#
interface Serial2/0/0:0
link-protocol tdm
mpls l2vc pw-template pe2pe 100
#
interface GigabitEthernet1/0/0
ip address 172.2.1.2 255.255.255.0
mpls
mpls ldp
clock slave
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
controller E1 1/0/0
using e1
#
interface Serial1/0/0:0
link-protocol ppp
ip address 192.168.1.2 255.255.255.0
#
return
```

12.10.7 Example for Configuring TDM PWE3 (Using the 8SA interface card)

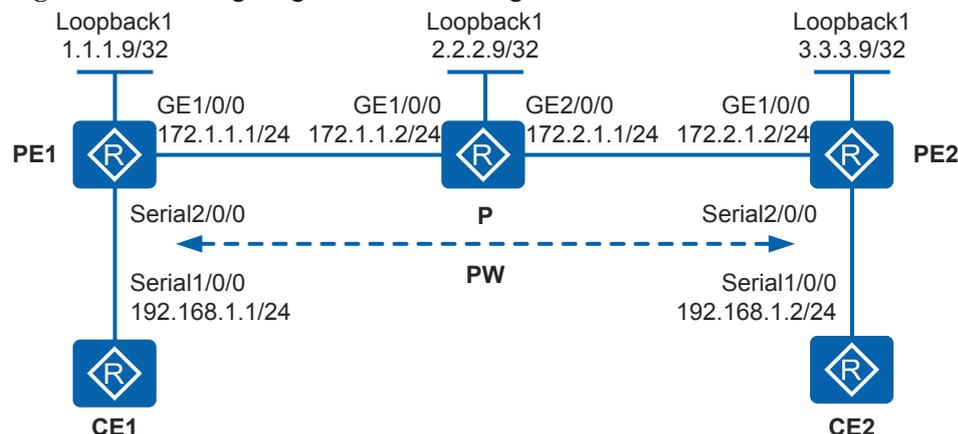
Networking Requirements

NOTE

Only the AR2220-S, AR2240-S, and AR3200 (using SRU40) can be used in this scenario.

As shown in [Figure 12-18](#), the MPLS network of an Internet service provider (ISP) provides the L2VPN service for users who access the network through low-speed TDM links. Many users connect to the network through PE1 and PE2, and users on the PEs change frequently. (This example lists only two user devices CE1 and CE2, and they are connected to the PEs which have 8SA interface cards installed.) A proper VPN solution is required to provide secure VPN services for users, save network resources, and simplify configuration when new users connect to the network.

Figure 12-18 Configuring TDM PWE3 using the 8SA interface card



Configuration Roadmap

Because users on the PEs change frequently, manual configuration is inefficient and may cause configuration errors. In this scenario, the two PEs can set up a remote LDP session and use the LDP protocol to synchronize user information through a dynamic PW. Compared with Martini, PWE3 reduces signaling costs and defines the multi-hop negotiation mode, making networking more flexible. PWE3 is recommended if network resources need to be saved. TDM PWE3 can be used to meet user requirements based on users' access modes.

The configuration roadmap is as follows:

1. Run an IGP protocol on the backbone network so that backbone devices can communicate.
2. Enable basic MPLS capabilities, set up an LSP tunnel on the backbone network, and establish a remote MPLS LDP peer relationship between the PEs at two ends of the PW.
3. Create an MPLS L2VC connection between CE1/PRI interfaces on the PEs to implement TDM PWE3, so that users can communicate with each other.

Procedure

Step 1 Configure IP addresses for the interfaces on the MPLS backbone network.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 172.1.1.1 255.255.255.0
[PE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure an IGP protocol on the MPLS backbone network.

Configure an IGP protocol on the MPLS backbone network. In this example, OSPF is used.

Configure PE1. The configuration on P and PE2 is similar to the configuration on PE1 and is not mentioned here.

```
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command. You can view that the devices have learnt routes to Loopback1 of each other.

Step 3 Enable MPLS, and set up LSPs and remote LDP sessions.

Enable MPLS on the MPLS backbone network and set up a remote MPLS peer relationship between the PEs.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] mpls
[PE1-GigabitEthernet1/0/0] mpls ldp
[PE1-GigabitEthernet1/0/0] quit
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
[PE2] mpls ldp remote-peer 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] quit
```

After the configuration is complete, run the **display mpls ldp session** command. You can view that LDP sessions are established between PEs and between PEs and P, and the session status is **Operational**.

Step 4 Configure user devices to access the PEs.

Configure interface parameters on the CEs and PEs because user devices access the PEs through low-speed TDM links.

Configure CE1.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface serial 1/0/0
[CE1-Serial1/0/0] link-protocol ppp
[CE1-Serial1/0/0] ip address 192.168.1.1 255.255.255.0
[CE1-Serial1/0/0] physical-mode async
[CE1-Serial1/0/0] quit
```

Configure PE1.

```
[PE1] interface serial 2/0/0
[PE1-Serial2/0/0] link-protocol tdm
[PE1-Serial2/0/0] physical-mode async
[PE1-Serial2/0/0] quit
```

Configure PE2.

```
[PE2] interface serial 2/0/0
[PE2-Serial2/0/0] link-protocol tdm
[PE2-Serial2/0/0] physical-mode async
[PE2-Serial2/0/0] quit
```

Configure CE2.

```
<Huawei> system-view
[Huawei] sysname CE2
[CE2] interface serial 1/0/0
[CE2-Serial1/0/0] link-protocol ppp
[CE2-Serial1/0/0] ip address 192.168.1.2 255.255.255.0
[CE2-Serial1/0/0] physical-mode async
[CE2-Serial1/0/0] quit
```

Step 5 Create a VC connection.

Enable MPLS L2VPN on PE1 and PE2, and create a VC connection between them.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] pw-template pe2pe
[PE1-pw-template-pe2pe] peer-address 3.3.3.9
```

```
[PE1-pw-template-pe2pe] jitter-buffer depth 8
[PE1-pw-template-pe2pe] tdm-encapsulation-number 8
[PE1-pw-template-pe2pe] quit
[PE1] interface serial 2/0/0
[PE1-Serial2/0/0] mpls l2vc pw-template pe2pe 100
[PE1-Serial2/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] pw-template pe2pe
[PE2-pw-template-pe2pe] peer-address 1.1.1.9
[PE2-pw-template-pe2pe] jitter-buffer depth 8
[PE2-pw-template-pe2pe] tdm-encapsulation-number 8
[PE2-pw-template-pe2pe] quit
[PE2] interface serial 2/0/0
[PE2-Serial2/0/0] mpls l2vc pw-template pe2pe 100
[PE2-Serial2/0/0] quit
```

Step 6 Verify the configuration.

Check the L2VPN connections on PEs. You can see that an L2VC connection has been set up and is in the Up state.

The display on PE1 is used as an example:

```
[PE1] display mpls l2vc interface serial 2/0/0
*client interface      : Serial2/0/0 is up
 Administrator PW     : no
 session state        : up
 AC status            : up
 VC state             : up
 Label state          : 0
 Token state          : 0
 VC ID                : 100
 VC type              : CESoPSN basic mode
 destination          : 3.3.3.9
 local group ID       : 0           remote group ID      : 0
 local VC label       : 1039        remote VC label      : 1045
 local TDM Encap Num  : 8           remote TDM Encap Num : 0
 jitter-buffer        : 8
 idle-code            : ff
 local rtp-header     : disable     remote rtp-header    : disable
 local bit-rate       : 0           remote bit-rate      : 0
 local AC OAM State   : up
 local PSN OAM State  : up
 local forwarding state : forwarding
 local status code    : 0x0
 remote AC OAM state  : up
 remote PSN OAM state : up
 remote forwarding state: forwarding
 remote status code   : 0x0
 ignore standby state : no
 BFD for PW           : unavailable
 VCCV State           : up
 manual fault         : not set
 active state         : active
 forwarding entry     : exist
 link state           : up
 local VC MTU         : --           remote VC MTU        : --
 local VCCV           : alert ttl lsp-ping bfd
 remote VCCV          : alert ttl lsp-ping bfd
 local control word   : disable     remote control word  : disable
 tunnel policy name   : --
 PW template name     : pe2pe
 primary or secondary : primary
 load balance type    : flow
 Access-port          : false
```

```
Switchover Flag      : false
VC tunnel/token info : 1 tunnels/tokens
  NO.0 TNL type      : lsp , TNL ID : 0x5
  Backup TNL type    : lsp , TNL ID : 0x0
create time          : 0 days, 0 hours, 1 minutes, 36 seconds
up time              : 0 days, 0 hours, 1 minutes, 36 seconds
last change time     : 0 days, 0 hours, 1 minutes, 36 seconds
VC last up time      : 2013/11/02 09:30:04
VC total up time     : 0 days, 0 hours, 1 minutes, 36 seconds
CKey                  : 9
NKey                  : 8
PW redundancy mode   : frr
AdminPw interface    : --
AdminPw link state   : --
Diffserv Mode        : pipe
Service Class        : ef
Color                 : green
DomainId              : --
Domain Name           : --
```

CE1 and CE2 can ping each other.

The display on CE1 is used as an example:

```
[CE1] ping 192.168.1.2
PING 192.168.1.2: 56 data bytes, press CTRL_C to break
  Reply from 192.168.1.2: bytes=56 Sequence=1 ttl=255 time=16 ms
  Reply from 192.168.1.2: bytes=56 Sequence=2 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=3 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=4 ttl=255 time=15 ms
  Reply from 192.168.1.2: bytes=56 Sequence=5 ttl=255 time=14 ms

--- 192.168.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 14/15/16 ms
```

---End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface Serial1/0/0
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
 physical-mode async
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
pw-template pe2pe
 peer-address 3.3.3.9
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
```

```
remote-ip 3.3.3.9
#
interface Serial2/0/0
 link-protocol tdm
 mpls l2vc pw-template pe2pe 100
 physical-mode async
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 1.1.1.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
 network 1.1.1.9 0.0.0.0
 network 172.1.1.0 0.0.0.255
#
return
```

- Configuration file of the P device

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
 ip address 172.1.1.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 ip address 172.2.1.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack1
 ip address 2.2.2.9 255.255.255.255
#
ospf 1
 area 0.0.0.0
 network 2.2.2.9 0.0.0.0
 network 172.1.1.0 0.0.0.255
 network 172.2.1.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
pw-template pe2pe
 peer-address 1.1.1.9
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
 remote-ip 1.1.1.9
#
interface Serial2/0/0
```

```

link-protocol tdm
mpls l2vc pw-template pe2pe 100
physical-mode async
#
interface GigabitEthernet1/0/0
ip address 172.2.1.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 172.2.1.0 0.0.0.255
#
return
    
```

- Configuration file of CE2

```

#
sysname CE2
#
interface Serial1/0/0
link-protocol ppp
ip address 192.168.1.2 255.255.255.0
physical-mode async
#
return
    
```

12.11 References for PWE3

This section provides references for PWE3.

The following table lists the references for PWE3.

Document	Description	Remarks
RFC3916	Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)	-
RFC3985	Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture	-
RFC4446	IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)	-
draft-ietf-pwe3-control-protocol-17	Pseudo wire Setup and Maintenance using the Label Distribution Protocol	-
draft-martini-pwe3-pw-switching-03	Pseudo Wire Switching	-
draft-ietf-pwe3-cw-00	PWE3 Control Word for use over an MPLS PSN	-
draft-ietf-pwe3-vcv-03	Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)	-
draft-ietf-pwe3-ethernet-encap-10	Encapsulation Methods for Transport of Ethernet Over MPLS Networks	-

Document	Description	Remarks
draft-ietf-pwe3-atm-encap-11	Encapsulation Methods for Transport of ATM Over MPLS Networks	-
draft-ietf-pwe3-cell-transport-05	PWE3 ATM Transparent Cell Transport Service	-
RFC 5085	Pseudowire Virtual Circuit Connectivity Verification (VCCV) A Control Channel for Pseudowires	The device does not support PW VCCV in L2TP V3 mode.

13 VPLS Configuration

About This Chapter

This chapter describes principles, applications, and configurations of the Virtual Private LAN Service (VPLS).

[13.1 Overview of VPLS](#)

This section describes the definition, background, and functions of VPLS.

[13.2 Understanding VPLS](#)

This section describes the implementation of VPLS.

[13.3 Application Scenarios for VPLS](#)

This section describes the application scenarios for VPLS.

[13.4 Licensing Requirements and Limitations for VPLS](#)

[13.5 Default Settings for VPLS](#)

This section provides the default settings for VPLS.

[13.6 Configuring Martini VPLS](#)

This section describes how to configure VPLS in Martini mode using LDP as the signalling protocol.

[13.7 \(Optional\) Configuring Inter-AS Martini VPLS](#)

Inter-AS VPLS allows a VPLS network to span multiple ASs on an MPLS backbone network.

[13.8 \(Optional\) Setting Related Parameters for a VSI](#)

This section describes how to set related parameters for a VSI.

[13.9 Maintaining VPLS](#)

VPLS maintenance includes collecting, clearing, and viewing traffic statistics on VPLS PWs, enabling or disabling VSIs, clearing MAC address entries, and detecting the VPLS network connectivity.

[13.10 Configuration Examples for VPLS](#)

This section provides several configuration examples of different VPLS networking, including networking requirements, configuration notes, configuration roadmap, configuration procedures, and configuration files.

[13.11 Troubleshooting VPLS](#)

This section describes common faults caused by incorrect VPLS configurations and provides the troubleshooting procedure.

13.12 References for VPLS

This section lists references for VPLS.

13.1 Overview of VPLS

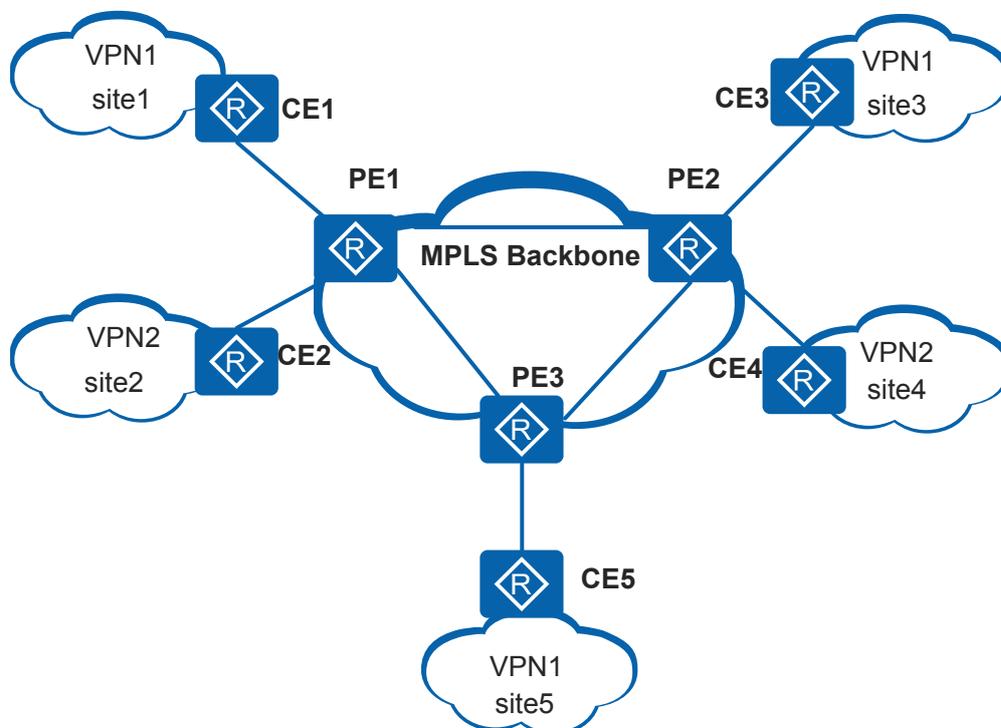
This section describes the definition, background, and functions of VPLS.

Definition

As an Multiprotocol Label Switching (MPLS)-based point-to-multipoint (P2MP) Layer 2 Virtual Private Network (L2VPN) service provided over a public network, the virtual private LAN service (VPLS) ensures that geographically isolated user sites can communicate over metropolitan area networks (MANs) and wide area networks (WANs) as if they were on the same local area network (LAN). VPLS is also called the Transparent LAN Service (TLS).

Figure 13-1 shows a typical VPLS networking mode. In this networking, users located in different geographical regions communicate with each other over different provider edges (PEs). From the perspective of users, a VPLS network is a Layer 2 switched network that allows them to communicate with each other in a way similar to communication over a LAN.

Figure 13-1 Typical VPLS networking



Purpose

As enterprises set up more and more branches in different regions and office flexibility increases, applications such as VoIP, instant messaging, and teleconferencing are increasingly widely used. This imposes high requirements for end-to-end (E2E) datacom technologies. A network capable of providing P2MP services is the key to datacom function implementation.

Traditional asynchronous transfer mode (ATM) and frame relay (FR) technologies provide only Layer 2 point-to-point (P2P) connections. In addition, those network types have disadvantages such as high construction costs, low speed, and complex deployment. The development of Internet Protocol (IP) has led to MPLS VPN technology, which can provide VPN services over an IP network and offers advantages such as easy configuration and flexible bandwidth control. MPLS VPNs can be classified into MPLS L2VPNs and MPLS L3VPNs.

- Traditional MPLS L2VPNs, such as the virtual leased lines (VLLs), can provide P2P services but not P2MP services over a public network.
- MPLS L3VPNs can provide P2MP services on the precondition that PEs keep routes destined for end users. This implementation requires high routing performance of PEs.

To solve the preceding problems, VPLS, an MPLS-based Ethernet technology, is introduced as an extension to a traditional MPLS L2VPN solution.

- Like Ethernet, VPLS supports P2MP communication.
- VPLS is a Layer 2 label switching technology. From the perspective of users, the entire MPLS IP backbone network is a Layer 2 switching device. PEs do not need to keep routes destined for end users.

VPLS provides a more complete multipoint communication solution, integrating the advantages provided by Ethernet and MPLS. By emulating traditional LAN functions, VPLS enables users on different LANs to communicate with each other over IP/MPLS networks as if they were on the same LAN.

Benefits

- VPLS networks can be constructed based on carrier's IP networks, reducing construction costs.
- VPLS networks inherit the high-speed advantage of the Ethernet.
- VPLS networks allow users to communicate over Ethernet links, regardless of whether these links are on WANs or LANs. This feature allows services to be rapidly and flexibly deployed.
- VPLS allows enterprises to control and maintain routing policies on their networks, improving the VPN network security and maintainability.

13.2 Understanding VPLS

This section describes the implementation of VPLS.

13.2.1 Implementation

Basic VPLS Transport Structure

Figure 13-2 shows an example of a VPLS network. The entire VPLS network is similar to a switch. PWs are established over MPLS tunnels between VPN sites to transparently transmit

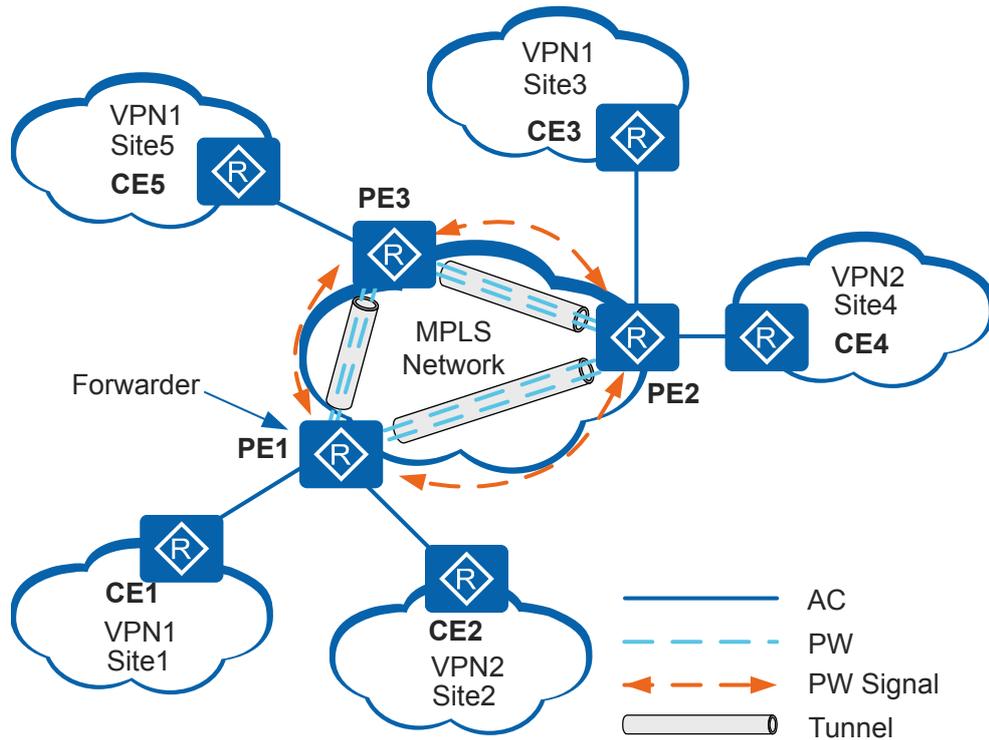
Layer 2 packets between sites. When forwarding packets, Provider edges (PEs) learn the source MAC addresses of these packets and create MAC entries, mapping MAC addresses to attachment circuits (ACs) and pseudo wires (PWs).

The following table describes the various concepts related to VPLS networks.

Table 13-1 Description of VPLS concepts

Name	Description
AC	<p>A link between a CE and a PE. An AC must be established using Ethernet interfaces.</p> <p>On a VPLS network, AC interfaces can be Ethernet interfaces, Ethernet sub-interfaces.</p>
PW	<p>A bidirectional virtual connection between two virtual switch instances (VSIs) residing on two PEs. A PW consists of a pair of unidirectional MPLS VCs transmitting in opposite directions.</p> <p>A PW is a direct tunnel connecting the local AC and the remote AC, and transparently transmits Layer 2 data.</p>
VSI	<p>A virtual switching unit on the switch for each VPLS. Each VSI has an independent MAC address table and a forwarder. A VSI is responsible for terminating PWs.</p>
PW signal	<p>A type of signaling used to create and maintain PWs. PW signaling is the foundation for VPLS implementation. Currently, the PW signaling is Label Distribution Protocol (LDP).</p>
Tunnel	<p>A connection between a local PE and a remote PE used to transparently transmit data between PEs. A tunnel can carry multiple PWs and the tunnel type can be Label Switched Path (LSP).</p>
Forwarder	<p>Similar to a VPLS forwarding table. After a PE receives packets from an AC, the forwarder of the PE selects a PW to forward these packets.</p>

Figure 13-2 Basic VPLS transmission process



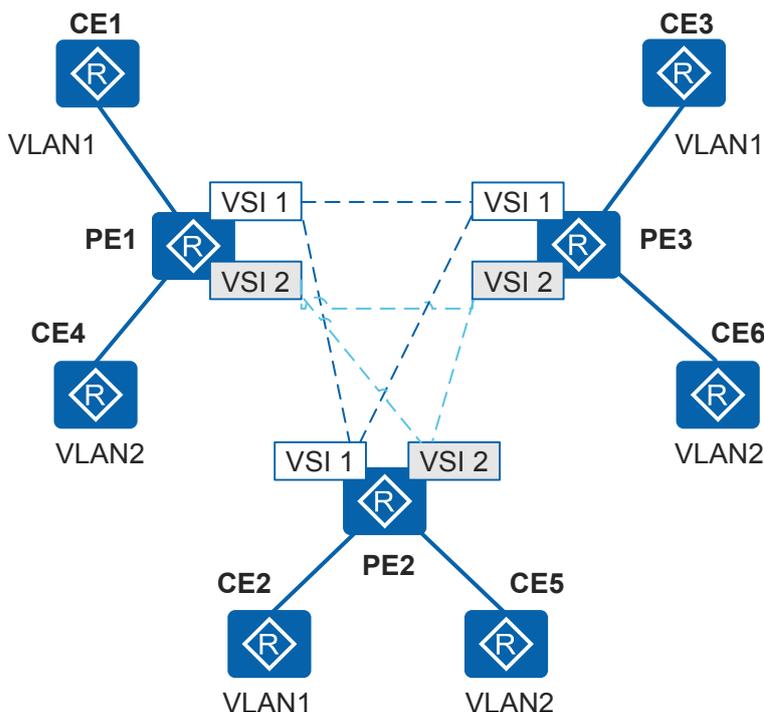
The following uses the unicast packets sent from CE1 to CE3 in VPN1 as an example to describe how a data flow is transmitted on a VPLS network.

1. PE1, PE2, and PE3 belong to the same VPLS domain. AC links connected to the VPLS domain are mapped to PWs through a VSI to generate the forwarder of the VSI.
2. When CE1 receives a Layer 2 packet from a user at Site1, it forwards the Layer 2 packet to PE1 through the AC link.
3. After receiving the packet, PE1 finds that the packet needs to be forwarded in VPLS mode. PE1 then selects a PW from the forwarder to forward the packet based on the destination MAC address of the packet.
4. PE1 generates double labels according to the forwarding entry of the PW. The inner private network label identifies the PW, and the outer public network label enables the packet to reach PE2 through the tunnel on the public network. Meanwhile, PE1 searches for the destination MAC address based on the MAC address table index and encapsulates the packet.
5. After the Layer 2 packet arrives at PE2 through the tunnel on the public network, the private network label becomes the outer label (the public network label has been popped out at the penultimate hop).
6. Upon receiving the packet, PE2 selects a VSI for forwarding the packet according to the private network label, removes the private network label, and selects the forwarder of the VSI. The forwarder forwards the Layer 2 packet to CE3 according to the destination MAC address.

VPLS Implementation Process

In **Figure 13-2**, transmission of packets between Customer Edges (CEs) relies on VSIs configured on PEs, and PWs established between the VSIs. **Figure 13-3** shows transmission of Ethernet frames over full-mesh PWs between PEs.

Figure 13-3 VPLS forwarding model



A VPLS network consists of a control plane and a forwarding plane.

- The control plane of a VPLS PE provides the PW establishment function, including:
 - Member discovery: a process in which a PE in a VSI discovers the other PEs in the same VSI. This process can be implemented manually.
 - Signaling mechanism: PWs between PEs in the same VSI are established, maintained, or torn down using signaling protocols.
- The forwarding plane of a VPLS PE provides the data forwarding function, including:
 - Encapsulation: After receiving Ethernet frames from a CE, a PE encapsulates the frames into packets and sends the packets to VPLS network.
 - Forwarding: A PE determines how to forward a packet based on the inbound interface and destination MAC address of the packet.
 - Decapsulation: After receiving packets from VPLS network, a PE decapsulates these packets into Ethernet frames and sends the frames to a CE.

13.2.2 PW Signaling Protocols

PW Signaling Protocols and VPLS Implementation Modes

A PW mainly uses LDP as its signaling protocol and LDP is the basis for implementation of **LDP VPLS**.

LDP VPLS

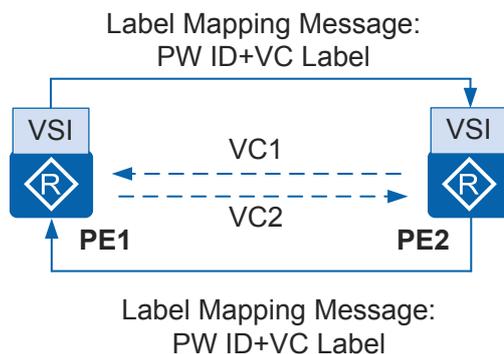
Introduction to LDP VPLS

LDP VPLS (Martini VPLS) uses a static discovery mechanism to discover VPLS members using LDP signaling. VPLS information is carried in extended TLV fields (type 128 and type 129 FEC TLVs) of LDP signaling packets. During the establishment of a PW, the label distribution mode is downstream unsolicited (DU) and the label retention mode is liberal.

Implementation process

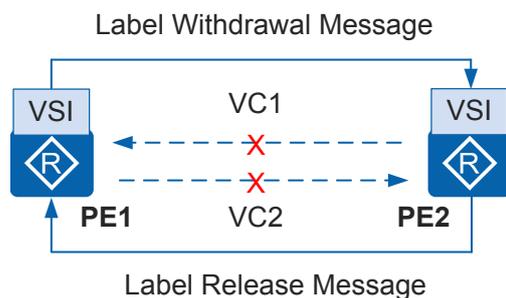
- **Figure 13-4** shows the process of establishing a PW using LDP signaling.

Figure 13-4 Establishing a PW using LDP signaling



- After PE1 is associated with a VSI, and PE2 is configured as a peer of PE1, PE1 sends a Label Mapping message to PE2 in DU mode if an LDP session already exists between PE1 and PE2. The Label Mapping message carries information required to establish a PW, such as the PW ID, VC label, and interface parameters.
 - Upon receipt of the message, PE2 checks whether itself has been associated with the VSI. If PE2 has been associated with the VSI and PW parameters on PE1 and PE2 are consistent, PE1 and PE2 belong to the same VSI. In this case, PE2 establishes a unidirectional VC named VC1 immediately after PE2 receives the Label Mapping message. Meanwhile, PE2 sends a Label Mapping message to PE1. After receiving the message, PE1 takes a similar sequence of actions to PE2 and establishes VC2.
- **Figure 13-5** shows the process of tearing down a PW using LDP signaling.

Figure 13-5 Tearing down a PW using LDP signaling



- After the peer configuration about PE2 is deleted from PE1, PE1 sends a Label Withdrawal message to PE2. After receiving the Label Withdrawal message, PE2 withdraws its local VC label, tears down VC1, and sends a Label Release message to PE1.
- After receiving the Label Release message, PE1 withdraws its local VC label and tears down VC2.

13.2.3 Packet Encapsulation

Packet Encapsulation on ACs

Packet encapsulation on ACs depends on the user access mode, which can be VLAN or Ethernet access. The default user access mode is VLAN access.

Table 13-2 Packet encapsulation on ACs

Packet Encapsulation Type	Description
VLAN	The header of each Ethernet frame sent between CEs and PEs carries a VLAN tag, known as the provider-tag (P-Tag). This is a service delimiter identifying users on an ISP network.
Ethernet	The header of each Ethernet frame sent between CEs and PEs does not carry a P-Tag. If the frame header contains a VLAN tag, it is an inner VLAN tag called the user-tag (U-Tag). A CE does not add the U-Tag to an Ethernet frame; instead, the tag is carried in a packet before the packet is sent to the CE. A U-Tag informs the CE to which VLAN the packet belongs, and is meaningless to PEs.

Packet Encapsulation on PWs

The PW ID and PW encapsulation type uniquely identify a PW. The PW IDs and PW encapsulation types configured on the two end PEs of a PW must be the same. The packet encapsulation types of packets on PWs can be raw or tagged. By default, packets on PWs are encapsulated in tagged mode.

Table 13-3 Packet encapsulation on PWs

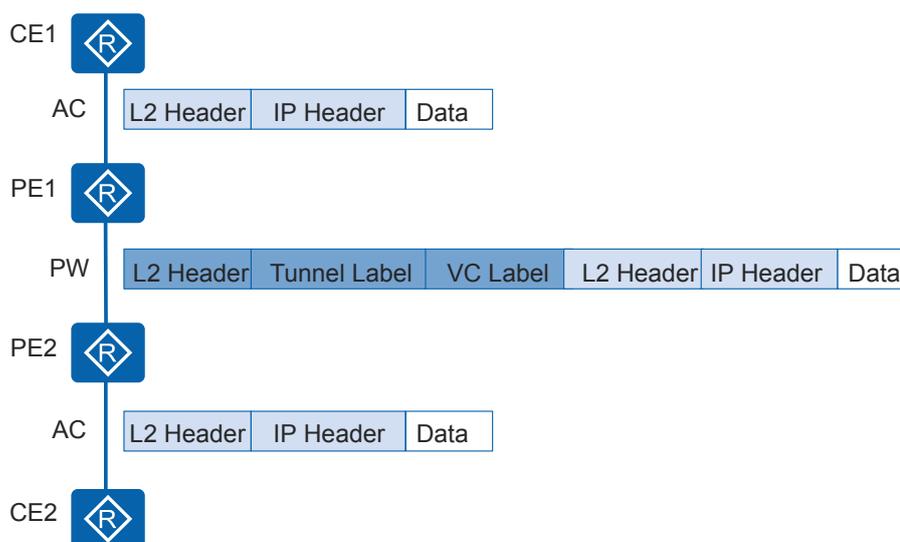
Packet Encapsulation Type	Description
Raw	Packets transmitted over a PW cannot carry P-Tags. If a PE receives a packet with the P-Tag from a CE, the PE strips the P-Tag and adds double labels (outer tunnel label and inner VC label) to the packet before forwarding it. If a PE receives a packet with no P-Tag from a CE, the PE directly adds double labels (outer tunnel label and inner VC label) to the packet before forwarding it. The PE determines whether to add the P-Tag to a packet based on actual configurations before sending it to a CE. The PE is not allowed to rewrite or remove an existing U-Tag.
Tagged	Packets transmitted over a PW must carry P-Tags. If a PE receives a packet with the P-Tag from a CE, the PE directly adds double labels (outer tunnel label and inner VC label) to the packet before forwarding it. If a PE receives a packet with no P-Tag from a CE, the PE adds a null P-Tag and double labels (outer tunnel label and inner VC label) to the packet before forwarding it. The PE determines whether to rewrite, remove, or preserve the P-Tag of a packet based on actual configurations before forwarding it to a CE.

VPLS Packets and Encapsulation Modes

Encapsulation modes of packets transmitted over ACs and PWs can be used together. The following uses Ethernet+raw encapsulation (without the U-Tag) and VLAN+tagged encapsulation (with the U-Tag) as examples to describe the packet exchange process.

- Ethernet+raw encapsulation (without the U-Tag)

Figure 13-6 Ethernet+raw encapsulation (without the U-Tag)



As shown in **Figure 13-6**, ACs use Ethernet encapsulation and PWs use raw encapsulation; packets transmitted from CEs to PEs do not carry U-Tags.

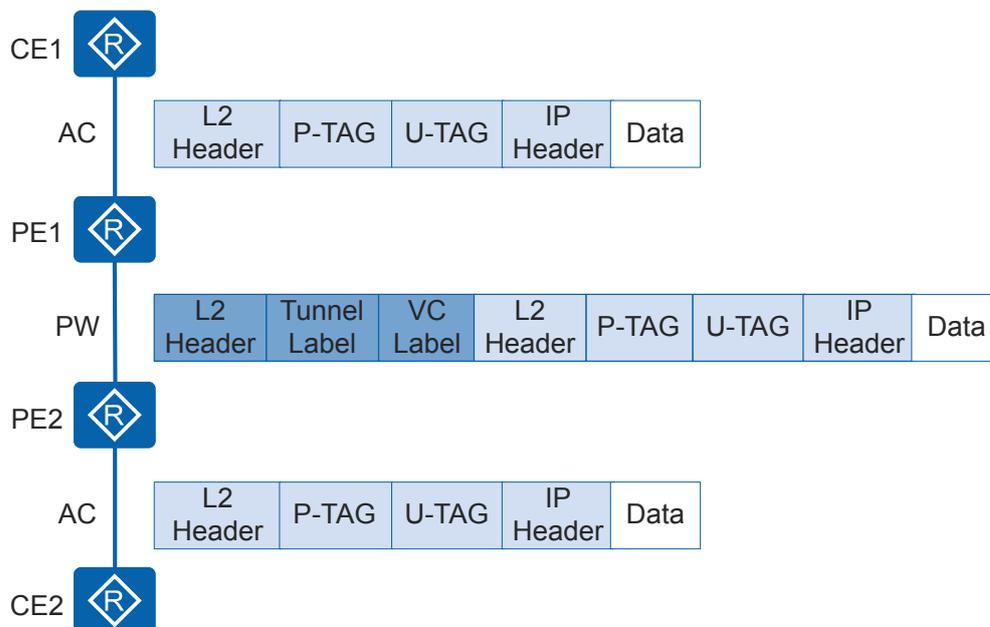
The packet exchange process is as follows:

- a. CE1 sends a Layer 2 packet without a U-Tag or P-Tag to PE1.
- b. PE1 searches the corresponding VSI for a forwarding entry and selects a tunnel and a PW to forward the packet based on the found forwarding entry.
- c. PE2 receives the packet from PE1 and decapsulates the packet by removing the Layer 2 encapsulation header added by PE1 and the inner VC label of the packet (the outer tunnel label has been popped out at the penultimate hop).
- d. PE2 sends the original Layer 2 packet to CE2.

The processing of sending a packet from CE2 to CE1 is similar to this process.

- VLAN+tagged encapsulation (with the U-Tag)

Figure 13-7 VLAN access in tagged mode (with the U-Tag)



As shown in **Figure 13-7**, ACs use VLAN encapsulation and PWs use tagged encapsulation; packets transmitted from CEs to PEs carry U-Tags and P-Tags.

The packet exchange process is as follows:

- a. CE1 sends a packet that has Layer 2 encapsulation and carries both a U-Tag and a P-Tag to PE1.
- b. Upon receipt, PE1 does not process the two tags. PE1 retains the U-Tag because it treats the U-tag service data.
- c. PE1 retains the P-Tag because a packet sent to a PW with the tagged packet encapsulation mode must carry a P-Tag.
- d. PE1 searches the corresponding VSI for a forwarding entry and selects a tunnel and a PW to forward the packet based on the found forwarding entry.

- e. PE1 adds double labels (outer tunnel label and inner VC label) to the packet based on the selected tunnel and PW, performs Layer 2 encapsulation, and forwards the packet to PE2.
- f. PE2 receives the packet from PE1 and decapsulates the packet by removing the Layer 2 encapsulation header added by PE1 and the inner VC label of the packet (the outer tunnel label has been popped out at the penultimate hop).
- g. PE2 forwards the original Layer 2 packet to CE2. The packet carries the U-Tag and P-Tag.

The processing of sending a packet from CE2 to CE1 is similar to this process.

Processing of Tags Carried in Packets

A PE device processes tags carried in packets based on the AC interface type.

Tag processing is not affected by the PW encapsulation type.

Table 13-4 PE's Tag Processing for Packets from an AC to a PW

AC Interface Type	Packet Processing on the AC Interface	Packet Processing on the PW Interface
Main interface (Ethernet or GE interface)	Does not process the packet.	<ul style="list-style-type: none"> ● Does not process the packet if the default VLAN of the AC interface is not VLAN 0. ● Adds VLAN 0 to the packet if the default VLAN of the AC interface is VLAN 0.
Dot1q termination sub-interface	Removes the outer VLAN tag.	
QinQ termination sub-interface	Removes the inner and outer VLAN tags.	

Table 13-5 PE's Tag Processing for Packets from a PW to an AC

AC Interface Type	Packet Processing on the PW Interface	Packet Processing on the AC Interface
Main interface (Ethernet or GE interface)	<ul style="list-style-type: none"> ● Does not process the packet if the default VLAN of the AC interface is not VLAN 0. ● Removes VLAN 0 from the packet if the default VLAN of the AC interface is VLAN 0. 	Does not process the packet.
Dot1q termination sub-interface		Adds an outer VLAN tag.
QinQ termination sub-interface		Adds inner and outer VLAN tags.

13.2.4 MAC Address Management

Background

A characteristic of the Ethernet is that a interface sends unicast packets with unknown destination MAC addresses, broadcast packets, and multicast packets to all other interfaces on the Ethernet. As an Ethernet-based technology, VPLS emulates an Ethernet bridge for user networks. To forward packets on a VPLS network, PEs must establish MAC address tables and forward packets based on MAC addresses or MAC addresses and VLAN tags.

MAC Address Learning and Flooding

MAC address learning

PEs create MAC address tables based on dynamic MAC address learning and associates destination MAC addresses with PWs.

Table 13-6 describes MAC address learning modes.

Table 13-6 MAC address learning modes

MAC Address Learning Mode	Description	Characteristic
Qualified	A PE learns the MAC addresses and VLAN tags of received Ethernet frames. In this mode, each user VLAN is an independent broadcast domain and has independent MAC address space.	The broadcast domain is confined to each user VLAN. Qualified learning can result in large Forwarding Information Base (FIB) table sizes, because the logical MAC address is now a VLAN tag + MAC address.
Unqualified	A PE learns only the MAC addresses of Ethernet frames. In this mode, all user VLANs share the same broadcast domain and MAC address space. The MAC address of each user VLAN must be unique.	If an AC interface is associated with multiple user VLANs, this AC interface must be a physical interface bound to a unique VSI.

NOTE

At present, the device supports only MAC address learning in unqualified mode.

Flooding

Packets with unknown addresses are broadcast in Ethernet. Therefore, in VPLS, the received packets with unknown unicast addresses, broadcast addresses, or multicast addresses are flooded to all the other interfaces. If these packets need to be forwarded in multicast mode, PEs use other methods such as Internet Group Management Protocol (IGMP) snooping.

Implementation

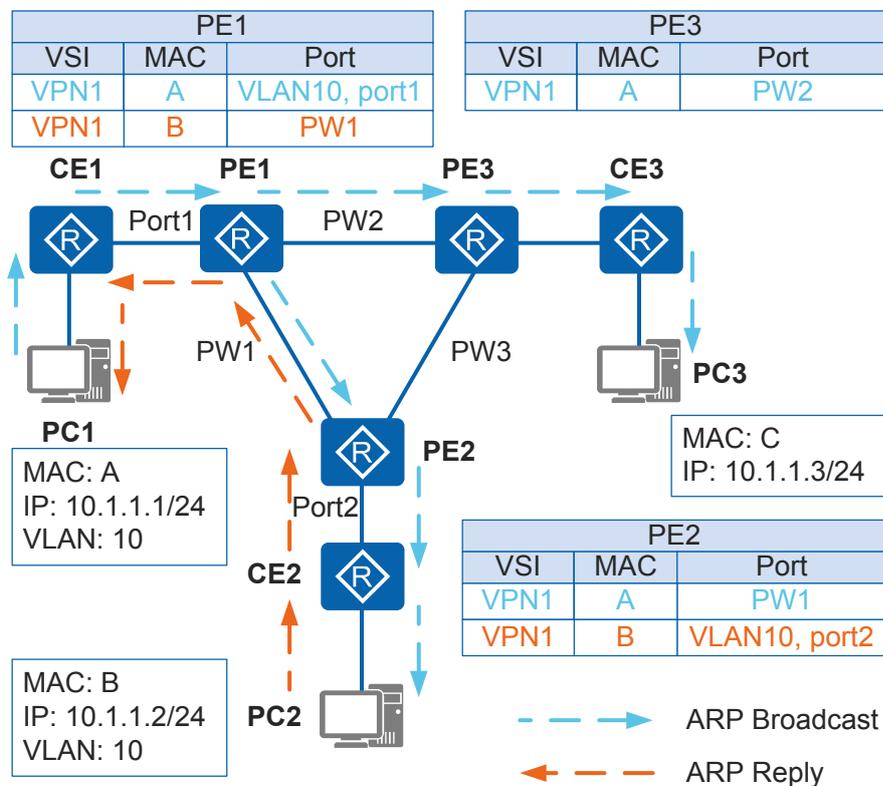
Table 13-7 describes the MAC address learning process.

Table 13-7 MAC address learning process

MAC Address Learning Process	Description
Learning MAC addresses from user-side packets	After receiving packets from a CE, a PE maps their source MAC addresses to AC interfaces. Figure 13-8 shows a mapping example with Port1.
Learning MAC addresses from PW-side packets	A PW consists of a pair of MPLS Virtual Circuits (VCs) transmitting in opposite directions. A PW will go Up only after the two MPLS VCs are established. After a PE receives a packet with an unknown source MAC address from a PW, the PE maps the source MAC address to the PW receiving the packet.

Figure 13-8 shows the process of MAC address learning and flooding on a PE. PC1 and PC2 both belong to VLAN10. When PC1 pings IP address 10.1.1.2, PC1 does not know the MAC address corresponding to this IP address and advertises an Address Resolution Protocol (ARP) Request packet.

Figure 13-8 MAC address learning and flooding process



1. After receiving the ARP Request packet sent by PC1 from Port1 (VLAN10) that connects to CE1, PE1 adds the MAC address of PC1 to its own MAC address table, as shown in the blue section of the MAC entry.
2. PE1 floods the ARP Request packet (the blue dotted line on PE1) to other interfaces (PW1 and PW2 are regarded as interfaces at this time).
3. After receiving the ARP Request packet from PW1, PE2 adds the MAC address of PC1 to its own MAC address table, as shown in the blue section of the MAC entry.
4. Based on split horizon, PE2 sends the ARP Request packet to only the interface connecting to CE2 (as indicated by the blue dashed line), but not to PW1. This ensures that only PC2 receives the ARP Request packet. VPLS split horizon ensures that packets received from public network PWs are forwarded to only private networks, not to other public network PWs.
5. After PC2 receives the ARP Request packet and finds that it is the destination of this packet, PC2 sends an ARP Reply packet to PC1 (as indicated by the orange dashed line).
6. After receiving the ARP Reply packet from PC2, PE2 adds the MAC address of PC2 to its own MAC address table, as indicated by the orange section of the MAC entry. The destination MAC address of the ARP Reply packet is the MAC address of PC1 (MAC A). After searching its MAC address table, PE2 sends the ARP Reply packet to PE1 over PW1.
7. After receiving the ARP Reply packet from PE2, PE1 adds the MAC address of PC2 to its own MAC address table, as shown in the orange section of the MAC entry. After searching its MAC address table, PE1 sends the ARP Reply packet to PC1 through Port1.
8. After receiving the ARP Reply packet from PC2, PC1 completes MAC address learning.
9. While advertising the ARP Request packet to PW1, PE1 also advertises the ARP Request packet to PE3 over PW2. After receiving the ARP Request packet, PE3 adds the MAC address of PC1 to its MAC address table, as shown in the blue section of the MAC entry. Based on split horizon, PE3 sends the ARP Request packet to only PC3. Because PC3 is not the destination of the ARP Request packet, PC3 does not send any ARP Reply packet.

MAC Address Withdrawal

Dynamic MAC addresses need to be updated and relearned. The VPLS draft defines a MAC Withdraw message with an optional MAC type-length-value (TLV) to remove or relearn the MAC address list.

When the topology changes, MAC Withdraw messages enable devices to delete matching MAC addresses quickly. MAC Withdraw messages are classified into two types:

- Messages with a MAC address list
- Messages without a MAC address list

When a backup link (AC link or VC link) becomes Up, a PE that detects the link status change receives a MAC Withdraw message carrying a list of MAC addresses to be relearned. After receiving the message, the PE updates the MAC address entries in the forward information base (FIB) table of the corresponding VSI, and sends the message to PEs directly connected to it through Label Distribution Protocol (LDP) sessions. If the MAC Withdraw message contains an empty MAC address list TLV, the PE deletes all the MAC addresses in the specified VSI except the MAC address learned from the PE that sends the message.

MAC Address Aging

An aging mechanism removes MAC entries that a PE no longer needs. If a MAC entry is not updated within a specified period of time, this entry will be aged.

13.2.5 Loop Prevention

On an Ethernet network, the Spanning Tree Protocol (STP) is often enabled to prevent loops. However, VPLS users are not aware of the Internet Service Provider (ISP) network. Therefore, STP enabled on the private network cannot prevent loops on the ISP network. VPLS uses full-mesh PWs and split horizon to prevent loops.

- The PEs in a VSI must be fully meshed. That is, a PE must create a tree to every other PE in the VSI.
- Each PE must support split horizon to avoid loops. Split horizon requires that packets received from a PW in a VSI should not be forwarded to other PWs in the VSI. Any two PEs in a VSI must communicate over a direct PW, which is why full-mesh PWs are required between PEs in a VSI.

The full-mesh PEs and split horizon ensure route reachability and prevent loops on a VPLS network. When a CE is connected to multiple PEs, or CEs on the same VPLS VPN are interconnected, VPLS cannot avoid loops. In such a situation, other methods must be used to prevent loops.

STP can run on an L2VPN private network, and all STP Bridge Protocol Data Units (BPDUs) are transparently transmitted over the ISP network.

13.2.6 Inter-AS VPLS

Inter-AS VPLS refers to the application of the VPLS across multiple autonomous systems (ASs). There are two inter-AS VPLS implementation modes: Option A and Option C.

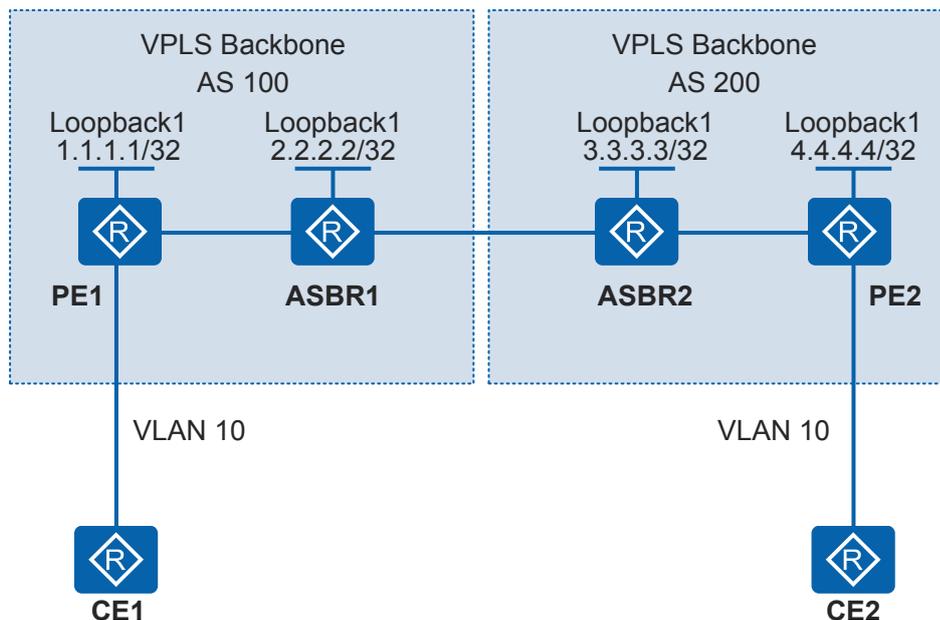
When using the inter-AS VPLS, you do not need to consider the learning or forwarding functions of VSIs, but consider the establishment of PWs between PEs. In this manner, the inter-AS VPLS has the same principle and implementation method as those of inter-AS L2VPN.

In Option A, configurations are easy, without the need to run MPLS between ASBRs or perform particular configurations for inter-AS communications. Nevertheless, this mode has the poor expansion ability and higher requirements for PEs. This mode is applicable in the early servicing stage when the number of inter-AS VPNs is small.

Inter-AS Martini VPLS (Option A)

Figure 13-9 shows the networking of the inter-AS Martini VPLS (Option A).

Figure 13-9 Networking diagram of inter-AS Martini VPLS (Option A)



The inter-AS Martini VPLS (Option A) is implemented as follows:

- An IGP protocol is configured on the backbone network to implement the communications between devices in the same AS.
- The basic MPLS functions are enabled on the backbone network; the dynamic LSP is set up between the PE and ASBR in the same AS; a remote LDP session is set up between the PE and ASBR if they are not directly connected.
- The VPLS connection is set up between the PE and ASBR in the same AS.

Inter-AS Martini VPLS (Option C)

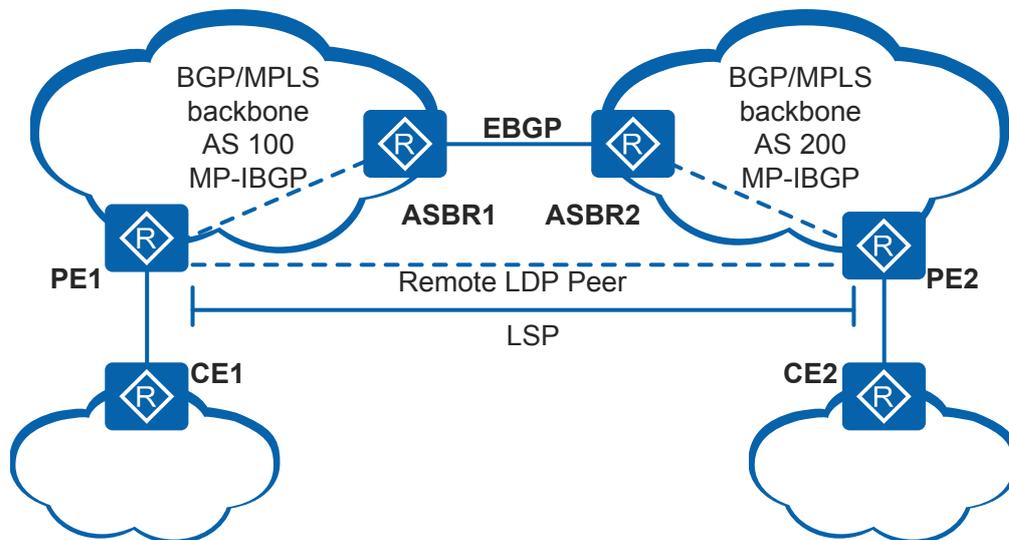
The OptionA scheme requires ASBRs to participate in the distribution and maintenance of PW labels. When multiple inter-AS PWs exist in each AS, ASBRs may be a bottleneck in network expansion.

Inter-AS VPWS Option C avoids this problem by freeing ASBRs from setting up and maintaining PWs. PW labels are directly switched between PEs, as shown in [Figure 13-10](#).

In inter-AS VPWS Option C:

- ASBRs advertise labeled IPv4 routes to PEs in their respective ASs through Multiprotocol Interior Border Gateway Protocol (MP-IBGP), and advertise labeled IPv4 routes received by PEs in their respective ASs to the ASBR peers in other ASs. ASBRs in the intermediate AS also advertise labeled IPv4 routes. As a result, an LDP LSP is set up between the ingress PE and the egress PE.
- PEs in different ASs set up remote MPLS LDP sessions to exchange PW information.

Figure 13-10 Networking diagram of inter-AS Martini VPLS (Option C)



Inter-AS VPWS Option C has the following advantages:

- Similar to the network on which L2VPN users belong to the same AS, intermediate devices do not need to store L2VPN information.
- Only PEs need to store L2VPN information. The devices in intermediate ASs only need to function as ordinary ASBRs that support IP forwarding and do not need to support L2VPN. Inter-AS virtual private wire service (VPWS) Option C is preferred when users need to communicate across a large number of ASs.

13.3 Application Scenarios for VPLS

This section describes the application scenarios for VPLS.

13.3.1 VPLS Application in Individual Services

Service Overview

Individual services such as high speed Internet (HSI), voice over IP (VoIP), and broadband TV (BTV) are usually carried over carriers' metropolitan area networks (MANs).

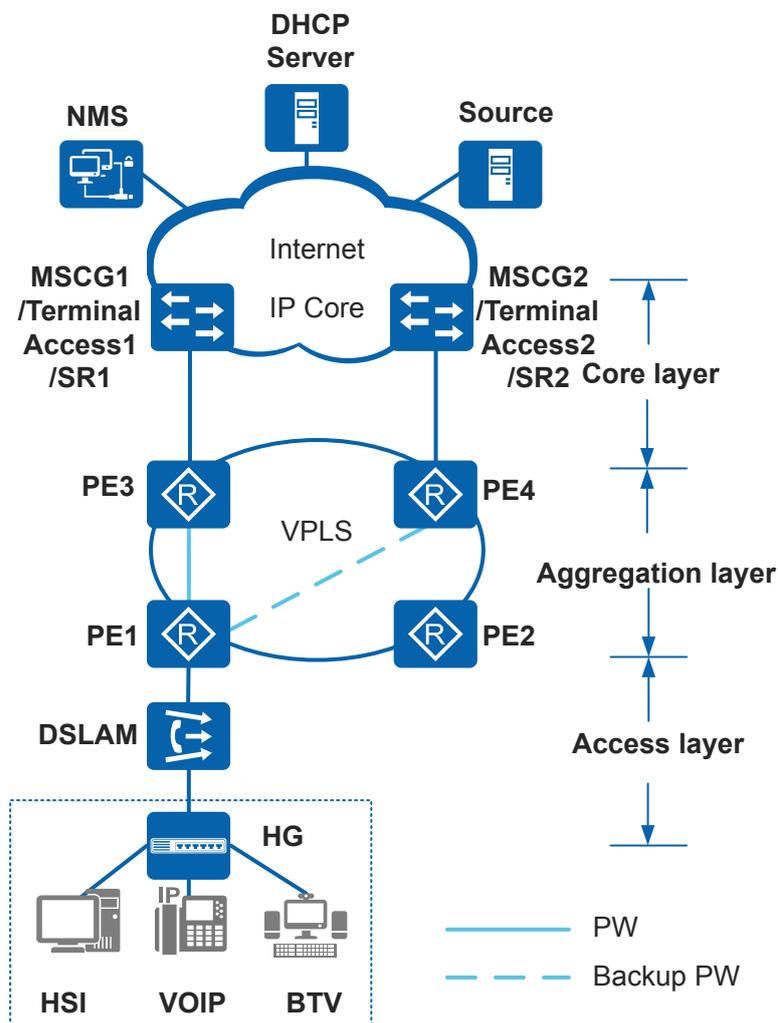
Individual service gateways (such as SR and terminal access gateway) are deployed at the MAN egress. Layer 2 packets of a user need to be transparently transmitted to a service gateway using VPLS or VLL technology. If Layer 2 packets are terminated on a PE and forwarded through Layer 3 routing, user information in the Layer 2 packets is lost. The service gateway fails to control the user because it cannot receive the user information. When the primary and secondary service gateways are deployed on the MAN, user traffic needs to be dual homed to the access service gateways. The VPLS technology must be used to achieve this goal.

Networking Description

Individual services are transmitted to the Internet over the access layer, convergence layer, and core layer of a MAN. **Figure 13-11** shows the typical networking for individual services.

- HSI services access the Internet over the MAN.
- VoIP services request IP addresses from the Dynamic Host Configuration Protocol (DHCP) server over the MAN.
- BTV multicast members apply for BTV services from multicast sources over the MAN.

Figure 13-11 Typical networking for individual services



Feature Deployment

VPLS is configured on PEs to transparently transmit traffic between them. **Figure 13-11** uses LDP VPLS as an example to show VPLS configuration:

- Access-layer devices

VLANs are configured to differentiate different types of users.

PPPoE over AAL5 (PPPoEoA) and PPP over AAL5 (PPPoA) are configured to allow access of HSI users through dialup.

Multicast VLAN and IGMP snooping are configured to transmit multicast services.
- Aggregation-layer devices

Interior Gateway Protocols (IGPs) are configured on PEs so that these PEs can communicate with each other.

Basic MPLS functions are configured on PEs so that these PEs can establish remote LDP sessions.

MPLS L2VPN and VSIs are configured on PEs.

A VPLS daisy chain is deployed on PEs to transmit multicast services.
- Core-layer devices

Authentication and accounting features are configured on terminal access gateway so that terminal access gateway can terminate HSI services.

IGPs are configured on Service Routers (SRs) so that these SRs can communicate with each other.

Basic MPLS functions are configured on SRs.

DHCP relay is configured on SRs, allowing VoIP users to obtain IP addresses from DHCP servers.

Layer 3 multicast features are configured on SRs so that these SRs can communicate with multicast sources.

13.3.2 VPLS Application in Enterprise Services

Service Overview

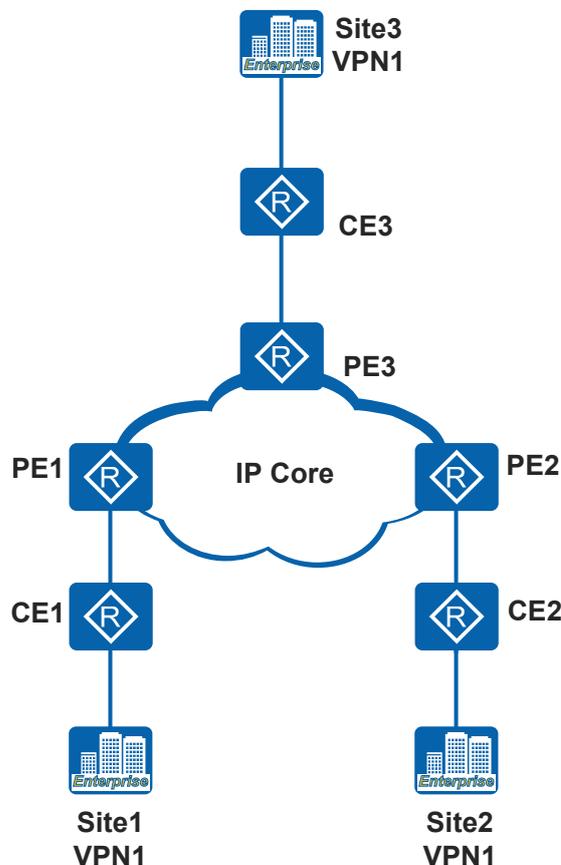
As enterprises set up more and more branches in different regions and office flexibility increases, applications such as instant messaging and teleconferencing are increasingly widely used. This imposes high requirements for E2E datacom technologies. A network capable of providing P2MP services is the key to datacom function implementation. To ensure enterprise data security, secure, reliable, and transparent data channels must be provided for multipoint transmission.

An enterprise has multiple branches located in different regions across a MAN. Layer 2 service packets between enterprise branches need to be transmitted over the MAN using the VPLS technology, so that the enterprise branches in different regions can communicate with each other.

Networking Description

Enterprise services are transmitted over a MAN. **Figure 13-12** shows a typical VPLS network transmitting enterprise services. An enterprise has multiple branches. Site1, Site2, and Site3 are R&D departments. VPLS features are deployed to implement Layer 2 network communication between the sites.

Figure 13-12 Typical networking for enterprise services



Feature Deployment

VPLS is configured on PEs to transparently transmit traffic between PEs. From the perspective of enterprise users, the public network is like a Layer 2 switch. Figure 13-12 uses LDP VPLS as an example to show VPLS configuration:

- Access-layer devices
VLAN is configured to differentiate different types of enterprise users.
- Convergence-layer devices
An IGP is configured on PEs for these PEs to communicate with each other.
Basic MPLS functions are configured on PEs so that these PEs can establish remote LDP sessions.
MPLS L2VPN and VSIs are configured on PEs. Dual-homing is used on a VPLS network to protect traffic.
Limit on the number of learned MAC addresses and traffic suppression are configured on PEs to protect data.

13.4 Licensing Requirements and Limitations for VPLS

Involved Network Elements

None

License Requirements

For VPLS-capable devices, their licensing requirements for the VPLS function are as follows:

- AR1200-S series: VPLS is a basic feature of the device and is not under license control.
- AR2200-S&AR3200-S series: By default, VPLS function is disabled on a new device. To use the VPLS function, apply for and purchase the following license from the Huawei local office.
 - AR2200-S series: AR2200 value-added service package for data services
 - AR3200-S series: AR3200 value-added service package for data services

Feature Limitations

- The device supports IPv4 VPLS only.
- The device supports only the unqualified mode in which the device learns MAC addresses.
- BPDUs including LACP, LLDP, and OAM packets cannot be transparently transmitted on a VPLS network by default. The device can transparently transmit BPDUs only after transparent transmission of BPDUs is enabled. For details, see (Optional) Configuring Transparent Transmission of BPDUs.

If both transparent transmission of BPDUs and EFM are enabled, the device terminates OAM packets locally.

- The device can select LSP tunnels to transmit the VPLS service.

The AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S series routers do not support VPLS.

13.5 Default Settings for VPLS

This section provides the default settings for VPLS.

Table 13-8 Default settings for VPLS

Parameter	Default Setting
MTU value in the VSI view	1500
LDP MAC Withdraw message sent by the VSI	Disabled
Encapsulation type of an interface in the VSI view	VLAN

Parameter	Default Setting
Statistics about the public network traffic on a specified Martini VPLS PW	Disabled

13.6 Configuring Martini VPLS

This section describes how to configure VPLS in Martini mode using LDP as the signalling protocol.

Pre-configuration Task

Before configuring Martini VPLS, complete the following tasks:

- Configuring the LSR ID on the PEs and Ps and enabling MPLS and MPLS LDP
- Establishing tunnels between PEs for transmitting data
- Setting up an LDP session if PEs are indirectly connected

Configuration Procedure

To configure Martini VPLS, perform the following configurations on PEs at both ends of a PW.

13.6.1 Creating a VSI and Configuring LDP Signaling

Context

When using LDP as the PW signaling, you must configure the VSI ID for a VSI. VSI IDs differentiate VSIs, and you can use these VSI IDs during PW signaling negotiation.

Do as follows on the PEs of the two ends of the PW:

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mpls l2vpn**

MPLS L2VPN is enabled and the MPLS L2VPN view is displayed.

Step 3 Run **quit**

Return to the system view.

Step 4 Run **vsi vsi-name static**

A VSI is created and the VSI view is displayed.

Step 5 Run **pwsignal ldp**

The PW signaling protocol is specified as LDP and the VSI-LDP view is displayed.

By default, no signaling mode is configured for a VSI.

Step 6 Run **vsi-id** *vsi-id*

The VSI ID is configured.

By default, no VSI ID is set.

 **NOTE**

The two ends of the VSI must agree on the same *vsi-id*.

The VSI exists only on the PE. One PE can have multiple VSIs. One VPLS on a PE has only one VSI.

Step 7 Run **peer** *peer-address* [**negotiation-vc-id** *vc-id*]

The VSI peer is configured.

By default, no peer is configured for a VSI.

Step 8 Run **peer** *peer-address* [**negotiation-vc-id** *vc-id*] **pw** *pw-name*

A PW is created and the VSI-LDP-PW view is displayed.

By default, no PW is created.

 **NOTE**

If you have created a PW, you can run the command **pw** *pw-name* to enter the VSI-LDP-PW view in the VSI-LDP view.

Step 9 (Optional) Run **undo interface-parameter-type vccv**

The device is configured to delete the VCCV byte following the interface parameter in Mapping packets.

By default, a mapping packet carries the VCCV byte.

Step 10 Run **quit**

Return to the VSI-LDP view.

Step 11 (Optional) Configure common VSI parameters.

Common VSI parameters include the VSI encapsulation mode, MTU for negotiation, and VSI description.

1. Run **quit**

Return to the VSI view.

2. Run **encapsulation** { **ethernet** | **vlan** }

The VSI encapsulation type is configured.

By default, the encapsulation type of the interface is VLAN.

3. Run **mtu** *mtu-value*

The maximum transmission unit (MTU) for packets sent by the VSI is configured.

The default value of MTU in the VSI view is 1500.

When configuring MTUs, note that MTUs of VSIs created for the same VPLS on different PEs must be the same.

 **NOTE**

- When an interface is bound to a VSI, the MTU can be configured in the interface view but does not take effect. The PW signaling uses the MTU that is configured in the VSI for PW MTU negotiation.
- On the router, the MTU value is used only for signaling negotiation and does not limit the size of forwarded packets.

4. Run **description** *description*

The VSI description is configured.

By default, the description of a VSI is empty.

---End

13.6.2 Binding VSIs to AC Interfaces

Context

Based on the type of link between a PE and a CE, a VSI is bound to an AC interface on the PE in one of the following modes:

- Binding the VSI with the Ethernet interface or GE interface when the PE and the CE are connected through the Ethernet interface
- Binding the VSI with the Ethernet sub-interface or GE sub-interface when the PE and the CE are connected through the Ethernet sub-interface

The sub-interfaces can be dot1q sub-interfaces, QinQ sub-interfaces, VLAN mapping sub-interfaces, or VLAN stacking sub-interfaces. For details on how to access the VPLS through a sub-interface, see *Configuring a Dot1q Termination Sub-interface and Connecting It to an L2VPN* and *Configuring a QinQ Termination Sub-interface and Connecting It to an L2VPN* in the *Huawei AR100-S&AR110-S&AR120-S&AR150-S&AR160-S&AR200-S&AR1200-S&AR2200-S&AR3200-S Series Enterprise Routers Configuration Guide - Ethernet*.

When Ethernet or GE interfaces are used as AC interfaces, the outer Tags carried in the packets sent from the AC to the PW are U Tags (inserted by user devices, which are meaningless to the SP) by default.

When sub-interfaces are used as AC interfaces, the outer Tags carried in the packets sent from the AC to the PW are P Tags (inserted by SP devices, which are used to differentiate user traffic) by default.

 **NOTE**

- In the VPLS application, different CEs are transparently connected in the same LAN segment through VSIs, and the IP addresses of the CEs must be different. The IP address of the interface that connects the PE to the CE and the IP address of the CE must be in different network segments. Otherwise, the local CE may learn incorrect ARP entries. This leads to traffic loss between CEs in the same VSI.
- When used on an AC-side interface, the **qinq protocol** command enables the system to identify incoming packets and Tag outgoing packets with the TPID. The TPID must use the default value 8100. Do not change the TPID.

Procedure

- Bind a VSI to an Ethernet interface.
Do as follows on the PEs at both ends of a PW:

- a. Run **system-view**
The system view is displayed.
 - b. Run **interface** *interface-type interface-number*
The Ethernet interface view is displayed.
 - c. (Optional) Run **undo portswitch**
The Layer 2 interface is configured as a Layer 3 interface.
 - d. (Optional) Run **mpls l2vpn default vlan**
The default VLAN of the Ethernet interface is set to VLAN 0.
By default, no default VLAN is configured for an Ethernet interface.
If the remote PE device can only receive packets with VLAN tags, this step is required before you bind a VSI to the local Ethernet interface.
 - e. Run **I2 binding vsi** *vsi-name*
The VSI is bound to the Ethernet interface.
- Bind a VSI to an Ethernet sub-interface.
Do as follows on the PEs at both ends of a PW:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **interface** *interface-type interface-number.subinterface-number*
The Ethernet sub-interface view is displayed.
 - c. Run one of the following commands to configure an Ethernet sub-interface based on site requirements.
 - Run **dot1q termination vid** *low-pe-vid*
The single VLAN ID for dot1q encapsulation on a sub-interface is configured.
 - Run **qinq termination pe-vid** *pe-vid ce-vid ce-vid1 [to ce-vid2]*
The Double VLAN IDs for QinQ encapsulation on a sub-interface is configured.
 - Run **vlan mapping vid** *vlan-id1 map-vlan vlan-id2*
Single-tagged VLAN mapping is configured on the sub-interface.
 - Run **vlan stacking vid** *vlan-id1 [to vlan-id2] pe-vid vlan-id3*
VLAN stacking is configured on the sub-interface.
 - d. (Optional) Run **mpls l2vpn default vlan**
The default VLAN of the Ethernet sub-interface is set to VLAN 0.
By default, no default VLAN is configured for an Ethernet sub-interface.
If the remote PE device can only receive packets with VLAN tags, this step is required before you bind a VSI to the local Ethernet sub-interface.
 - e. Run **I2 binding vsi** *vsi-name*
The VSI is bound with the Ethernet sub-interface.

----End

13.6.3 Verifying the Martini VPLS Configuration

Prerequisites

All Martini VPLS configurations are complete.

Procedure

- Run the **display vsi** [*name vsi-name*] [*verbose*] command to check information about a VPLS VSI.
- Run the **display l2vpn ccc-interface vc-type** { *all* | *vc-type* } [*down* | *up*] command to check information about the interface used by an L2VPN connection.
- Run the **display vsi remote ldp** [[*router-id ip-address*] [*pw-id pw-id*] | *verbose*] command to check information about a remote VSI.
- Run the **display vpls connection** [*ldp* | *vsi vsi-name*] [*down* | *up*] [*verbose*] command to check information about a VPLS connection.
- Run the **display vpls forwarding-info** [*vsi vsi-name* [*peer peer-address* [*negotiation-vc-id vc-id* | *remote-site site-id*]] | *state* { *up* | *down* }] [*verbose*] command to check forwarding information of all VSIs.
- Run the **display vsi services** { *all* | *vsi-name* | *interface interface-type interface-number* | *vlan vlan-id* } command to check information about the AC interface associated with the VSI.
- Run the **display vsi pw out-interface** [*vsi vsi-name*] command to check information about the outgoing interface of a PW in a VSI.
- Run the **display mpls label-stack vpls vsi** *vsi-name* *peer peer-ip-address* *vc-id vc-id* command to check the information about label stacks in a VPLS scenario.

----End

13.7 (Optional) Configuring Inter-AS Martini VPLS

Inter-AS VPLS allows a VPLS network to span multiple ASs on an MPLS backbone network.

Usage Scenario

If multiple ASs exist on an MPLS backbone network, the VPLS network carried over the MPLS backbone network must be an inter-AS VPLS network. Currently, the following inter-AS VPLS solutions are available:

- Inter-AS VPLS Option A:
This solution is recommended for scenarios where few inter-AS VPLS PWs are required. In this solution, ASBRs must support VSIs, manage VPLS label blocks, and reserve interfaces for inter-AS PWs. This solution poses high performance requirements for ASBRs, but does not require inter-AS configurations to be performed on ASBRs.
- Inter-AS VPLS Option C:
This solution is recommended for scenarios where large numbers of inter-AS Martini L2VPN are required. In this solution, ASBRs do not need to create or maintain PWs and are not a significant factor in network expansion.

Configuration Procedure

Perform one or more of the following configurations as required.

13.7.1 Configuring Inter-AS Martini VPLS in OptionA Mode

When VPLS is deployed in a large area, PEs may belong to different ASs. In this case, PWs cannot be established between PEs through LDP. To solve the problem in a normal network, configure inter-AS Martini VPLS.

Pre-configuration Task

Before configuring inter-AS Martini VPLS in OptionA mode, complete the following tasks:

- Configuring IGP for the MPLS backbone network of each AS to ensure IP connectivity of the backbone network within an AS.
- Configuring basic MPLS functions on the MPLS backbone network of each AS.
- Configuring MPLS LDP and establishing the LDP LSP for the MPLS backbone network of each AS.
- Establishing a tunnel between the PE and ASBR within an AS.

Context

The configuration is described as follows:

- Configuring Martini VPLS in each AS.
- An ASBR considers a peer ASBR as a CE.
- No inter-AS configuration needs to be performed on ASBRs.
- No IP address needs to be configured for the interfaces connecting ASBRs.

For details, see [13.6 Configuring Martini VPLS](#).

NOTE

In inter-AS Martini VPLS OptionA, each ASBR must reserve an AC interface for each inter-AS VC. OptionA can be used when the number of inter-AS VCs is small. Compared with the L3VPN, the inter-AS L2VPN OptionA consumes more resources and requires more configuration workload. Therefore, the inter-AS L2VPN OptionA is not recommended.

Verifying the Configuration

- Run the **display vsi** [**name** *vsi-name*] [**verbose**] command to check information about a VPLS VSI.
- Run the **display vsi remote ldp** [**router-id** *ip-address*] [**pw-id** *pw-id*] command to check information about a remote VSI.
- Run the **display vpls connection** [**ldp** | **vsi** *vsi-name*] [**down** | **up**] [**verbose**] command to check information about a VPLS connection.
- Run the **ping vpls mac** *mac-address* **vsi** *vsi-name* [**vlan** *vlan-id*] [**-c** *count*] [**-m** *time-value*] [**-s** *packsize*] [**-t** *timeout*] [**-exp** *exp*] [**-r** *replymode*] [**-h** *t1*] * command to check the connectivity of Layer 2 links on the VPLS network.
- Run the **trace vpls mac** *mac-address* **vsi** *vsi-name* [**vlan** *vlan-id*] [**-t** *timeout*] [**-f** *first-ttl*] [**-m** *max-ttl*] [**-exp** *exp*] [**-r** *replymode*] * command to check the PEs and P that packets

pass from the sender to the receiver and check the connectivity of Layer 2 links, which helps locate the faulty node on the network.

 **NOTE**

In OptionA mode, the **ping** and **trace** functions support intra-AS detection.

13.7.2 Configuring Inter-AS Martini VPLS in OptionC Mode

In inter-AS LDP VPLS Option C, ASBRs do not need to maintain inter-AS LDP VPLS information or reserve interfaces for inter-AS LDP VPLS PWs. As LDP VPLS information is exchanged only between PEs, this solution requires few resources and is easy to deploy.

Pre-configuration Task

Before configuring inter-AS Martini VPLS in OptionC mode, complete the following tasks:

- Configure IGP for the MPLS backbone network of each AS to ensure IP connectivity of the backbone network within an AS.
- Configure basic MPLS functions on the MPLS backbone network of each AS.
- Configure MPLS LDP and establish the LDP LSP for the MPLS backbone network of each AS.
- Set up the IBGP peer relationship between the PE and the ASBR of the same AS and set up the EBGP peer relationship between two inter-AS ASBRs.

Procedure

Step 1 Configure the capability to exchange labeled IPv4 routes.

- Perform the following steps on each PE:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp** { *as-number-plain* | *as-number-dot* }
The Border Gateway Protocol (BGP) view is displayed.
 - c. Run **peer** *ipv4-address* **as-number** *as-number*
An IBGP peer relationship is established between the local PE and ASBR in the same AS.
 - d. Run **peer** *ipv4-address* **connect-interface loopback** *interface-number*
A loopback interface is specified as the outbound interface of the BGP session.
 - e. Run **peer** *ipv4-address* **label-route-capability**
Exchange of the labeled IPv4 routes with the ASBR in the same AS is enabled.
- Perform the following steps on each ASBR:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **interface** *interface-type* *interface-number*
The view of the interface connecting to the peer ASBR is displayed.
 - c. Run **ip address** *ip-address* { *mask* | *mask-length* }
An IP address is configured for the interface.

- d. Run **mpls**
MPLS is enabled on the interface.
- e. Run **mpls ldp**
MPLS LDP is enabled on the interface.
- f. Run **quit**
Return to the system view.
- g. Run **bgp** { *as-number-plain* | *as-number-dot* }
The BGP view is displayed.
- h. Run **peer** *ipv4-address* **as-number** *as-number*
An IBGP peer relationship is established between the local PE and the remote PE in the same AS.
- i. Run **peer** *ipv4-address* **connect-interface loopback** *interface-number*
A loopback interface is specified as the outbound interface of the BGP session.
- j. Run **peer** *ipv4-address* **label-route-capability**
Exchange of the labeled IPv4 routes with the remote PE in the same AS is enabled.
- k. Run **peer** *ipv4-address* **as-number** *as-number*
The peer ASBR is specified as the EBGP peer.
- l. Run **peer** *ipv4-address* **label-route-capability** [**check-tunnel-reachable**]
The capability to exchange labeled IPv4 routes with the peer ASBR is configured.
In inter-AS LDP VPLS Option C, an inter-AS LSP must be established and the public network routes advertised between PEs and ASBRs carry MPLS labels.
An EBGP peer relationship must be established between ASBRs in different ASs for them to exchange labeled IPv4 routes.
The public network routes carrying MPLS labels are advertised through MP-BGP. According to RFC 3107, label mappings about routes can be piggybacked inside the BGP Update messages that are used to advertise these routes. This feature is implemented through an extended BGP attribute, which enables BGP peers to process labeled IPv4 routes.
By default, BGP peers cannot process labeled IPv4 routes.

Step 2 Configure a routing policy on the ASBR to control label distribution.

After the routing policy is applied to an ASBR, the ASBR allocates MPLS labels to routes received from a PE in the local AS before advertising the routes to the remote ASBR, and reallocates MPLS labels to labeled IPv4 routes advertised to a PE in the local AS.

By default, an IPv4 route does not carry any MPLS label.

Perform the following steps on each ASBR:

1. Run **system-view**
The system view is displayed.
2. Run **route-policy** *policy-name1* **permit node** *seq-number*
A routing policy applicable to the local PE is created.
The device reallocates MPLS labels to labeled IPv4 routes advertised to a PE in the local AS.

3. Run **if-match mpls-label**

Filtering out labeled IPv4 routes is configured as a matching rule for the routing policy.

4. Run **apply mpls-label**

Allocating MPLS labels to IPv4 routes is configured as an action for the routing policy.

5. Run **quit**

Return to the system view.

6. Run **route-policy policy-name2 permit node seq-number**

A routing policy applicable to the peer ASBR is created.

When an ASBR advertises the routes received from a PE in the same AS to the peer ASBR, the ASBR allocates MPLS labels to the routes.

7. Run **apply mpls-label**

Allocating MPLS labels to IPv4 routes is configured as an action for the routing policy.

8. Run **quit**

Return to the system view.

9. Run **bgp { as-number-plain | as-number-dot }**

The BGP view is displayed.

10. Run **peer ipv4-address route-policy policy-name1 export**

The routing policy applicable to the local PE is applied.

11. Run **peer ipv4-address route-policy policy-name2 export**

The routing policy applicable to the peer ASBR is applied.

Step 3 Establish remote MPLS LDP sessions between PEs.

Configure an ASBR to send the loopback interface IP addresses of a PE device used for peer relationship establishment to the ASBRs of other ASs and the PEs in different ASs.

- Perform the following steps on each ASBR:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **bgp { as-number-plain | as-number-dot }**
The BGP view is displayed.
 - c. Run **network ip-address [mask | mask-length] [route-policy route-policy-name]**
The capability to advertise the local PE's loopback interface address used for BGP sessions to the peer ASBR is configured.
 - d. Run **quit**
Return to the system view.
- Perform the following steps on each PE:
 - a. Run **system-view**
The system view is displayed.
 - b. Run **mpls ldp remote-peer peer-name**
The name of the remote MPLS LDP session is specified.

Remote MPLS LDP sessions need to be established between PEs, so that they can directly exchange PW information.

- c. Run **remote-ip** *ip-address*

The peer IP address of the remote MPLS LDP session is specified.

Step 4 Configure Martini VPLS on each PE. For configuration details, see [13.6 Configuring Martini VPLS](#).

---End

13.8 (Optional) Setting Related Parameters for a VSI

This section describes how to set related parameters for a VSI.

Pre-configuration Task

Before setting VSI-related parameters, complete the following tasks:

- [13.6 Configuring Martini VPLS](#)

Configuration Procedure

After creating a VSI and assigning a signaling protocol to it, you can adjust common parameters of the VSI. According to different applicable environments, you can determine whether to modify MAC address learning modes and MAC address entries.

Perform the following operations on the PE.

13.8.1 Configuring a PE to Send MAC Withdraw Messages to Remove MAC Address Entries

Context

The local device cannot detect failure of the AC. When the AC is Down, the VSI remains Up and the device does not delete the corresponding MAC address entry. This results in a black hole when the local device sends data flows to the remote device. The MAC address withdraw function solves this problem by enabling the local VSI to delete the local MAC address and notify all remote peers when AC is faulty and the VSI remains Up.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **vsi** *vsi-name* [**static**]

The VSI view is displayed.

Step 3 Run **pwsignal ldp**

LDP is configured as the PW signaling protocol and the VSI-LDP view is displayed.

Step 4 Run **mac-withdraw enable**

The PE is configured to send a MAC Withdraw message when the AC or PW status changes.

By default, this function is disabled.

Step 5 Run **interface-status-change mac-withdraw enable**

The PE is configured to send LDP MAC Withdraw messages to all peers when the status of the AC-side interface bound to the VSI changes.

By default, the PE does not send LDP MAC Withdraw messages to all peers when the status of the AC-side interface bound to the VSI changes.

----End

13.8.2 Configuring MAC Withdraw Loop Detection

Context

This section describes how to configure Media Access Control (MAC) Withdraw loop detection to prevent MAC Withdraw message loops.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mpls l2vpn**

Layer 2 virtual private network (L2VPN) is enabled, and the L2VPN view is displayed.

Step 3 Run **vpls mac-withdraw loop-detect enable**

MAC Withdraw loop detection is enabled.

By default, MAC Withdraw loop detection is disabled.

----End

Verifying the Configuration

- Run the **display vsi [name vsi-name] verbose** command to check the MAC Withdraw configuration.
- Run the **display vsi [name vsi-name] mac-withdraw loop-detect** command to check information about MAC Withdraw message loops.

13.8.3 Configuring MAC Address Learning

Context

In VPLS, packets are forwarded according to MAC address forwarding entries. In most cases, MAC address learning can be performed automatically. Nevertheless, to prevent attacks and troubleshoot faults, you can adopt the VSI-based MAC address management mechanism provided by the router.

Do as follows on the PEs of the two ends of the PW:

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **mac-address aging-time** *aging-time*

The aging time of MAC address entries for the VPLS is configured.

By default, the aging time of dynamic MAC address entries is 300 seconds.

Step 3 Run **mac-address static** *mac-address interface-type interface-number vsi vsi-name*

Static MAC address entries are configured.

By default, no static MAC address entry is configured in the system.

Step 4 Run **mac-address blackhole** *mac-address vsi vsi-name*

MAC address blackhole entries are configured.

By default, no blackhole MAC address entry is configured for the VSI.

Step 5 Run **vsi** *vsi-name* [**static**]

The VSI view is displayed.

Step 6 Run **pwsignal ldp**

The PW signaling protocol is specified as LDP and the VSI-LDP view is displayed.

Step 7 Run **vsi-id** *vsi-id*

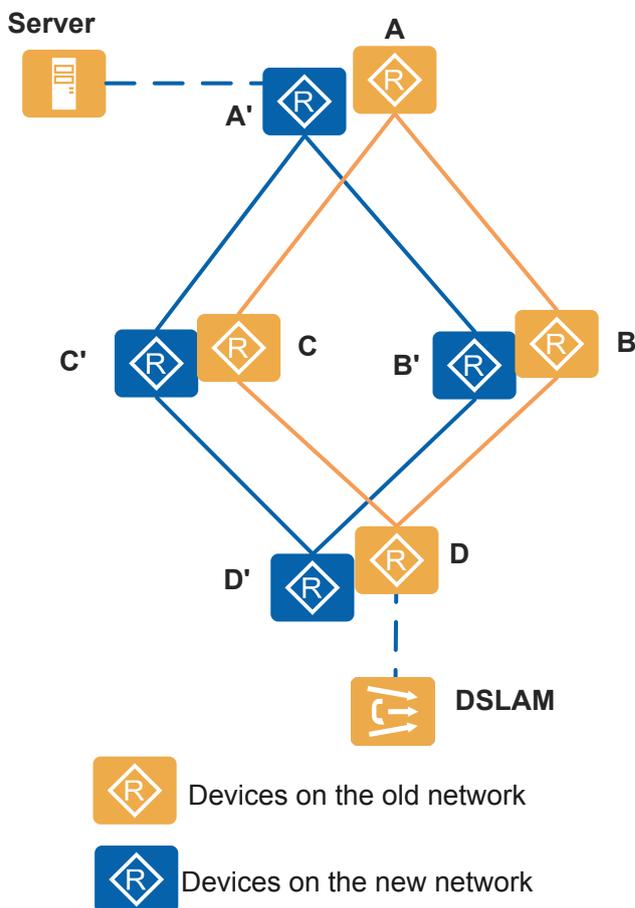
The VSI ID is configured.

----End

13.8.4 Configuring a VSI to Ignore the AC Status

Context

Figure 13-13 Networking diagram of configuring a VSI to ignore the AC status



As shown in [Figure 13-13](#), devices A, B, C, and D are on the old network, while devices A', B', C', and D' are on the new network, if the services running on the old network will switch to the new network, and you want to check whether the VSI on the new network can work normally before the service switchover, you need to configure the VSI to ignore the AC status on D'. After the configuration, the VSI on D' keeps Up before the DSLAM is connected to the new network.

If an AC interface is Down and the PW is Up, the VSI remains Up after being enabled to ignore AC status. If an AC interface is Up and the PW is Down, the VSI remains Up after being enabled to ignore AC status.

Do as follows on the PE (D' in [Figure 13-13](#)):

Procedure

Step 1 Run system-view

The system view is displayed.

Step 2 Run the following commands as required:

- If you want to prevent the status of all the VSIs from being affected by the status of the AC. Run **mpls l2vpn vpls ignore-ac-state**
 - a. The MPLS L2VPN view is display.
 - b. Prevents the status of a VSI from being affected by the status of the AC.
- If you want to prevent the status of one VSI from being affected by the status of the AC. Run:
 - a. `vsi vsi-name [static]`
The VSI view is displayed.
 - b. `ignore-ac-state`
Prevents the status of a VSI from being affected by the status of the AC.

---End

Follow-up Procedure

The **vpls ignore-ac-state** or **ignore-ac-state** are used only before the service switchover between a new VPLS network and an old one. After the service switchover, run the **undo vpls ignore-ac-state** or **undo ignore-ac-state** command to restore the default setting.

13.9 Maintaining VPLS

VPLS maintenance includes collecting, clearing, and viewing traffic statistics on VPLS PWs, enabling or disabling VSIs, clearing MAC address entries, and detecting the VPLS network connectivity.

13.9.1 Collecting Traffic Statistics on a VPLS PW

Context

To analyze the network traffic model, collect traffic statistics on the specified PW.

Perform the following steps on the device configured with the VSI:

Procedure

- Step 1** Run the **system-view** command to enter the system view.
- Step 2** Run the **vsi vsi-name [static]** command to enter the VSI view.
- Step 3** Run the **pwsignal ldp** command to configure LDP as the PW signaling protocol and enter the VSI-LDP view.
- Step 4** Choose either of the following methods to enable traffic statistics collection based on the number of PWs.
 - When there are many PWs:
Run the **traffic-statistics enable** command to enable statistics collection on public network traffic of all PWs in Martini VPLS mode.
 - When there are a few PWs:

Run the **traffic-statistics peer** *peer-address* [**negotiation-vc-id** *vc-id*] **enable** command to enable statistics collection on public network traffic of a specified PW in Martini VPLS mode.

----End

13.9.2 Clearing the Traffic Statistics

Context



The traffic statistics information cannot be restored after you clear it. So, confirm the action before you use the command.

Procedure

- Run the **reset traffic-statistics vsi all** command in the user view to clear the traffic statistics on all VPLS PWs.
- Run the **reset traffic-statistics vsi name** *vsi-name* command in the user view to clear the traffic statistics on all VPLS PWs of a specified VSI.
- Run the **reset traffic-statistics vsi name** *vsi-name* **peer** *peer-address* [**negotiation-vc-id** *vc-id*] command in the user view to clear the traffic statistics on a specified VPLS PW of a specified VSI.

----End

13.9.3 Checking Traffic Statistics on a VPLS PW

Context



Within five minutes, if a PW goes Down, traffic before the PW is Down cannot be used to compute the traffic rate in the five minutes.

After the traffic statistics function is enabled on a VPLS PW, you can run the following commands in any view to view the running status of traffic on the VPLS PW.

Procedure

- Run the **display traffic-statistics vsi** *vsi-name* command to view public network traffic statistics on all PWs of a specified VSI.
- Run the **display traffic-statistics vsi** *vsi-name* **peer** *peer-address* [**negotiation-vc-id** *vc-id*] command to view public network traffic statistics on a specified PW of a specified VSI.

----End

13.9.4 Enabling or Disabling VSI

Context

Sometimes, to halt services, you can disable a VSI temporarily, and then add, cancel, or adjust VSI functions.

Procedure

- Enable VSI
 - a. Run the **system-view** command to enter the system view.
 - b. Run the **vsi** *vsi-name* command to enter the vsi view.
 - c. Run the **undo shutdown** command to enable VSI.
- Disable VSI
 - a. Run the **system-view** command to enter the system view.
 - b. Run the **vsi** *vsi-name* command to enter the vsi view.
 - c. Run the **shutdown** command to disable VSI.



The **shutdown** command affects the PW connection. The AC is Down, and the Layer 2 forwarding table is deleted.

----End

13.9.5 Clearing MAC Address Entries

Context



After MAC address entries are cleared, the entries cannot be restored. Confirm the action before you clear the entries.

Procedure

- Run the **undo mac-address static** *mac-address interface-type interface-number* **vsi** *vsi-name* command to clear MAC address entries for a VSI.
- Run the **undo mac-address** [**dynamic** | **all**] command to clear dynamic, or all MAC address entries.
- Run the **undo mac-address static** command to clear static MAC address entries.
- Run the **undo mac-address blackhole** [**vsi** *vsi-name*] command to clear blackhole MAC address entries.

----End

13.9.6 Checking Connectivity of the VPLS Network

Context

To check connectivity of a VPLS network, configure a VPLS network and run the following functions on PE devices.

Procedure

- Checking VPLS network connectivity
 - Run the **ping vpls mac** *mac-address* **vsi** *vsi-name* [**vlan** *vlan-id* | **-c** *count* | **-m** *time-value* | **-s** *packsize* | **-t** *timeout* | **-exp** *exp* | **-r** *replymode* | **-h** *tll*] * command to check connectivity of the Layer 2 forwarding link on the VPLS network.
 - Run the **ping vpls** [**-c** *echo-number* | **-m** *time-value* | **-s** *data-bytes* | **-t** *timeout-value* | **-r** *reply-mode* | **-exp** *exp-value* | **-v**] * **vsi** *vsi-name* **peer** *peer-address* [**negotiate-vc-id** *vc-id*] command to check connectivity of the link between PEs on the Martini VPLS network.
 - Run the **trace vpls mac** *mac-address* **vsi** *vsi-name* [**vlan** *vlan-id*] [**-t** *timeout* | **-f** *first-ttl* | **-m** *max-ttl* | **-exp** *exp* | **-r** *replymode*] * command to check PEs and P devices along the PW on the VPLS network. This command is also used to check the connectivity of Layer 2 forwarding links and locate faults on the network.
 - Run the **tracert vpls** [**-exp** *exp-value* | **-f** *first-ttl* | **-m** *max-ttl* | **-r** *reply-mode* | **-t** *timeout-value*] * **vsi** *vsi-name* **peer** *peer-address* [**negotiate-vc-id** *vc-id*] [**full-lsp-path**] command to check connectivity of the Martini VPLS network.
- Checking ping/trace packet statistics
 - Run the **display vpls-ping statistics** command to check the number of sent and received VPLS MAC ping packets.
 - Run the **display vpls-trace statistics** command to check the number of sent and received VPLS MAC trace packets.
- Clearing ping/trace packet statistics
 - Run the **reset vpls-ping statistics** command to clear VPLS MAC ping statistics.
 - Run the **reset vpls-trace statistics** command to clear VPLS MAC trace statistics.

----End

13.9.7 Configuring the Upper and Lower Alarm Thresholds for VPLS VCs

Context

A device supports only a limited number of VPLS VCs. After the total number of VPLS VCs created on a device exceeds a certain limit, the device performance deteriorates. To prevent device performance deterioration caused by excessive VPLS VCs, configure the upper and lower alarm thresholds for VPLS VCs. You can flexibly adjust the upper and lower alarm thresholds for VPLS VCs based on actual requirements.

Procedure

- Step 1** Run the **system-view** command to enter the system view.
- Step 2** Run the **mpls l2vpn** command to enter the MPLS L2VPN view.
- Step 3** Run the **mpls l2vpn vsi-pw limit threshold-alarm upper-limit *upper-limit-value* lower-limit *lower-limit-value*** command to configure the upper and lower alarm thresholds for VPLS VCs.

By default, the upper and lower alarm thresholds are 80% and 70% respectively.

- *upper-limit-value* specifies the upper alarm threshold for VPLS VCs. If the proportion of VPLS VCs created to the maximum VPLS VCs allowed reaches this threshold, a VPLS VC threshold-crossing alarm is reported.
- *lower-limit-value* specifies the lower alarm threshold for VPLS VCs. If the proportion of VPLS VCs created to the maximum VPLS VCs allowed falls below this threshold, a VPLS VC threshold-crossing clear alarm is reported.
- *upper-limit-value* must be greater than *lower-limit-value*.

----End

13.9.8 Checking MPLS L2VPN Usage Information

Context

In routine network maintenance, you can learn overall MPLS L2VPN information by checking MPLS L2VPN specifications and usage information.

Procedure

- Run the **display mpls l2vpn resource** command to check MPLS L2VPN specifications and usage information.

----End

13.10 Configuration Examples for VPLS

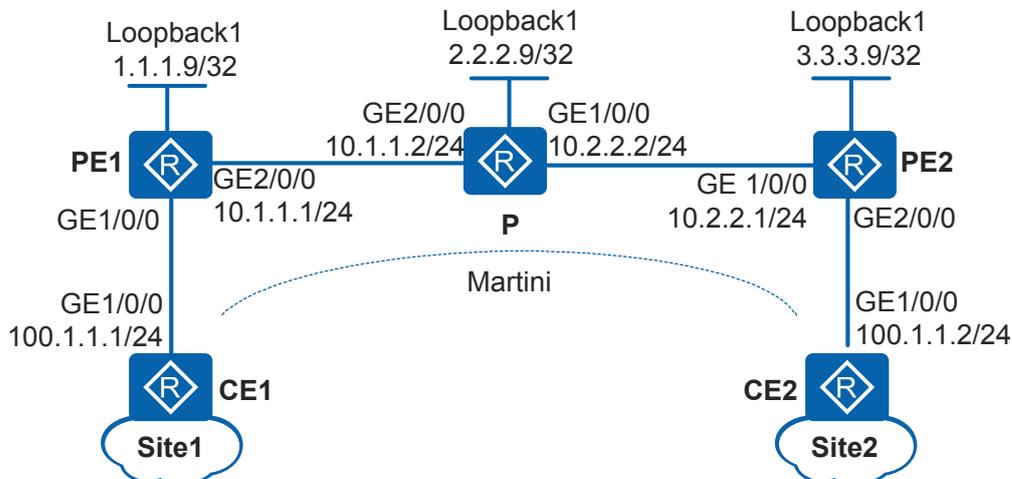
This section provides several configuration examples of different VPLS networking, including networking requirements, configuration notes, configuration roadmap, configuration procedures, and configuration files.

13.10.1 Example for Configuring Martini VPLS

Networking Requirements

Figure 13-14 shows a backbone network built by an enterprise. Few branch sites are distributed on the network (only two sites are shown in this example). Site1 connects to PE1 through CE1 and then connects to the backbone network. Site2 connects to PE2 through CE2 and then connects to the backbone network. Users at Site1 and Site2 need to communicate at Layer 2 and user information needs to be reserved when Layer 2 packets are transmitted over the backbone network.

Figure 13-14 Networking diagram for configuring Martini VPLS



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure transparent transmission of Layer 2 packets over the backbone network using VPLS to enable users at Site1 and Site2 to communicate at Layer 2 and reserve user information when Layer 2 packets are transmitted over the backbone network.
2. Use Martini VPLS to implement Layer 2 communication between CEs on an enterprise network with few sites.
3. Configure the IGP routing protocol on the backbone network to implement data transmission on the public network between PEs.
4. Configure basic MPLS functions and LDP on the backbone network to support VPLS.
5. Establish tunnels for transmitting data between PEs to prevent data from being known by the public network.
6. Enable MPLS L2VPN on PEs to implement VPLS.
7. Create VSIs on PEs, specify LDP as the signaling protocol, and bind VSIs to AC interfaces to implement Martini VPLS.

Procedure

Step 1 Configure IP addresses for interfaces on the CE, PE and P devices according to [Figure 13-14](#).

Configure CE1. The configuration on PE1, P, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure the IGP protocol. OSPF is used in this example.

When configuring OSPF, advertise the 32-bit address of the loopback interface (LSR IDs) on PE1, P and PE2.

Configure PE1. The configuration on P and PE2 is similar to the PE1, and is not mentioned here.

```
[PE1] interface loopback 1
[PE1-LoopBack1] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack1] quit
[PE1] ospf 1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

After the configuration is complete, run the **display ip routing-table** command on PE1, P, and PE2. You can view the routes learned by PE1, P, and PE2 from each other.

Step 3 Configure basic MPLS functions and LDP.

Configure PE1.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

Configure the P.

```
[P] mpls lsr-id 2.2.2.9
[P] mpls
[P-mpls] quit
[P] mpls ldp
[P-mpls-ldp] quit
[P] interface gigabitethernet 2/0/0
[P-GigabitEthernet2/0/0] mpls
[P-GigabitEthernet2/0/0] mpls ldp
[P-GigabitEthernet2/0/0] quit
[P] interface gigabitethernet 1/0/0
[P-GigabitEthernet1/0/0] mpls
[P-GigabitEthernet1/0/0] mpls ldp
[P-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls lsr-id 3.3.3.9
[PE2] mpls
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] mpls
[PE2-GigabitEthernet1/0/0] mpls ldp
[PE2-GigabitEthernet1/0/0] quit
```

After the configuration is complete, run the **display mpls ldp session** command on PE1, P and PE2. You can see that peer relationships are set up between PE1 and P, and between P and PE2. The status of the peer relationship is Operational. Run the **display mpls lsp** command to view the LSP status.

Step 4 Set up remote LDP sessions between PEs.

Configure PE1.

```
[PE1] mpls ldp remote-peer 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] remote-ip 3.3.3.9
[PE1-mpls-ldp-remote-3.3.3.9] quit
```

Configure PE2.

```
[PE2] mpls ldp remote-peer 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] remote-ip 1.1.1.9
[PE2-mpls-ldp-remote-1.1.1.9] quit
```

After the configuration is complete, run the **display mpls ldp session** command on PE1 or PE2, and you can see that the status of the peer relationship between PE1 and PE2 is Operational. That is, the peer relationship is set up.

Take the display on PE1 for example.

```
[PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID                Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
2.2.2.9:0             Operational DU   Passive  0000:00:11  46/45
3.3.3.9:0             Operational DU   Passive  0000:00:01  8/8
-----
TOTAL: 2 session(s) Found.
```

Step 5 Enable MPLS L2VPN on PEs.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
```

Step 6 Configure LDP VPLS on PEs.

Configure PE1.

```
[PE1] vsi a2 static
[PE1-vsi-a2] pwsignal ldp
[PE1-vsi-a2-ldp] vsi-id 2
[PE1-vsi-a2-ldp] peer 3.3.3.9
[PE1-vsi-a2-ldp] quit
[PE1-vsi-a2] quit
```

Configure PE2.

```
[PE2] vsi a2 static
[PE2-vsi-a2] pwsignal ldp
[PE2-vsi-a2-ldp] vsi-id 2
[PE2-vsi-a2-ldp] peer 1.1.1.9
[PE2-vsi-a2-ldp] quit
[PE2-vsi-a2] quit
```

Step 7 Bind the interface on the PE to the VSI.

Configure PE1.

```
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] l2 binding vsi a2
[PE1-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] l2 binding vsi a2
[PE2-GigabitEthernet2/0/0] quit
```

Step 8 Verify the configuration.

After the network becomes stable, run the **display vsi name a2 verbose** command on PE1, and you can see that VSI a2 sets up a PW to PE2, and the status of the VSI is Up.

```
[PE1] display vsi name a2 verbose

***VSI Name          : a2
 Administrator VSI   : no
 Isolate Spoken      : disable
 VSI Index           : 0
 PW Signaling        : ldp
 Member Discovery Style : static
 PW MAC Learn Style  : unqualify
 Encapsulation Type  : vlan
 MTU                  : 1500
 Diffserv Mode       : uniform
 Service Class       : --
 Color               : --
 DomainId            : 255
 Domain Name         :
 Ignore AcState      : disable
 P2P VSI             : disable
 Create Time         : 0 days, 0 hours, 1 minutes, 3 seconds
 VSI State           : up

 VSI ID              : 2
 *Peer Router ID     : 3.3.3.9
 Negotiation-vc-id   : 2
 primary or secondary : primary
 ignore-standby-state : no
 VC Label            : 1024
 Peer Type           : dynamic
 Session             : up
 Tunnel ID           :
 Broadcast Tunnel ID : 0x0
 Broad BackupTunnel ID : 0x0
 CKey                : 6
 NKey                : 5
 Stp Enable          : 0
 PwIndex             : 0
 Control Word        : disable
 BFD for PW          : unavailable

 Interface Name      : GigabitEthernet1/0/0
 State               : up
 Access Port         : false
 Last Up Time        : 2017/07/02 17:13:47
 Total Up Time       : 0 days, 0 hours, 1 minutes, 3 seconds

**PW Information:

 *Peer Ip Address    : 3.3.3.9
 PW State            : up
 Local VC Label      : 4096
 Remote VC Label     : 4096
 Remote Control Word : disable
 PW Type             : label
 Local VCCV          : alert lsp-ping
 Remote VCCV         : alert lsp-ping
```

```
Tunnel ID          : 0x1a
Broadcast Tunnel ID : 0x1a
Broad BackupTunnel ID : 0x0
Ckey               : 0x6
Nkey               : 0x5
Main PW Token      : 0x1a
Slave PW Token     : 0x0
Tnl Type           : LSP
OutInterface       : GigabitEthernet2/0/0
Backup OutInterface :
Stp Enable         : 0
PW Last Up Time    : 2017/07/02 17:14:47
PW Total Up Time   : 0 days, 0 hours, 0 minutes, 3 seconds
```

CE1 and CE2 can ping each other.

Take the display on CE1 for example.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=31 ms
  Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=10 ms
  Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=28 ms
--- 100.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/15/31 ms
```

---End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
vsi a2 static
pwsignal ldp
vsi-id 2
peer 3.3.3.9
#
mpls ldp
#
mpls ldp remote-peer 3.3.3.9
remote-ip 3.3.3.9
#
interface GigabitEthernet1/0/0
l2 binding vsi a2
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
```

```
mpls
mpls ldp
#
interface LoopBack1
ip address 1.1.1.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 1.1.1.9 0.0.0.0
network 10.1.1.0 0.0.0.255
#
return
```

- Configuration file of P

```
#
sysname P
#
mpls lsr-id 2.2.2.9
mpls
#
mpls ldp
#
interface GigabitEthernet1/0/0
ip address 10.2.2.2 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
ip address 10.1.1.2 255.255.255.0
mpls
mpls ldp
#
interface LoopBack1
ip address 2.2.2.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 2.2.2.9 0.0.0.0
network 10.1.1.0 0.0.0.255
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
vsi a2 static
pwsignal ldp
vsi-id 2
peer 1.1.1.9
#
mpls ldp
#
mpls ldp remote-peer 1.1.1.9
remote-ip 1.1.1.9
#
interface GigabitEthernet1/0/0
ip address 10.2.2.1 255.255.255.0
mpls
mpls ldp
#
interface GigabitEthernet2/0/0
l2 binding vsi a2
#
```

```
interface LoopBack1
ip address 3.3.3.9 255.255.255.255
#
ospf 1
area 0.0.0.0
network 3.3.3.9 0.0.0.0
network 10.2.2.0 0.0.0.255
#
return
```

- Configuration file of CE2

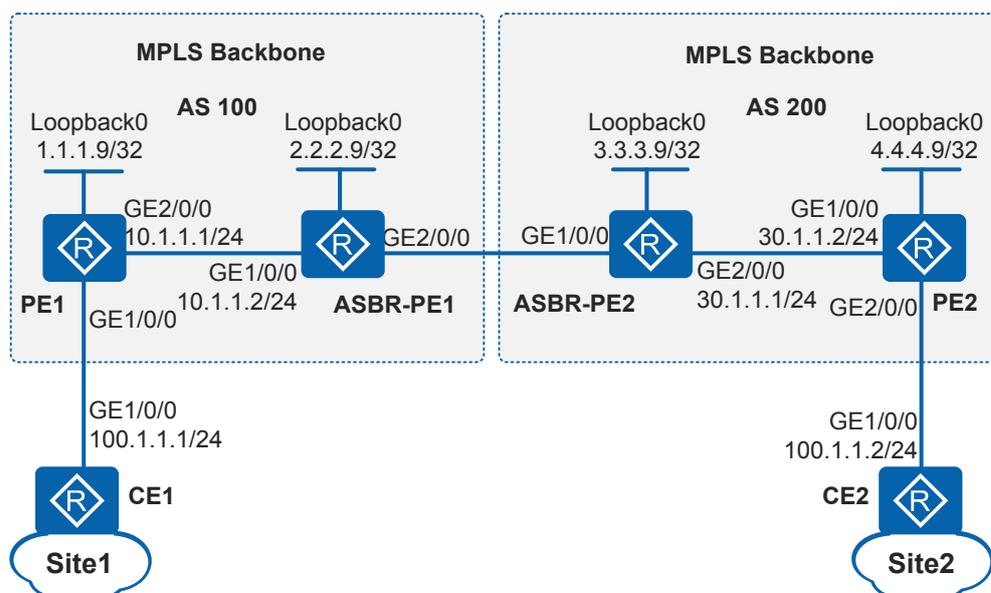
```
#
sysname CE2
#
interface GigabitEthernet1/0/0
ip address 100.1.1.2 255.255.255.0
#
return
```

13.10.2 Example for Configuring Inter-AS Martini VPLS in OptionA Mode

Networking Requirements

As shown in [Figure 13-15](#), on an enterprise network, Site1 connects to PE1 through CE1 and then connects to the VPLS domain of AS 100. Site2 connects to PE2 through CE2 and then connects to the VPLS domain of AS 200. The network environments of the branch sites are stable. AS 100 and AS 200 communicate with each other through ASBR-PE1 and ASBR-PE2. IS-IS is used as the IGP on the MPLS backbone network in an AS. Users at Site1 and Site2 need to communicate at Layer 2 and user information needs to be reserved when Layer 2 packets are transmitted over the backbone network.

Figure 13-15 Networking diagram of configuring inter-AS Martini VPLS in OptionA mode



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure transparent transmission of Layer 2 packets over the backbone network using VPLS to enable users at Site1 and Site2 to communicate at Layer 2 and reserve user information when Layer 2 packets are transmitted over the backbone network.
2. Use Martini VPLS to implement Layer 2 communication between CEs when the network environments of the branch sites are stable.
3. Configure the IGP routing protocol on the backbone network to implement communication between devices within an AS on the public network.
4. Configure basic MPLS functions and LDP on PEs on the backbone network to support VPLS.
5. Establish tunnels for transmitting data between PEs within an AS to prevent data from being known by the public network. Establish dynamic LSPs between ASBR-PEs and PEs in the same AS. If PEs and ASBR-PEs are not directly connected, establish remote LDP sessions.
6. Enable MPLS L2VPN on PEs to implement VPLS.
7. Create a VSI on PEs, specify LDP as the signaling protocol, and bind the VSI to the AC interface in the same AS to implement Martini VPLS.
8. To implement VPLS inter-AS OptionA, configure the peer ASBR as the CE on the ASBR PE, and bind VSIs to peer interfaces.

Procedure

Step 1 Configure IP addresses for interfaces according to [Figure 13-15](#).

Configure CE1. The configuration on PE1, ASBR-PE1, ASBR-PE2, PE2, and CE2 is similar to the configuration on CE1 and is not mentioned here.

```
<Huawei> system-view
[Huawei] sysname CE1
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] ip address 100.1.1.1 255.255.255.0
[CE1-GigabitEthernet1/0/0] quit
```

Step 2 Configure the IGP on the MPLS backbone network.

Configure the IGP on the MPLS backbone network to achieve connectivity between the PEs and ASBR PEs. Note that IS-IS must be enabled on Loopback0.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the PE1, and is not mentioned here.

```
[PE1] isis 1
[PE1-isis-1] network-entity 10.0000.0000.0001.00
[PE1-isis-1] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] isis enable 1
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface loopback 0
[PE1-LoopBack0] isis enable 1
[PE1-LoopBack0] quit
```

After the configuration is complete, the ASBR-PE and PE in the same AS can establish an IS-IS neighbor. Run the **display isis peer** command, and you can see that the IS-IS neighbor is in Up state.

The information displayed on PE1 is used as an example.

```
[PE1] display isis peer

Peer information for ISIS(1)

System Id      Interface      Circuit Id      State HoldTime Type  PRI
-----
0000.0000.0002 GE2/0/0        0000.0000.0002.01 Up    23s    L1 (L1L2) 64
0000.0000.0002 GE2/0/0        0000.0000.0002.01 Up    22s    L2 (L1L2) 64

Total Peer(s): 2
```

ASBR-PEs and PEs in the same AS can Ping each other.

The information displayed on PE1 is used as an example.

```
[PE1] ping 2.2.2.9
PING 2.2.2.9: 56 data bytes, press CTRL_C to break
Reply from 2.2.2.9: bytes=56 Sequence=1 ttl=255 time=180 ms
Reply from 2.2.2.9: bytes=56 Sequence=2 ttl=255 time=90 ms
Reply from 2.2.2.9: bytes=56 Sequence=3 ttl=255 time=60 ms
Reply from 2.2.2.9: bytes=56 Sequence=4 ttl=255 time=60 ms
Reply from 2.2.2.9: bytes=56 Sequence=5 ttl=255 time=100 ms

--- 2.2.2.9 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
round-trip min/avg/max = 60/98/180 ms
```

Step 3 Configure basic MPLS functions and LDP.

Enable basic MPLS functions on the MPLS backbone network. Establish a dynamic LDP LSP between the PE and ASBR PE in the same AS.

Configure PE1. The configuration on ASBR-PE1, ASBR-PE2, and PE2 is similar to the PE1, and is not mentioned here.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] mpls
[PE1-GigabitEthernet2/0/0] mpls ldp
[PE1-GigabitEthernet2/0/0] quit
```

After this step, an LSP is established between the PE and ASBR-PE in the same AS.

Run the **display mpls ldp session** command to view the LDP LSP status.

ASBR-PE1 is used as an example.

```
[ASBR-PE1] display mpls ldp session

LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
A '*' before a session means the session is being deleted.
-----
PeerID          Status      LAM  SsnRole  SsnAge      KASent/Rcv
-----
1.1.1.9:0       Operational DU   Active  0000:00:19  79/79
-----
TOTAL: 1 session(s) Found.
```

Step 4 Configure LDP VPLS and bind VSIs to interfaces.

Configure VSIs on PEs and ASBR PEs respectively and bind the VSIs to the related interfaces.

Configure PE1.

```
[PE1] mpls l2vpn
[PE1-l2vpn] quit
[PE1] vsi a1 static
[PE1-vsi-a1] pwsignal ldp
[PE1-vsi-a1-ldp] vsi-id 2
[PE1-vsi-a1-ldp] peer 2.2.2.9
[PE1-vsi-a1-ldp] quit
[PE1-vsi-a1] quit
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] l2 binding vsi a1
[PE1-GigabitEthernet1/0/0] quit
```

Configure ASBR-PE1.

```
[ASBR-PE1] mpls l2vpn
[ASBR-PE1-l2vpn] quit
[ASBR-PE1] vsi a1 static
[ASBR-PE1-vsi-a1] pwsignal ldp
[ASBR-PE1-vsi-a1-ldp] vsi-id 2
[ASBR-PE1-vsi-a1-ldp] peer 1.1.1.9
[ASBR-PE1-vsi-a1-ldp] quit
[ASBR-PE1-vsi-a1] quit
[ASBR-PE1] interface gigabitethernet 2/0/0
[ASBR-PE1-GigabitEthernet2/0/0] l2 binding vsi a1
[ASBR-PE1-GigabitEthernet2/0/0] quit
```

Configure ASBR-PE2.

```
[ASBR-PE2] mpls l2vpn
[ASBR-PE2-l2vpn] quit
[ASBR-PE2] vsi a1 static
[ASBR-PE2-vsi-a1] pwsignal ldp
[ASBR-PE2-vsi-a1-ldp] vsi-id 3
[ASBR-PE2-vsi-a1-ldp] peer 4.4.4.9
[ASBR-PE2-vsi-a1-ldp] quit
[ASBR-PE2-vsi-a1] quit
[ASBR-PE2] interface gigabitethernet 1/0/0
[ASBR-PE2-GigabitEthernet1/0/0] l2 binding vsi a1
[ASBR-PE2-GigabitEthernet1/0/0] quit
```

Configure PE2.

```
[PE2] mpls l2vpn
[PE2-l2vpn] quit
[PE2] vsi a1 static
[PE2-vsi-a1] pwsignal ldp
[PE2-vsi-a1-ldp] vsi-id 3
[PE2-vsi-a1-ldp] peer 3.3.3.9
[PE2-vsi-a1-ldp] quit
[PE2-vsi-a1] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] l2 binding vsi a1
[PE2-GigabitEthernet2/0/0] quit
```

Step 5 Verify the configuration.

After the preceding configurations are complete, run the **display vsi name a1 verbose** command on PE1, and you can see that the VSI named **a1** has established a PW to PE2, and the status of the VSI is **Up**.

Take the display on PE1 and ASBR-PE2 for example.

```
[PE1] display vsi name a1 verbose
```

```
***VSI Name          : a1
Administrator VSI    : no
Isolate Spoken       : disable
VSI Index            : 0
PW Signaling         : ldp
Member Discovery Style : static
PW MAC Learn Style   : unqualify
Encapsulation Type   : vlan
MTU                  : 1500
Diffserv Mode        : uniform
Service Class        : --
Color                : --
DomainId             : 255
Domain Name          :
Ignore AcState       : disable
P2P VSI              : disable
Create Time          : 0 days, 3 hours, 30 minutes, 31 seconds
VSI State            : up

VSI ID               : 2
*Peer Router ID      : 2.2.2.9
Negotiation-vc-id    : 2
primary or secondary : primary
ignore-standby-state : no
VC Label             : 23552
Peer Type            : dynamic
Session              : up
Tunnel ID            :
Broadcast Tunnel ID  : 0x0
Broad BackupTunnel ID : 0x0
CKey                 : 6
NKey                 : 5
Stp Enable           : 0
PwIndex              : 0
Control Word         : disable
BFD for PW           : unavailable

Interface Name       : GigabitEthernet1/0/0
State                : up
Access Port          : false
Last Up Time         : 2017/07/02 15:41:59
Total Up Time        : 0 days, 0 hours, 1 minutes, 2 seconds

**PW Information:

*Peer Ip Address     : 2.2.2.9
PW State             : up
Local VC Label       : 23552
Remote VC Label      : 23552
Remote Control Word  : disable
PW Type              : label
Local VCCV           : alert lsp-ping
Remote VCCV          : alert lsp-ping
Tunnel ID            : 0x20020
Broadcast Tunnel ID  : 0x20020
Broad BackupTunnel ID : 0x0
Ckey                 : 0x6
Nkey                 : 0x5
Main PW Token        : 0x20020
Slave PW Token       : 0x0
Tnl Type             : LSP
OutInterface         : GigabitEthernet2/0/0
Backup OutInterface  :
Stp Enable           : 0
PW Last Up Time      : 2017/07/02 15:41:59
PW Total Up Time     : 0 days, 0 hours, 1 minutes, 3 seconds
```

CE1 and CE2 can ping each other.

Take the display on CE1 for example.

```
[CE1] ping 100.1.1.2
PING 100.1.1.2: 56 data bytes, press CTRL_C to break
  Reply from 100.1.1.2: bytes=56 Sequence=1 ttl=255 time=172 ms
  Reply from 100.1.1.2: bytes=56 Sequence=2 ttl=255 time=156 ms
  Reply from 100.1.1.2: bytes=56 Sequence=3 ttl=255 time=156 ms
  Reply from 100.1.1.2: bytes=56 Sequence=4 ttl=255 time=156 ms
  Reply from 100.1.1.2: bytes=56 Sequence=5 ttl=255 time=156 ms

--- 100.1.1.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 156/159/172 ms
```

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
ip address 100.1.1.1 255.255.255.0
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 1.1.1.9
mpls
#
mpls l2vpn
#
vsi a1 static
pwsignal ldp
vsi-id 2
peer 2.2.2.9
#
mpls ldp
#
isis 1
network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
l2 binding vsi a1
#
interface GigabitEthernet2/0/0
ip address 10.1.1.1 255.255.255.0
isis enable 1
mpls
mpls ldp
#
interface LoopBack0
ip address 1.1.1.9 255.255.255.255
isis enable 1
#
return
```

- Configuration file of ASBR-PE1

```
#
sysname ASBR-PE1
#
mpls lsr-id 2.2.2.9
mpls
#
```

```
mpls l2vpn
#
vsi a1 static
 pwsignal ldp
  vsi-id 2
  peer 1.1.1.9
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 l2 binding vsi a1
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
 isis enable 1
#
return
```

● Configuration file of ASBR-PE2

```
#
 sysname ASBR-PE2
#
mpls lsr-id 3.3.3.9
mpls
#
mpls l2vpn
#
vsi a1 static
 pwsignal ldp
  vsi-id 3
  peer 4.4.4.9
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 l2 binding vsi a1
#
interface GigabitEthernet2/0/0
 ip address 30.1.1.1 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
 isis enable 1
#
return
```

● Configuration file of PE2

```
#
 sysname PE2
#
mpls lsr-id 4.4.4.9
mpls
#
mpls l2vpn
#
```

```
vsi a1 static
 pwsignal ldp
 vsi-id 3
 peer 3.3.3.9
#
mpls ldp
#
isis 1
 network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
 ip address 30.1.1.2 255.255.255.0
 isis enable 1
 mpls
 mpls ldp
#
interface GigabitEthernet2/0/0
 l2 binding vsi a1
#
interface LoopBack0
 ip address 4.4.4.9 255.255.255.255
 isis enable 1
#
return
```

- Configuration file of CE2

```
#
 sysname CE2
#
interface GigabitEthernet1/0/0
 ip address 100.1.1.2 255.255.255.0
#
return
```

13.11 Troubleshooting VPLS

This section describes common faults caused by incorrect VPLS configurations and provides the troubleshooting procedure.

13.11.1 VSI Cannot Go Up in Martini VPLS Mode

Fault Symptom

After Martini VPLS is configured, the VSI cannot go Up.

Procedure

Step 1 Run the **display vsi name** *vsi-name* command to check whether the encapsulation types on both ends are the same.

- If the encapsulation types on both ends are different, run the **encapsulation** { **ethernet** | **vlan** } command in the VSI view to change the encapsulation type on one end to ensure that the two ends use the same encapsulation type.
- If the encapsulation types on both ends are the same, go to step 2.

NOTE

A VSI can be Up only when encapsulation types configured on both ends are the same.

Step 2 Run the **display vsi name** *vsi-name* command to check whether MTUs of the two ends are the same.

- If MTUs of the two ends are different, run the **mtu mtu-value** command in the VSI view to change the MTU on one end to ensure that the two ends use the same MTU.
- If MTUs of the two ends are the same, go to step 3.

 **NOTE**

A VSI can be Up only when MTUs configured for the two ends are the same.

Step 3 Run the **display vsi name vsi-name verbose** command to check whether VSI IDs or negotiation IDs on both ends are the same.

- If VSI IDs or negotiation IDs on the two ends are different, run the **vsi-id vsi-id** command in the VSI-LDP view to change the VSI ID on one end, or run the **peer peer-address [negotiation-vc-id vc-id]** command to change the negotiation ID on one end to ensure that VSI IDs or negotiation IDs on the two ends are the same.
- If the VSI IDs or negotiation IDs on both ends are the same, go to step 4.

Step 4 Run the **display vsi name vsi-name verbose** command to check whether the LDP session is Up.

- If the LDP session is Down, see "LDP Session Goes Down" to make the LDP session go Up.
- If the LDP session is Up, go to step 5.

 **NOTE**

The two ends can perform L2VPN negotiation only after the LDP session is Up.

Step 5 Run the **display vsi name vsi-name verbose** command to check whether the AC interfaces on both ends are Up.

If the AC interfaces on the two ends are Down, see "Physical Interconnection&Interface Type" to make the AC interfaces go Up.

----End

13.12 References for VPLS

This section lists references for VPLS.

The following table lists the references for VPLS.

Document No.	Description
RFC 4448	Encapsulation Methods for Transport of Ethernet over MPLS Networks
RFC 4762	Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling

14 VXLAN Configuration

About This Chapter

This document describes VXLAN features on the device and provides principles, configuration procedures and configuration examples.

[14.1 Overview of VXLANs](#)

[14.2 Understanding VXLANs](#)

[14.3 Application Scenario](#)

[14.4 Licensing Requirements and Limitations for VXLAN](#)

This section describes VXLAN configuration notes.

[14.5 Configuring VXLAN \(in Static Mode\)](#)

[14.6 Configuring VXLAN \(in BGP EVPN Mode\)](#)

[14.7 Configuration Examples for VXLANs](#)

[14.8 References for VXLANs](#)

[14.9 Further Reading](#)

14.1 Overview of VXLANs

Definition

As defined by RFC 7348, Virtual eXtensible Local Area Network (VXLAN) is a Network Virtualization over Layer 3 (NVO3) technology that uses the MAC in User Datagram Protocol (MAC-in-UDP) mode to encapsulate packets.

Background

Cloud computing has become the new trend in enterprise IT construction with its features such as high system utilization, low manpower and management costs, flexibility, and strong scalability. As a core technology of cloud computing, server virtualization has a wide range of applications.

 **NOTE**

For detailed description about server virtualization, see [14.9.1 Server Virtualization](#).

The wide application of server virtualization technology greatly increases computing density in a data center. In addition, VMs need to freely migrate on the network to meet service change requirements. These bring challenges to traditional data center networks of the Layer 2 + Layer 3 architecture.

- VM scale limited by network devices' table entry capacities

On a traditional Layer 2 network, data packets are forwarded at Layer 2 based on the MAC address table. Server virtualization leads to an exponential growth of the number of VMs and the number of MAC addresses of the VM network interface cards (NICs). However, the MAC address table size of a Layer 2 device at the access side is incapable to meet this change.

- Insufficient network isolation capabilities

While VLAN is the most commonly used network isolation technology, it has its own limitations. The VLAN field in packets is only 12 bits long, which means that at most 4096 VLANs can be used on a network. In public cloud or other cloud computing scenarios involving tens of thousands or even more tenants, VLAN technology can no longer meet network isolation requirements.

 **NOTE**

A tenant is a complete collection of logical resources deployed on a data center network, including network resources such as VLANs and IP address pools, as well as computing resources such as physical servers and virtual machines (VMs). Each tenant has its own tenant administrator to orchestrate and deploy network services.

- Limited VM migration scope

VMs on a data center network frequently migrate due to server resource issues, such as high CPU usage and insufficient memory.

VM migration is a process in which a VM moves from one physical server to another. To ensure uninterrupted services during VM migration, the IP and MAC addresses of VMs must remain unchanged. To meet this requirement, server migration must occur in a Layer 2 network. However, a traditional Layer 2 network limits the VM migration scope.

VXLAN addresses the preceding problems:

- For VM scale limitations imposed by table entry capacities

VXLAN encapsulates original data packets sent from VMs in the same region into UDP packets, with the IP and MAC addresses used on the physical network in outer headers. The network is only aware of the encapsulated parameters. This greatly reduces the number of MAC address entries required on large Layer 2 networks.

- For limited network isolation capabilities

VXLAN uses a VXLAN Network Identifier (VNI) field similar to the VLAN ID field to identify users. The VNI field has 24 bits and can identify up to 16M VXLAN segments, effectively isolating massive tenants in cloud computing scenarios.

- For limited VM migration scope

VXLAN encapsulates original packets sent by VMs over a VXLAN tunnel. VMs at two ends of a VXLAN tunnel do not need to know the physical architecture of the transmission network. In this way, VMs using IP addresses in the same network segment are in a Layer 2 domain logically, even if they are on different physical Layer 2 networks. VXLAN technology constructs a virtual large Layer 2 network over a Layer 3 network, so that VMs are on the same large Layer 2 network so long as there are

reachable routes between them. The virtual large Layer 2 network enlarges the VM migration scope.

 **NOTE**

For detailed description about large Layer 2 network, see [14.9.2 Large Layer 2 Network](#).

Purpose

VXLAN is developed to implement server virtualization and free VM migration on data center networks. As a VPN technology, VXLAN can also be used on campus networks to provide Layer 2 interconnection between dispersed physical sites and Layer 3 interconnection between sites.

Currently, related devices and multiple Layer 2 and Layer 3 network technologies need to be deployed on campus networks to implement Layer 2 and Layer 3 interconnection between tenant sites. Overlay-based VXLAN technology establishes Layer 2 virtual networks between any networks with reachable routes to implement Layer 2 interconnection. Layer 3 interconnection is implemented between sites by VXLAN Layer 3 gateway at the same time. In all, VXLAN realizes faster and more flexible site interconnection.

14.2 Understanding VXLANs

14.2.1 VXLAN Network Architecture

VXLAN is an NVO3 network virtualization technology that encapsulates data packets sent from original hosts into UDP packets and encapsulates IP and MAC addresses used on the physical network in outer headers before sending the packets over an IP network. The virtual tunnel endpoint (VTEP) then decapsulates the packets and sends the packets to the destination host.

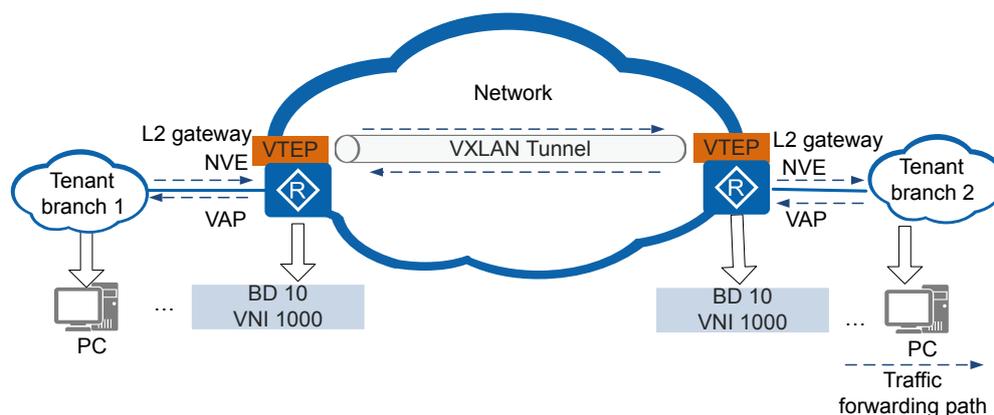
By leveraging VXLAN, a virtual network can accommodate a large number of tenants. Tenants can plan their own virtual networks without being limited by physical network IP addresses or broadcast domains. This technology significantly simplifies network management.

Similar to a traditional VLAN, a VXLAN also allows for intra- and inter-VXLAN communication.

Intra-VXLAN Communication

VXLAN technology constructs a virtual Layer 2 network over a Layer 3 network, implementing Layer 2 communication between hosts. [Figure 14-1](#) shows intra-VXLAN communication.

Figure 14-1 Intra-VXLAN Communication



Involved concepts

- **VXLAN Network Identifier (VNI)**
 A VNI is similar to a VLAN ID on a traditional network, and it identifies a VXLAN segment. Tenants on different VXLAN segments cannot communicate at Layer 2. One tenant may have one or more VNIs. A VNI consists of 24 bits and supports up to 16M tenants.
- **Broadcast Domain (BD)**
 Similar to VLANs divided on a traditional network, BD is used for broadcast domain division on a VXLAN.
 On a VXLAN, to allow Layer 2 communication between hosts in a BD, VNIs and BDs are mapped in 1:1 mode.
- **VXLAN Tunnel Endpoints (VTEP)**
 A VTEP encapsulates and decapsulates VXLAN packets.
 The source and destination IP addresses in a VXLAN packet are the IP addresses of the local and remote VTEPs, respectively. A pair of VTEP addresses defines one VXLAN tunnel. A source VTEP encapsulates packets and selects a tunnel to forward them. The corresponding destination VTEP decapsulates the received packets.
- **Virtual Access Point (VAP)**
 A VAP is a VXLAN service access point used for service access based on VLANs or packet encapsulation modes. For more information, see [14.2.3.1 Packet Identification](#):
 - Service access based on VLANs: The 1:1 or N:1 mapping between VLANs and BDs is configured on VTEPs. When a VTEP receives a service packet, it forwards the packet in a BD based on the mapping between VLANs and BDs.
 - Service access based on packet encapsulation modes: Layer 2 sub-interfaces are created on a downlink physical interface of a VTEP, and different encapsulation modes are configured for these sub-interfaces to enable different interfaces to receive different data packets. The 1:1 mapping between Layer 2 sub-interfaces and BDs is also defined. Then service packets are sent to specific Layer 2 sub-interfaces after reaching the VTEP. That is, packets are forwarded in a BD based on the mapping between Layer 2 sub-interfaces and BDs.
- **Network Virtualization Edge (NVE)**
 An NVE is a network entity used to implement network virtualization functions. After packets are encapsulated and decapsulated through NVEs, a Layer 2 VXLAN can be

established between NVEs over the basic Layer 3 network. In the preceding figure, routers serve as NVEs.

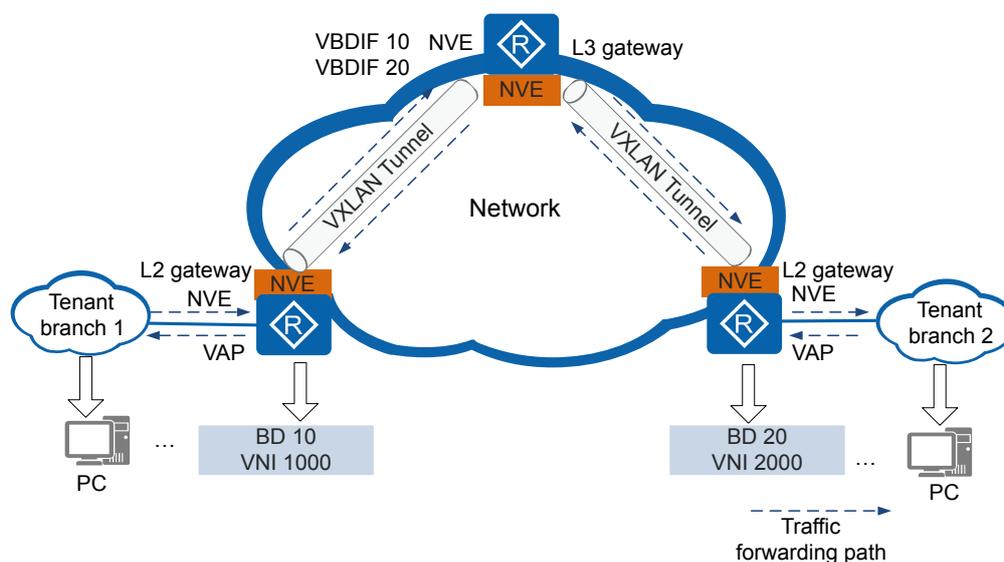
- Layer 2 gateway

Similar to a Layer 2 access device on a traditional network, it allows tenant access to VXLANs and intra-subnet VXLAN communication in the same network segment.

Inter-VXLAN Communication

Hosts in different BDs cannot directly communicate at Layer 2. VXLAN Layer 3 gateways need to be configured to implement Layer 3 communication between hosts. **Figure 14-2** shows inter-VLAN communication.

Figure 14-2 Inter-VXLAN Communication



Involved concepts

- Layer 3 gateway

On a traditional network, users in different VLANs cannot directly communicate at Layer 2. Layer 2 communication is also not allowed between VXLANs identified by different VNIs or between VXLANs and non-VXLANs. To address these problems, the VXLAN Layer 3 gateway is introduced to enable data transmission between VXLANs or between VXLANs and non-VXLANs.

The VXLAN Layer 3 gateway is used for cross-subnet communication on the VXLAN and external network access.

- VBDIF interface

On a traditional network, VLANIF interfaces are used to enable communication between different BDs. Similarly, VBDIF interfaces are introduced in a VXLAN to implement such function.

The VBDIF interface is configured on the VXLAN Layer 3 gateway and is a Layer 3 logical interface based on BDs. After IP addresses are configured for VBDIF interfaces, VXLANs on different network segments, VXLANs and non-VXLANs, and Layer 2 and Layer 3 networks can communicate with each other.

Comparison Between VXLAN and VLAN

The following table lists the differences between VXLAN and VLAN.

Table 14-1 Comparison between VXLAN and VLAN

Item	VLAN	VXLAN
Concept	Virtual Local Area Network.	Virtual Extensible Local Area Network.
Implementation Method	A physical LAN is divided into multiple BDs logically to limit the network to a small geographic range.	Layer 2 virtual networks are established between networks with reachable routes. Such networks are not subject to geographical restrictions and can deliver a large-scale scalability.
Supported capacity	VLAN is the most commonly used network isolation technology. The VLAN field in packets is only 12 bits in length, which means that only a maximum of 4096 VLANs can be used on a network. In public cloud or other cloud computing scenarios involving tens of thousands or even more tenants, VLAN technology can no longer meet network isolation requirements.	VXLAN is a new network isolation technology defined in IETF RFC 7348. It has a 24-bit segment identifier (VNI) and can isolate up to 16M (about 16 million) tenants. This technology effectively enables isolation of mass tenants in cloud computing.
Network division mode	VLAN IDs are used to divide broadcast domains. Hosts within a BD can communicate at Layer 2.	BDs are used to divide broadcast domains. Hosts within a BD can communicate at Layer 2.
Encapsulation mode	A VLAN tag is added to packets.	During VXLAN encapsulation, a VXLAN header, UDP header, IP header, and outer MAC header are added in sequence to an original packet. For details, see 14.2.2 Packet Encapsulation Format .
Network communication mode	Inter-VLAN communication is implemented by VLANIF interfaces. As Layer 3 logical interfaces, VLANIF interfaces enable Layer 3 communication between VLANs.	Communication between VLANs or between VXLANs and non-VXLANs is implemented by VBDIF interfaces. VBDIF interfaces are configured on VXLAN Layer 3 gateways and are Layer 3 logical interfaces based on BDs.

Item	VLAN	VXLAN
Benefits	<p>Limits broadcast domains: A broadcast domain is limited in a VLAN, which saves bandwidth and improves network processing capabilities.</p> <p>Enhances LAN security: Packets from different VLANs are separately transmitted. Hosts in a VLAN cannot directly communicate with hosts in another VLAN.</p>	<p>Location-independent capability: Services can be deployed flexibly at any location, solving network expansion issues related to server virtualization.</p> <p>Flexible network deployment: VXLANs are constructed over the traditional network. They are easy to deploy and highly scalable while preventing broadcast storms on a large Layer 2 network.</p> <p>Cloud service adaptation: A VXLAN is able to isolate ten millions of tenants and support large-scale deployment of cloud services.</p> <p>Technical advantage: VXLAN uses MAC-in-UDP encapsulation. Such encapsulation mode does not rely on MAC addresses of hosts, reducing the number of MAC address entries required on a large Layer 2 network.</p>

14.2.2 Packet Encapsulation Format

During VXLAN encapsulation, a VXLAN header, UDP header, IP header, and Ethernet header are added in sequence to an original packet.

Figure 14-3 shows the packet encapsulation format.

Figure 14-3 VXLAN packet format

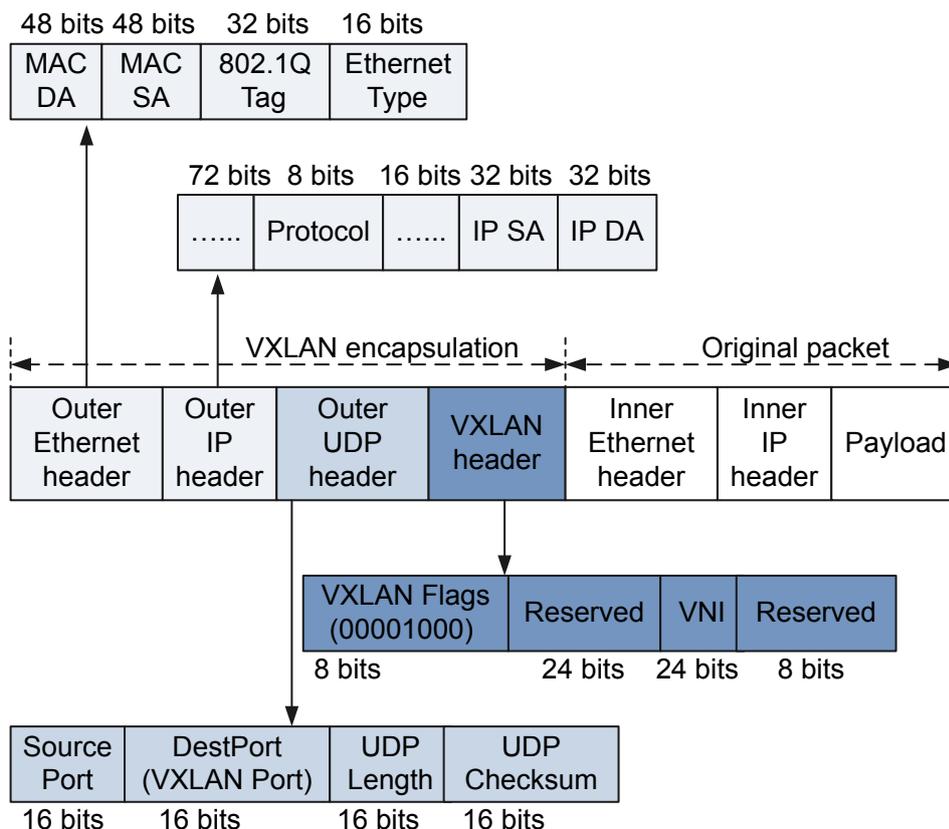


Table 14-2 describes headers added to an original packet during VXLAN encapsulation.

Table 14-2 Description of headers added to an original packet

Field	Description
VXLAN header	<ul style="list-style-type: none"> ● VXLAN Flags: specifies flags (8 bits). The value is 00001000. ● VNI: specifies an identifier (24 bits) used to identify a VXLAN segment, with up to 16M tenants. Users in different VXLAN segments cannot directly communicate at Layer 2. ● Reserved: The two reserved fields (24 and 8 bits respectively) are set to 0.
Outer UDP header	<ul style="list-style-type: none"> ● DestPort: specifies the destination UDP port number. The value is 4789. ● Source Port: specifies the source port number. It is the hash value calculated using parameters in the inner Ethernet frame header.

Field	Description
Outer IP header	<ul style="list-style-type: none">● IP SA: specifies the source IP address, which is the IP address of the source VTEP.● IP DA: specifies the destination IP address, which is the IP address of the destination VTEP.
Outer Ethernet header	<ul style="list-style-type: none">● MAC DA: specifies the destination MAC address, which is the MAC address of the next-hop device on the route to the destination VTEP.● MAC SA: specifies the source MAC address, which is the MAC address of the source VTEP that sends the packet.● 802.1Q Tag (optional): specifies the VLAN tag in the packet.● Ethernet Type: specifies the type of the Ethernet frame. The value of this field is 0x0800 when an IP packet is transmitted.

14.2.3 VXLAN Implementation

VXLAN needs to be deployed on a downlink interface to provide access services and an uplink interface to establish a VXLAN tunnel. After VXLAN is deployed on both interfaces, packets can be forwarded on the VXLAN network. VXLAN implementation is described in three steps: Packet Identification, VXLAN tunnel establishment, and Packet Forwarding.

14.2.3.1 Packet Identification

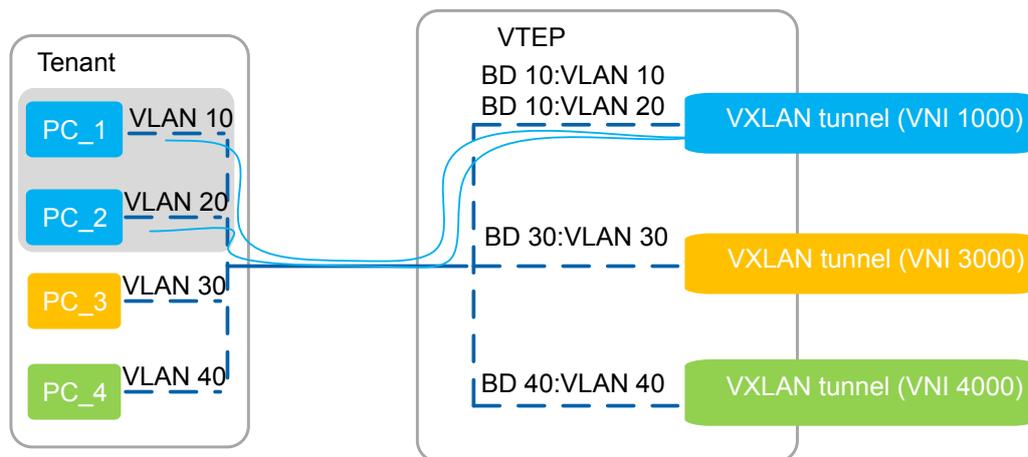
On a VXLAN network, VNIs are mapped to BDs in 1:1 mode. After a packet reaches a VTEP, the VTEP can identify the BD to which the packet belongs, then select a correct tunnel to forward the packet. Two methods are available for a VTEP to identify the VXLAN to which a packet belongs.

VXLAN Identification by VLAN

The 1:1 or N:1 mapping between VLANs and BDs is configured on VTEPs based on network planning. After a VTEP receives a service packet, it correctly selects a VXLAN tunnel to forward the packet based on the mapping between VLANs and BDs and the mapping between BDs and VNIs.

In [Figure 14-4](#), VLAN 10 and VLAN 20 belong to BD 10. The mapping between VLANs 10 and 20 and BD 10, as well as the mapping between BD 10 and VNI 1000 are configured on the VTEP. After the VTEP receives a packet from PC_1 or PC_2, the VTEP forwards the packet over the VXLAN tunnel for VNI 1000.

Figure 14-4 VXLAN identification by VLAN



VXLAN Identification by Encapsulation Mode

An encapsulation mode defines packet processing based on whether a packet contains VLAN tags. To implement VXLAN identification by encapsulation mode, Layer 2 sub-interfaces need to be configured on a downlink physical interface of a VTEP, and different encapsulation modes need to be configured for these sub-interfaces. The 1:1 mapping between Layer 2 sub-interfaces and BDs should also be defined. Then service packets are sent to specific Layer 2 sub-interfaces after reaching the VTEP. The VTEP selects a correct VXLAN tunnel to forward packets based on the mapping between Layer 2 sub-interfaces and BDs and the mapping between BDs and VNIs.

Table 14-3 lists the four encapsulation modes and packet processing methods.

Table 14-3 Packet processing in different encapsulation modes

Encapsulation Mode	Allowed Packet Type	Packet Encapsulation	Packet Decapsulation
dot1q	With specified VLAN tag	Removes the VLAN tag from original packets.	<ul style="list-style-type: none"> ● Removes the VLAN tag and adds a specified VLAN tag before forwarding them, if the inner packets contain a VLAN tag. ● Adds a specified VLAN tag to the original inner packets before forwarding them, if the inner packets do not contain a VLAN tag.
untag	Without VLAN tags	Does not perform any operation on the original packets.	<ul style="list-style-type: none"> ● Removes the outer VLAN tag before forwarding them, if the inner packets contain a VLAN tag. ● Directly forwards the packets if the inner packets do not contain VLAN tags.
default	All packets regardless of whether they contain VLAN tags	Does not perform any operation on the original packets.	Does not perform any operation on the original packets.

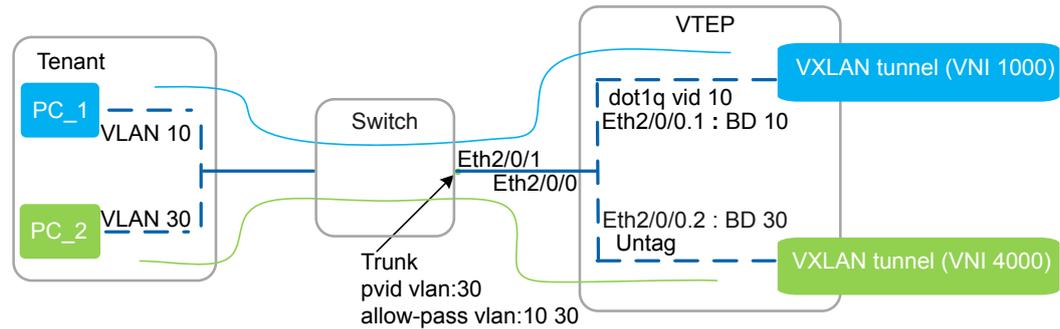
Encapsulation Mode	Allowed Packet Type	Packet Encapsulation	Packet Decapsulation
qinq	With specified double VLAN tags	Removes all the VLAN tags from original packets.	<ul style="list-style-type: none"> ● Removes all the VLAN tags and adds specified double VLAN tags before forwarding them, if the inner packets contain VLAN tags. ● Adds specified double VLAN tags to the original inner packets before forwarding them, if the inner packets do not contain VLAN tags.

In **Figure 14-5**, the physical interface Eth2/0/0 on the VTEP has two Layer 2 sub-interfaces, which are configured with different encapsulation modes and associated with different BDs. PC_1 and PC_2 belong to VLAN 10 and VLAN 30, respectively. An uplink interface on the Layer 2 switch connecting to the VTEP is configured as a trunk interface with the PVID 30 and is configured to allow packets from VLANs 10 and 30 to pass through. When a packet from PC_1 reaches this interface, the interface transparently transmits the packet to the VTEP because the VLAN ID of the packet is different from the default VLAN ID of the interface. When a packet from PC_2 reaches this interface, the interface removes the VLAN tag 30 from the packet before forwarding it to the VTEP because the VLAN ID of the packet is the same as the default VLAN ID of the interface. As a result, when the packets reach Eth2/0/0 on the VTEP, the packet from PC_1 contains VLAN tag 10, while the packet from PC_2 does not contain a VLAN tag. To distinguish the two types of packets, Layer 2 sub-interfaces of the **dot1q** and **untag** types need to be configured on Eth2/0/0:

- The encapsulation mode of the Layer 2 sub-interface Eth2/0/0.1 is **dot1q**, allowing packets with VLAN tag 10 to enter the VXLAN tunnel.
- The encapsulation mode of the Layer 2 sub-interface Eth2/0/0.2 is **untag**, allowing packets without a VLAN tag to enter the VXLAN tunnel.

After packets from PC_1 or PC_2 reach the VTEP, the VTEP sends the packets to different Layer 2 sub-interfaces based VLAN tags in the packets. Then, the VTEP chooses a correct VXLAN tunnel to forward the packets based on the mapping between sub-interface and BD, as well as the mapping between BD and VNI.

Figure 14-5 VXLAN identification by encapsulation mode



14.2.3.2 Tunnel Establishment

A VXLAN tunnel is specified by a pair of VXLAN Tunnel Endpoint (VTEP) IP addresses. A static VXLAN tunnel can be created after the VXLAN Network Identifiers (VNIs) and IP addresses are configured for the source and destination VTEPs, and there is a reachable route between the two VTEP IP addresses.

A VXLAN tunnel is specified by a pair of VTEP IP addresses. After two VTEPs obtain the IP addresses from each other, a VXLAN tunnel is established so long as there is a reachable route between the two IP addresses.

- A static VXLAN tunnel can be created by manually specifying the VNI of the tunnel source and destination as well as the IP addresses of source and destination VTEPs. This leads to heavy configuration workload and poor flexibility. Therefore, static VXLAN tunnels do not apply on a large-scale network.
- BGP EVPN can be used to dynamically establish VXLAN tunnels by establishing a BGP EVPN peer relationship between two VTEPs and using BGP EVPN routes to transmit VNIs and VTEP IP addresses between the peer. In this mode, the EVPN protocol automatically discovers VTEPs and dynamically creates VXLAN tunnels. With high flexibility, this mode applies on a large-scale VXLAN network.

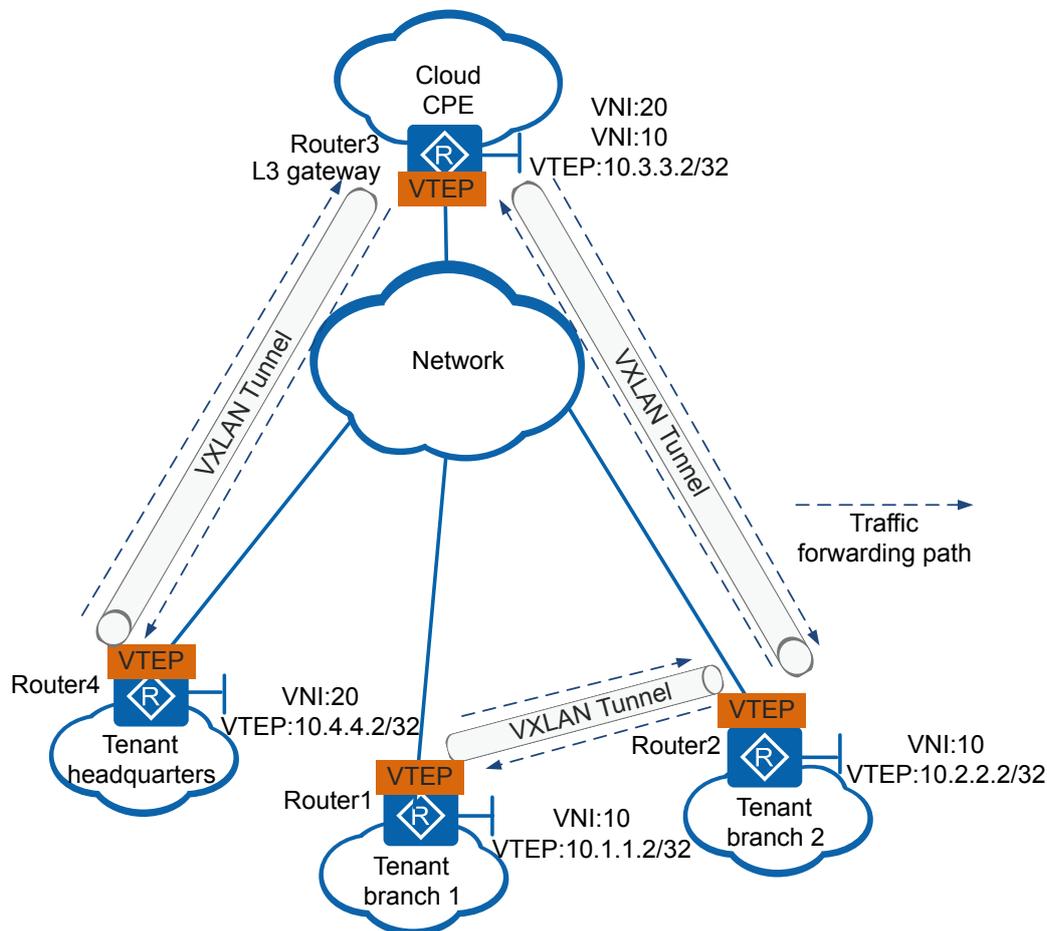
In **Figure 14-6**, Router1, Router2, Router3, and Router4 are enterprise egress gateways. VXLAN tunnels need to be established to enable communication between tenant branches and headquarters, as well as between tenant branches.

In this situation, VXLAN tunnels are established in the following two ways:

- Tenant branches 1 and 2 are in the same subnet and VNI. Tenants in a VNI are in the same logical Layer 2 network, so they can directly communicate at Layer 2 over a VXLAN tunnel. To enable communication between tenant branches 1 and 2, the VNI and VTEP IP addresses need to be manually configured on Router1 and Router2. So long as Router1 and Router2 have a reachable route to the peer VTEP, a VXLAN tunnel can be established between them.
- The tenant branch 2 and headquarters belong to different subnets and VNIs. They cannot directly communicate over a VXLAN tunnel, and a VXLAN Layer 3 gateway needs to be configured. To enable communication between the tenant branch 2 and headquarters, the static VNIs and VTEP IP addresses need to be manually configured for Router2 and Router3, as well as Router4 and Router3. So long as Router2 and Router3 have a reachable route to the peer VTEP, a VXLAN tunnel can be established between them.

Similarly, so long as Router4 and Router3 have a reachable route to the peer VTEP, a VXLAN tunnel can be established between them.

Figure 14-6 VXLAN networking



14.2.3.3 Packet Forwarding

VXLAN encapsulates Layer 2 network packets to transmit them over traditional Layer 3 networks by constructing a large Layer 2 network among the Layer 3 networks.

MAC Address Learning

On a VXLAN, dynamic MAC address learning is supported to enable user communication. **Figure 14-7** and **Figure 14-7** describe the MAC address learning process during communication between hosts on the same subnet. For the first time, PC_1 does not know the MAC address of PC_2. PC_1 then sends an ARP broadcast packet to request the MAC address of PC_2.

Figure 14-7 Forwarding process of ARP request packets

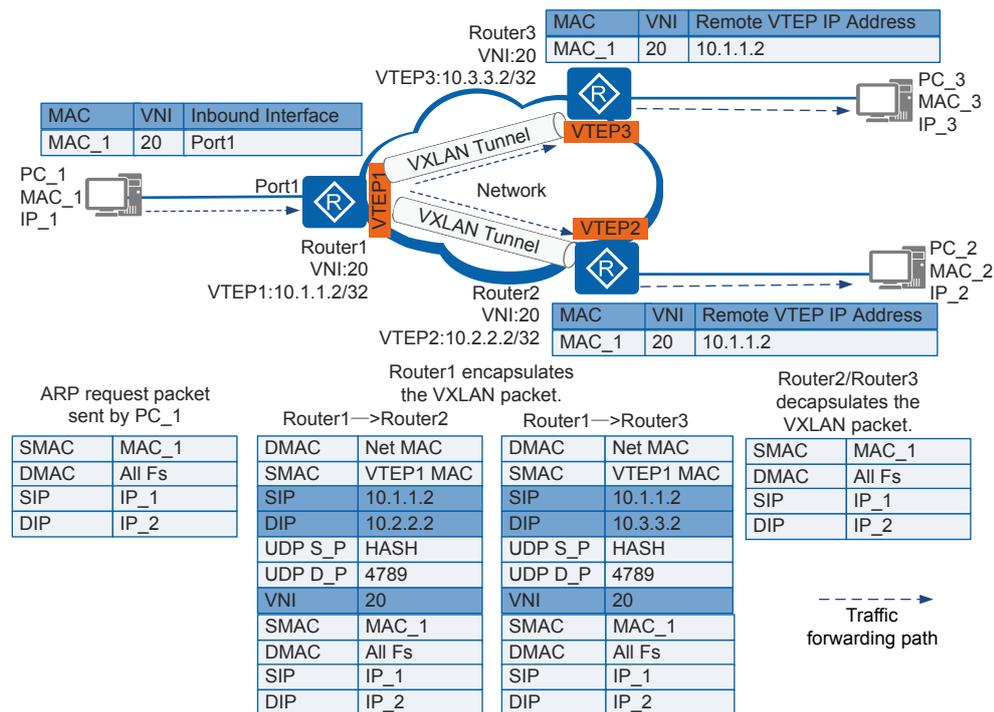


Figure 14-7 shows the forwarding process of ARP request packets.

1. PC_1 sends an ARP broadcast packet with the source MAC address MAC_1, all-F destination MAC address, source IP address IP_1, and destination IP address IP_2 to request the MAC address of PC_2.
2. Router1 receives an ARP request packet sent by PC_1, and chooses a VXLAN tunnel based on the interface that receives the packet. Due to the 1:1 mapping between interfaces and BDs, VTEP1 obtains the VNI to which the packet belongs after determining the BD of the packet. VTEP1 then learns the mapping between MAC_1, VNI, and interface receiving the packet, and saves the mapping to the local MAC address table. After obtaining the ingress replication list for the VNI based on the corresponding BD, VTEP1 replicates packets and performs VXLAN tunnel encapsulation. During packet encapsulation, VTEP1 adds the IP addresses of the source VTEP (VTEP1) and destination VTEPs (VTEP2 and VTEP3), the MAC addresses of VTEP1 and the next-hop device on the route to the destination VTEP as the outer source IP, destination IP, source MAC, and destination MAC addresses respectively. The encapsulated packet is transmitted based on the outer MAC and IP addresses until it reaches the destination VTEP (VTEP2/VTEP3).
3. VTEP2/VTEP3 on Router2/Router3 receives the encapsulated VXLAN packet and decapsulates the packet to obtain the original packet sent by PC_1. VTEP2 and VTEP3 also learn the mapping between PC_1's MAC address, VNI, and IP address of the remote VTEP, and saves the mapping in the local MAC address table. VTEP 2 and VTEP 3 then process the packet based on the interface configuration and broadcast the packet in the corresponding Layer 2 domain.
4. After receiving the ARP request packet, PC_2 and PC_3 check whether the destination IP address in the packet is its own IP address. If so, PC_2 or PC_3 sends an ARP reply packet. If not, PC_2 or PC_3 discards the packet.

Figure 14-8 Forwarding process of ARP reply packets

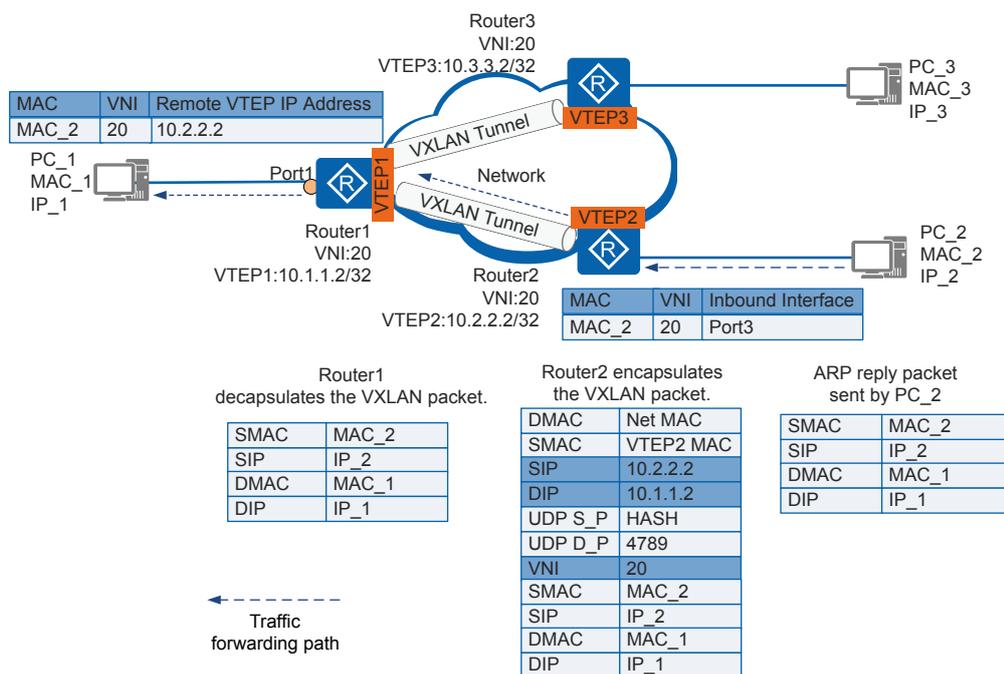


Figure 14-8 shows the forwarding process of ARP reply packets.

1. PC_2 sends a unicast ARP reply packet after learning the MAC address of PC_1. The source MAC, destination MAC, source IP, and destination IP addresses in the packet are MAC_2, MAC_1, IP_2, and IP_1, respectively.
2. After receiving the ARP reply packet, VTEP2 checks the VNI to which the packet belongs. At the same time, VTEP2 learns the mapping between MAC_2, VNI, and interface receiving the packet, and saves the mapping to the local MAC address table. VTEP2 then encapsulates the packet. During packet encapsulation, VTEP2 adds the IP addresses of the source VTEP (VTEP2) and destination VTEP (VTEP1), the MAC addresses of VTEP2 and the next-hop device on the route to VTEP1 as the outer source IP, destination IP, source MAC, and destination MAC addresses respectively. The encapsulated packet is transmitted based on the outer MAC and IP addresses until it reaches the destination VTEP (VTEP1).
3. VTEP1 receives the encapsulated VXLAN packet and decapsulates the packet to obtain the original packet sent by PC_2. VTEP1 also learns the mapping between PC_2's MAC address, VNI, and IP address of the remote VTEP (VTEP2), and saves the mapping in the local MAC address table. VTEP1 then sends the decapsulated packet to PC_1. After learning the MAC address of each other, PC_1 and PC_2 communicate in unicast mode.

Intra-Subnet Packet Forwarding

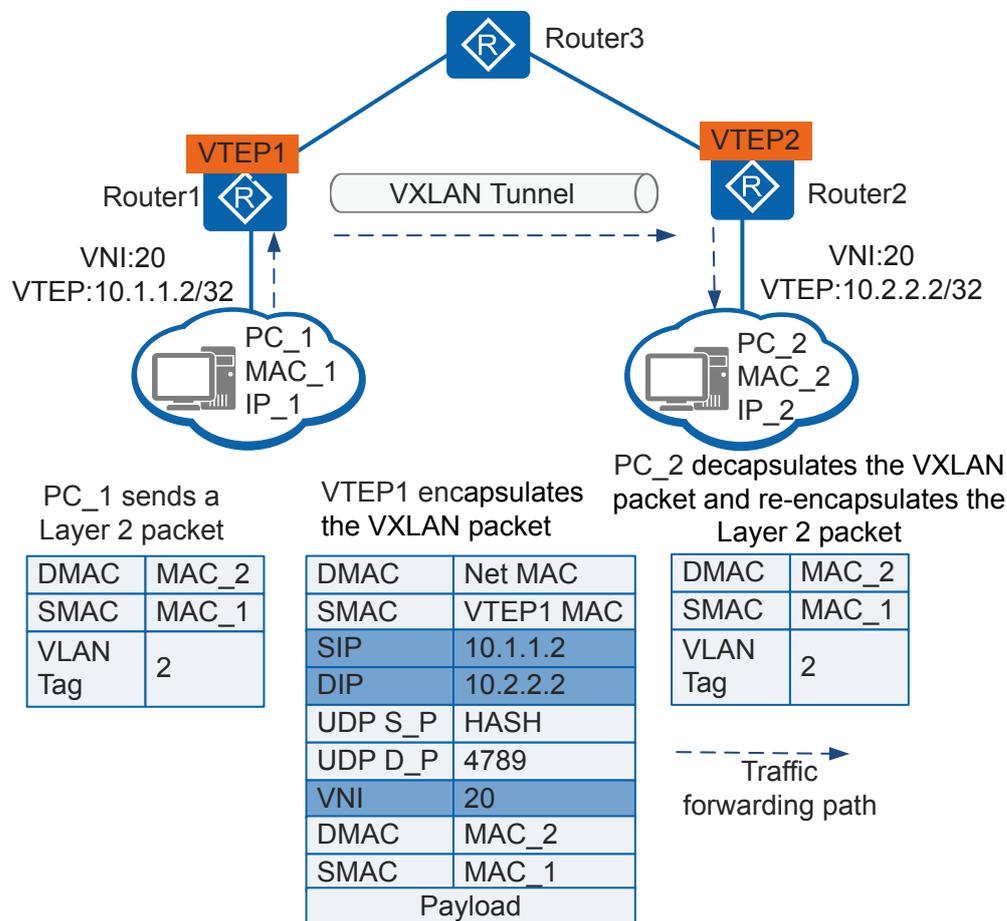
The packet forwarding process is classified into known unicast packet forwarding and broadcast, unknown unicast, and multicast (BUM) packet forwarding based on the type of destination MAC address in original packets.

Only VXLAN Layer 2 gateways can implement known unicast packet forwarding in a subnet and BUM packet forwarding. The forwarding processes do not require any Layer 3 gateway.

● **Forwarding Process of Known Unicast Packets**

Figure 14-9 shows the known unicast packet forwarding process.

Figure 14-9 Forwarding process of known unicast packets



- After Router1 receives a packet from PC_1, Router1 determines the Layer 2 bridge domain (BD) of the packet based on the access interface and VLAN information carried in the packet, and searches for the outbound interface and encapsulation information based on the BD.
- The VXLAN Tunnel Endpoint (VTEP1) on Router1 encapsulates the packet based on the found encapsulation information and forwards the packet to the outbound interface.
- After the VTEP2 on Router2 receives the VXLAN packet, it verifies the UDP destination port number, source and destination IP addresses, and VXLAN Network Identifier (VNI) of the packet to determine its validity. After confirming that the packet is valid, the VTEP obtains the BD based on the VNI and decapsulates the VXLAN packet to obtain the inner Layer 2 packet.
- Router2 finds out the outbound interface and encapsulation information in the local MAC address table based on the destination MAC address in the inner Layer 2 packet, adds a VLAN tag to the packet, and forwards it to PC_2.

Packet forwarding from PC_2 to PC_1 is similar to that described above, and is not mentioned here.

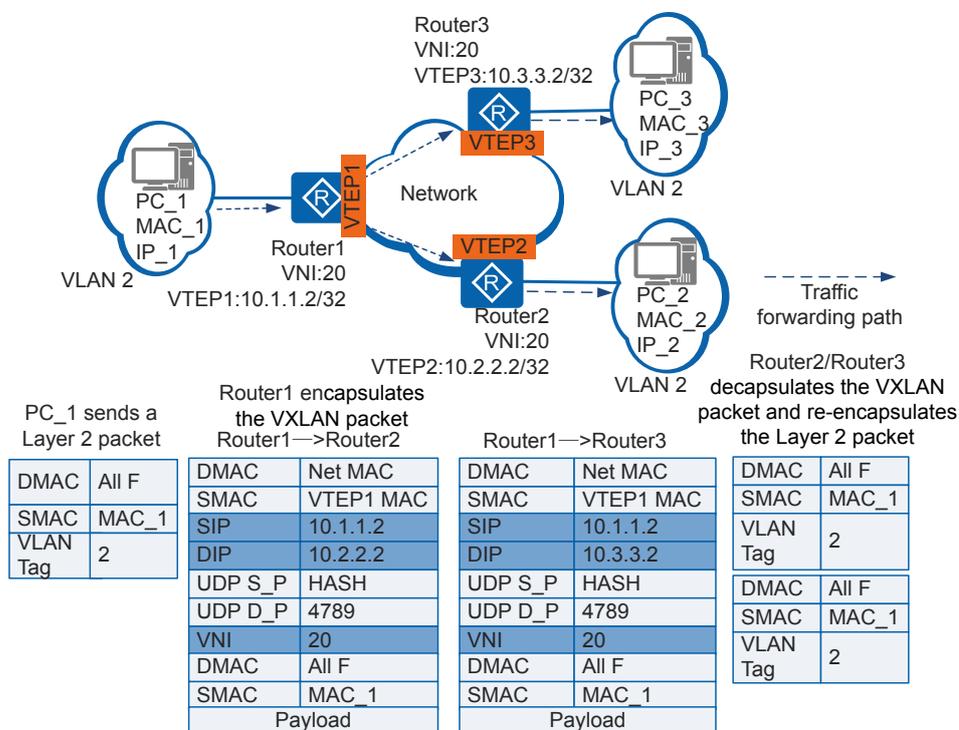
● **BUM Packet Forwarding Process**

After a BUM packet enters the VXLAN tunnel, the source VTEP replicates the BUM packet based on the tunnel list and encapsulates the original and replicated packets. When these packets leave the VXLAN tunnels, the destination VTEPs decapsulate them. **Figure 14-10** shows the BUM packet forwarding process.

NOTE

On a VXLAN network, multiple destination VTEP IP addresses can be configured for a VNI, and the list of these IP addresses is regarded as a tunnel list. After an interface receives the BUM packet, the source VTEP replicates the packet based on the tunnel list and forwards it to all the VTEPs with the same VNI.

Figure 14-10 BUM packet forwarding process



- After Router1 receives a packet from PC_1, Router1 determines the BD of the packet based on the access interface and VLAN information carried in the packet.
- The VTEP1 on Router1 obtains the tunnel list for the VNI based on the BD, replicates the packet based on the tunnel list, and performs VXLAN tunnel encapsulation before forwarding it to the outbound interface.
- After the VTEP2 on Router2 or Router3 receives the VXLAN packet, it verifies the UDP destination port number, source and destination IP addresses, and VNI of the packet to determine its validity. After confirming that the packet is valid, the VTEP obtains the BD based on the VNI and decapsulates the VXLAN packet to obtain the inner Layer 2 packet.
- Router2 or Router3 checks the destination MAC address of the inner Layer 2 packet and finds that it is a BUM packet, Router2 or Router3 then broadcasts the packet in

the corresponding BD to the user side. Router2 or Router3 finds out all outbound interfaces to the user side and encapsulation information in the local MAC address table, adds a VLAN tag to the packet, and forwards it to PC_2 or PC_3.

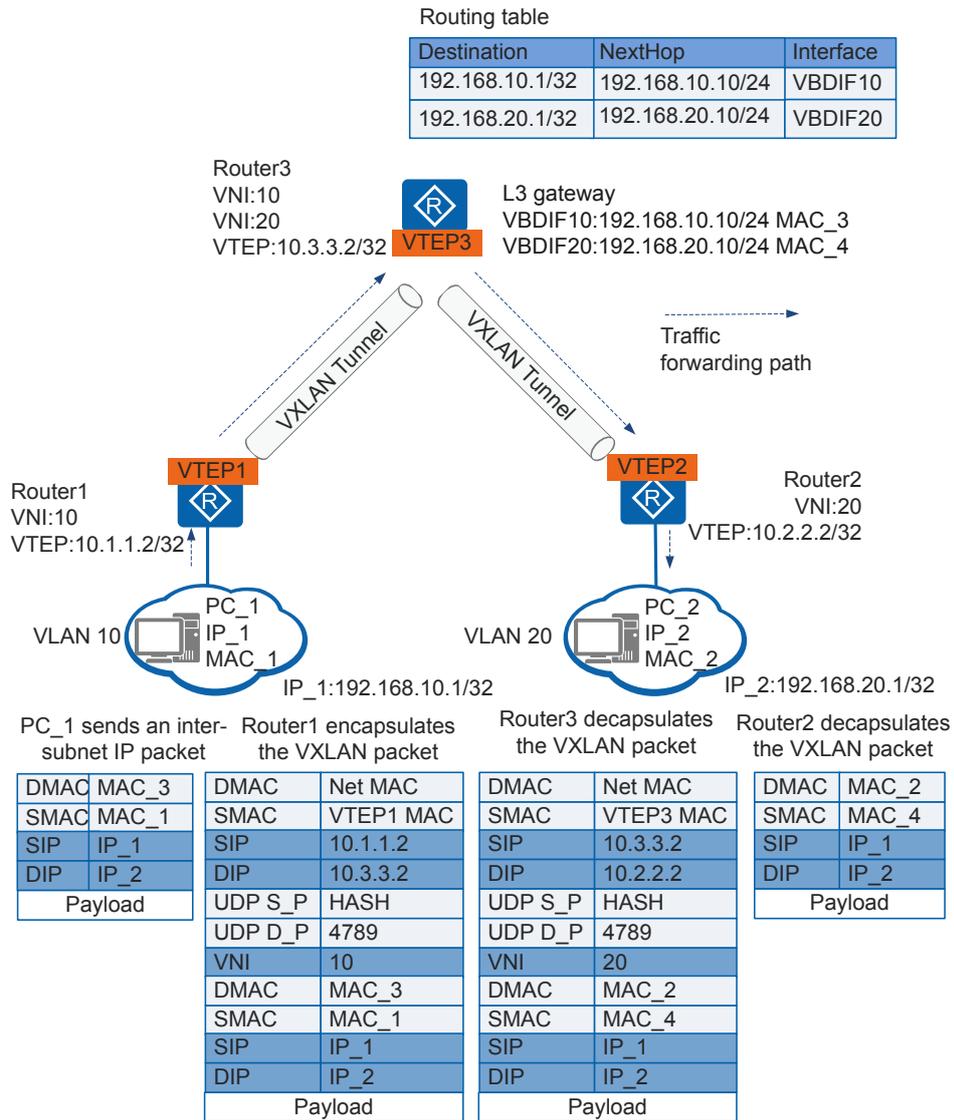
Packet forwarding from PC_2 or PC_3 to PC_1 is similar to the known unicast packet forwarding process, and is not mentioned here.

Inter-Subnet Packet Forwarding

If end users in a VXLAN site need to access the Internet or communicate with end users in another VXLAN site, a VXLAN Layer 3 gateway needs to be deployed to provide end users with Layer 3 services. [Figure 14-11](#) shows the inter-subnet packet forwarding process.

IP addresses of PC_1 and PC_2 are in different network segments. For the first time, PC_1 needs to broadcast an ARP request packet to request the MAC address of VBDIF 10. After obtaining the MAC address of the gateway, PC_1 sends a data packet to the gateway. The gateway also broadcasts an ARP request packet to request the MAC address of PC_2. After obtaining the MAC address, the gateway forwards the data packet to PC_2. The preceding MAC address learning process is the same as that in [MAC Address Learning](#).

Figure 14-11 Inter-subnet packet forwarding process



1. After Router1 receives a packet from PC₁, Router1 determines the Layer 2 BD of the packet based on the access interface and VLAN information carried in the packet, and searches for the outbound interface and encapsulation information based on the BD.
2. The VTEP1 on Router1 performs VXLAN tunnel encapsulation based on the outbound interface and encapsulation information, and forwards the packet to Router3.
3. Router3 decapsulates the received VXLAN packet, finds that the destination MAC address in the inner packet is MAC₃ of the Layer 3 gateway interface VBDIF10, and determines that the packet needs to be forwarded at Layer 3.
4. Router3 removes the Ethernet header from the inner packet to parse the destination IP address. It then searches the routing table based on the destination IP to obtain the next-hop address, and searches ARP entries based on the next hop to obtain the destination MAC address, VXLAN tunnel outbound interface, and VNI.

5. Router3 re-encapsulates the VXLAN packet and forwards it to Router2. The source MAC address in the Ethernet header of the inner packet is MAC_4 of the Layer 3 gateway interface VBDIF20.
6. After the VTEP2 on Router2 receives the VXLAN packet, it verifies the UDP destination port number, source and destination IP addresses, and VXLAN Network Identifier (VNI) of the packet to determine its validity. The VTEP then obtains the BD based on the VNI, decapsulates the packet to obtain the inner Layer 2 packet, and searches for the outbound interface and encapsulation information in the corresponding BD.
7. Router2 adds a VLAN tag to the packet based on the outbound interface and encapsulation information, and forwards the packet to PC_2.

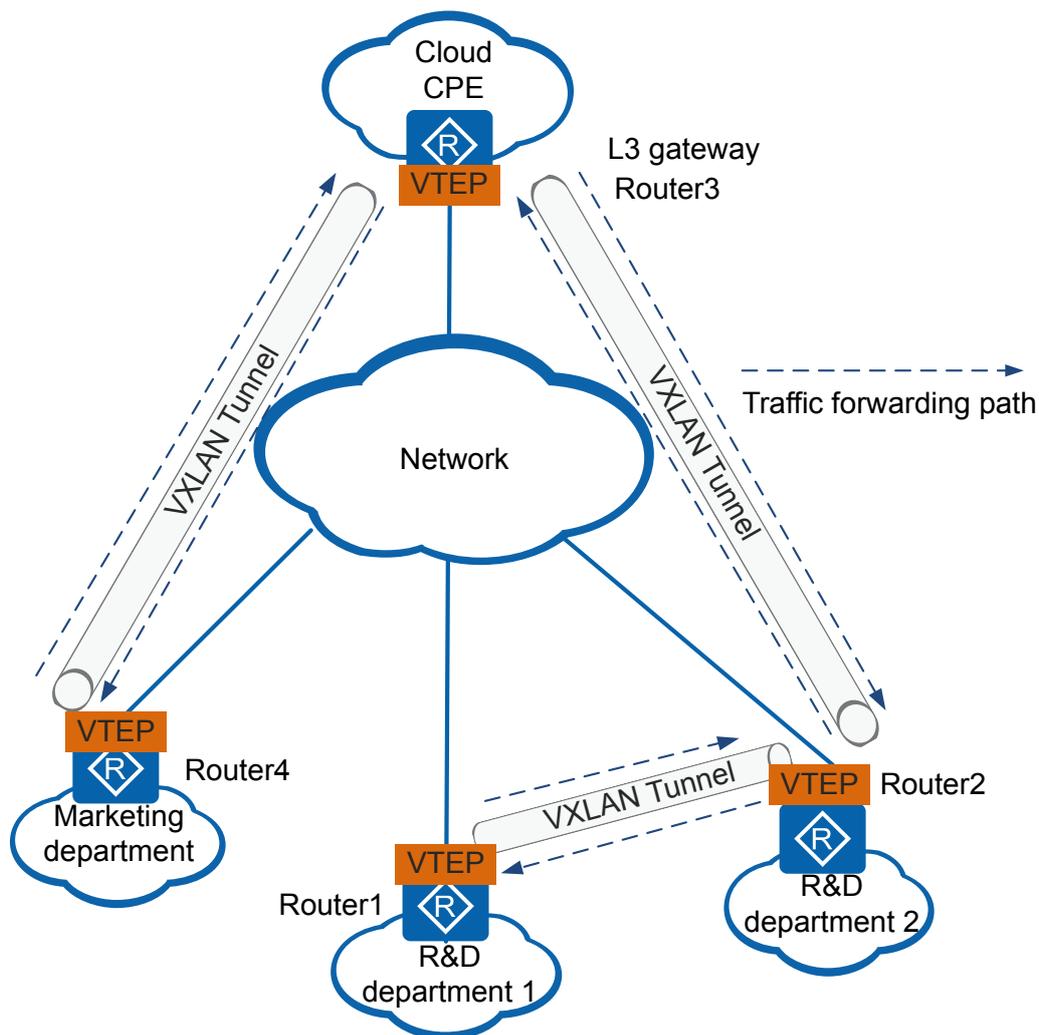
Packet forwarding from PC_2 to PC_1 is similar to that described above, and is not mentioned here.

14.3 Application Scenario

Networking Description

In [Figure 14-12](#), an enterprise deploys its departments in different areas. To facilitate management and maintenance, departments with the same service requirements are planned in the same network segment, while those with different service requirements are planned in different network segments. End users in the same or different departments need to communicate with each other. For example, R&D departments 1 and 2 need to communicate in the same network segment; the R&D department 2 and marketing department need to communicate across different network segments.

Figure 14-12 Communication between end users on a VXLAN network



VXLAN provides Layer 2 interconnection for dispersed physical sites. For example, in [Figure 14-12](#), Router1 and Router2 are VXLAN Layer 2 gateway, and they establish a VXLAN tunnel to enable end users in R&D departments 1 and 2 to communicate with each other in the same network segment.

VXLAN provides Layer 3 interconnection for tenants in different sites. For example, when the R&D department 2 wants to communicate with the marketing department, Router3 functions as the VXLAN Layer 3 gateway to establish VXLAN tunnels with Router2 and Router4 respectively.

After static VXLAN tunnels are established between the routers, they dynamically learn flow table information, such as MAC address entries and ARP entries. After flow table information is learned, end users in the same or different network segments can communicate with each other over the VXLAN tunnels.

14.4 Licensing Requirements and Limitations for VXLAN

This section describes VXLAN configuration notes.

Involved Network Element

None

License Support

VXLAN is a basic feature of the device and is not under license control.

Feature Dependencies and Limitations

- The router does not support VXLAN over MPLS LSP tunnel. If VXLAN packets received from a peer are encapsulated by MPLS, the VTEP fails to decapsulate the packets.
- The router does not support VXLAN over GRE tunnel. If VXLAN packets received from a peer are encapsulated by GRE, the VTEP fails to decapsulate the packets.
- In VXLAN scenarios, do not advertise routes destined for the local VTEP address to the peer end of a VXLAN tunnel through a VBDIF interface during route configuration. Otherwise, the next hop of the remote VTEP address of the VXLAN tunnel may be the VBDIF interface on the ingress of the tunnel, causing a loop on the device.

Only the AR100-S, AR110-S, AR120-S, AR150-S, AR160-S, AR200-S, AR1200-S series and AR2220E-S support VXLAN.

Only the AR100-S, AR120-S, AR150-S, AR160-S, AR200-S, AR1200-S series and AR2220E-S support EVPN.

AR2220E-S and AR2240-S&AR3260-S series (except AR2240C-S) can run the **portswitch** command to change the working mode of all ports on the SRU, 4GECS and 2X10GL from Layer 3 mode to Layer 2 mode. The interface that change the working mode to layer 2 only support VXLAN.

14.5 Configuring VXLAN (in Static Mode)

Pre-configuration Tasks

Before configuring user communication over VXLAN tunnels, ensure Layer 3 route reachability.

Configuration Process

Figure 14-13 shows the process of configuring communication between end users in the same VXLAN network segment.

Figure 14-13 Configuring communication between end users in the same VXLAN network segment

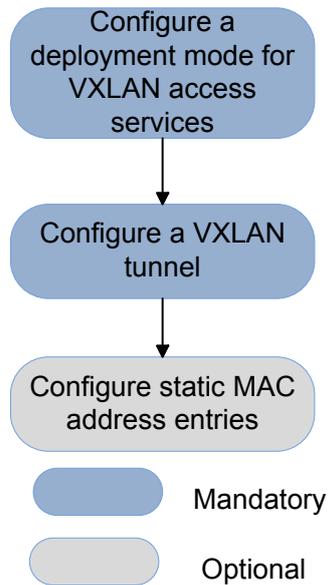
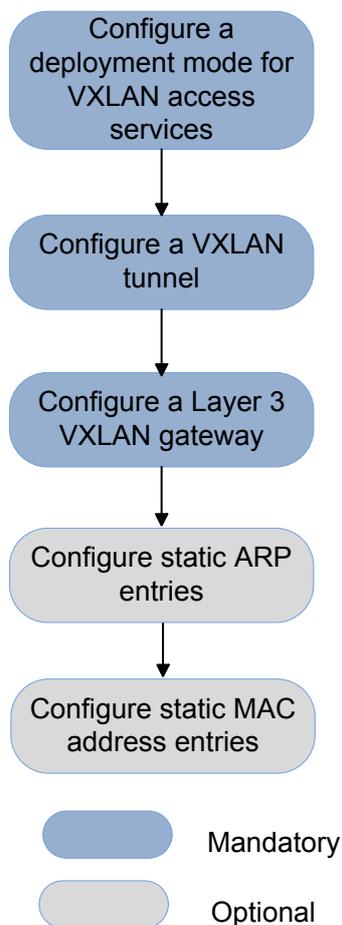


Figure 14-14 shows the process of configuring communication between end users in different VXLAN network segments.

Figure 14-14 Configuring communication between end users in different VXLAN network segments



14.5.1 Configuring Deployment Mode for VXLAN Access Service

Context

When configuring VXLAN on a device, you need to select a deployment mode for the VXLAN access service on the downlink interface.

At the access side, two methods are available for deploying VXLAN services:

- Based on VLAN: You can associate one or more VLANs with a BD to add users in these VLANs to the BD. This VLAN-based mode implements larger-granularity control, but is easy to configure. It applies to VXLAN deployment on a live network.
- Based on encapsulation mode: The device sends packets of different encapsulation modes to different Layer 2 sub-interfaces based on the VLAN tags contained in the packets. You can bind a Layer 2 sub-interface to a BD to add specified users to the BD. This mode implements refined and flexible control but requires more complex configuration. It applies to VXLAN deployment on a new network.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bridge-domain** *bd-id*

A BD is created and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run **description** *description*

The description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run **quit**

Exit from the BD view and return to the system view.

Step 5 Configure a service access point.

- Based on VLAN:
 - a. Run **vlan** *vlan-id*
A VLAN is created and the VLAN view is displayed.
 - b. Run **quit**
Exit from the VLAN view and return to the system view.
 - c. Run **bridge-domain** *bd-id*
The view of an existing BD is displayed.
 - d. Run **I2 binding vlan** *vlan-id*
A VLAN is associated with the BD so that data packets can be forwarded in the BD.
By default, a VLAN is not associated with a BD.

NOTE

- The VLANs to be bound to the BD have been created.
 - One VLAN can be associated with only one BD, but one BD can be associated with multiple VLANs.
 - After a global VLAN is associated with a BD, you need to add corresponding interfaces to the VLAN. An Eth-Trunk cannot be added to the VLAN.
- Based on encapsulation mode:
 - a. Run **interface** *interface-type interface-number.subnum mode I2*
A Layer 2 sub-interface is created, and the sub-interface view is displayed.
 - b. Run **encapsulation** { **dot1q vid** *pe-vid* | **default** | **untag** | **qinq vid** *vlan-vid ce-vid* }
An encapsulation mode is configured for a Layer 2 sub-interface to specify the type of packets that can pass through the sub-interface.
By default, the encapsulation mode of packets allowed to pass a Layer 2 sub-interface is not configured.

 NOTE

When configuring an encapsulation mode on a Layer 2 sub-interface, pay attention to the following points:

- The VLAN ID in **dot1q** mode or outer VLAN ID in **qinq** mode cannot be the same as the allowed VLAN of the corresponding main interface or the global VLAN.
- On the same main interface, the VLAN ID in **dot1q** mode and the outer VLAN ID in **qinq** mode must be different.
- After NAC authentication is configured on the main interface, the traffic encapsulation type on a Layer 2 sub-interface cannot be set to **default**.
- When the encapsulation mode of a Layer 2 sub-interface is **default**, the corresponding main interface cannot be added to any VLAN, including VLAN 1.
- Before the encapsulation mode of a Layer 2 sub-interface is set to **default**, the main interface has only one sub-interface.
- After the encapsulation mode of a Layer 2 sub-interface is set to **default**, no other sub-interface can be created on the main interface.
- When the encapsulation mode of a Layer 2 sub-interface is set to **untag**, other sub-interfaces of the main interface cannot be set to **untag**.
- You can configure only one encapsulation mode for each Layer 2 sub-interface. If an encapsulation mode has been configured for a Layer 2 sub-interface, run the **undo encapsulation** command to delete the original mode before you configure another mode.
- When the device functions as a Layer 3 VXLAN gateway, the traffic encapsulation type on a Layer 2 sub-interface cannot be set to **default**.

c. Run **bridge-domain** *bd-id*

A specified Layer 2 sub-interface is associated with a BD so that data packets can be forwarded in the BD.

By default, a Layer 2 sub-interface is not associated with a BD.

---End

14.5.2 Configuring a VXLAN Tunnel

Context

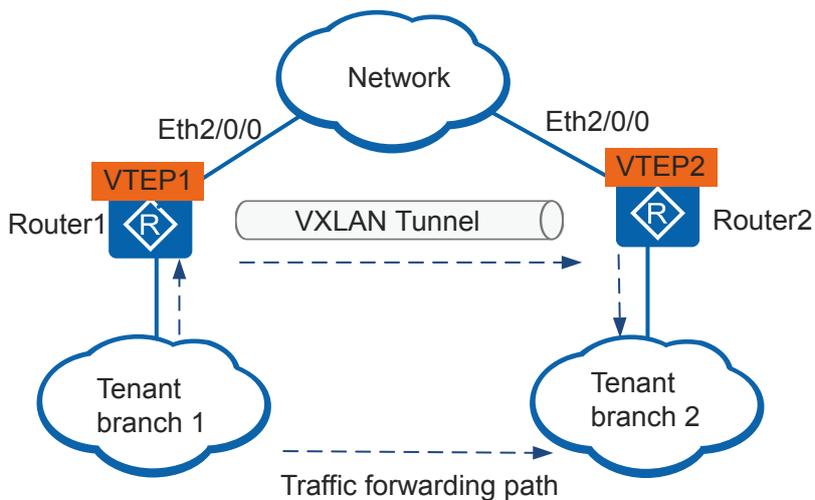
When configuring VXLAN on a device, you need to configure related information for VXLAN tunnel establishment on an uplink interface.

A VXLAN tunnel is established based on the IP addresses of two VXLAN Tunnel Endpoints (VTEPs). Therefore, you need to configure the source VTEP IP address and destination VTEP IP address on the devices on both ends of a tunnel.

Take Router1 in [Figure 14-15](#) as an example. The following describes the configurations required for establishment of a VXLAN tunnel:

- Source VTEP IP address: source IP address in a VXLAN packet, that is, IP address of Eth2/0/0 on Router1
- Destination VTEP IP address: destination IP address in a VXLAN packet, that is, IP address of Eth2/0/0 on Router2

Figure 14-15 VXLAN tunnel establishment



NOTE

- You need to run the **vni head-end peer-list** command to configure the corresponding VTEP address even if the source VTEP matches only one destination VTEP.
- Run the **ping** command to check whether a reachable route exists between two ends of the tunnel. If there is a reachable route, the tunnel can be established and packets can be normally forwarded. If the two devices have a route to each other but the route is unreachable, the tunnel can still go Up but packets cannot be forwarded.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bridge-domain *bd-id***

The BD view is displayed.

Step 3 Run **vxlan vni *vni-id***

A VNI is configured for the BD.

By default, no VNI is associated with a BD.

Step 4 Run **quit**

Exit from the BD view and return to the system view.

Step 5 Run **interface nve *nve-number***

An NVE interface is created, and the NVE interface view is displayed.

Step 6 Run **source *ip-address***

An IP address is configured for the source VTEP.

By default, no IP address is configured for a source VTEP.

Step 7 Run `vni vni-id head-end peer-list ip-address &<1-10>`

An IP address is configured for the destination VTEP with a specified VNI.

By default, no IP address is configured for the destination VTEP with a specified VNI.

Step 8 Run `quit`

Exit from the NVE interface view and return to the system view.

---End

14.5.3 Configuring a Layer 3 VXLAN Gateway

Context

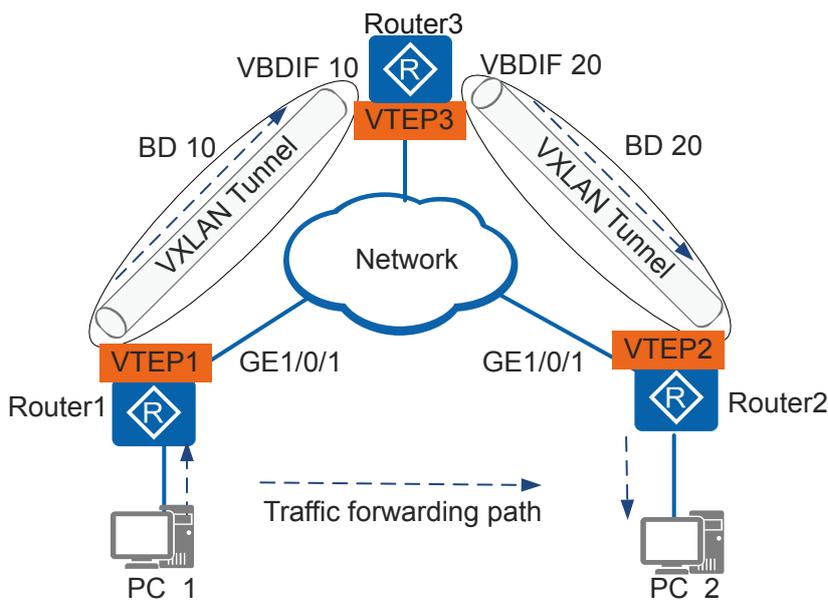
A VBDIF interface is configured on a VXLAN Layer 3 gateway to forward packets across network segments. You do not need to create a VBDIF interface for communication between users in the same network segment.

If end users in a VXLAN site need to access the Internet or communicate with end users in another VXLAN site, a VXLAN Layer 3 gateway needs to be deployed to provide end users with Layer 3 services.

In [Figure 14-16](#), after you create a logical Layer 3 VBDIF interface and configure an IP address for the VBDIF interface, the VBDIF interface functions as the gateway for tenants in the BD to forward packets at Layer 3 based on the IP address. Each BD has only one VBDIF interface.

To ensure that users in different network segments can communicate with each other, ensure that the default gateway address is the IP address of the VBDIF interface on the VXLAN Layer 3 gateway.

Figure 14-16 Layer 3 VXLAN gateway networking



Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **interface vbdif** *bd-id*

A VBDIF interface is created and the VBDIF interface view is displayed.

NOTE

The number of the VBDIF interface must match an existing BD ID.

Step 3 Run **ip address** *ip-address* { *mask* | *mask-length* } [**sub**]

An IP address is configured for the VBDIF interface to implement Layer 3 communication.

By default, no IP address is configured for a VBDIF interface.

Step 4 (Optional) Run **mac-address** *mac-address*

A MAC address is configured for the VBDIF interface.

By default, the MAC address of a VBDIF interface is the system MAC address.

----End

14.5.4 (Optional) Configuring Static ARP Entries

Context

Static ARP entries are manually configured and maintained. They cannot be aged and overridden by dynamic ARP entries. You can configure static ARP entries on a Layer 3 VXLAN gateway to improve communication security. Static ARP entries ensure communication between the local device and a specified device by using a specified MAC address so that attackers cannot modify mappings between IP addresses and MAC addresses in static ARP entries.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **arp static** *ip-address mac-address vni vni-id source-ip source-ip-address peer-ip peer-ip-address*

A static ARP entry of a VXLAN tunnel is configured.

By default, a static MAC address entry of a VXLAN tunnel is not configured.

NOTE

The value of *ip-address* must be in the same network segment as that of the Layer 3 gateway.

Step 3 On a VXLAN network, mapping between IP addresses and MAC addresses can be configured on the user access side. When you configure static ARP entries by VLAN, the **interface interface-type interface-number** parameter must be a Layer 2 Ethernet interface. When you configure static ARP entries by traffic encapsulation type, the **interface interface-type interface-number** parameter must be a Layer 2 sub-interface.

- By VLAN:
Run the **arp static ip-address mac-address vni vni-id vid vlan-id interface interface-type interface-number** command to configure static ARP entries.
- By traffic encapsulation type:
 - If the traffic encapsulation type is **dot1q**, run the **arp static ip-address mac-address vni vni-id vid vlan-id interface interface-type interface-number** command to configure static ARP entries. The value of **vid vlan-id** in this command must be the same as that in the **encapsulation dot1q vid pe-vid** command.
 - If the traffic encapsulation type is **qinq**, run the **arp static ip-address mac-address vni vni-id vid vlan-id cevid ce-vid interface interface-type interface-number** command to configure static ARP entries. The values of **vid vlan-id** and **cevid ce-vid** in this command must be the same as the values in the **encapsulation qinq vid vlan-vid ce-vid ce-vid** command.
 - If the traffic encapsulation type is **untag**, run the **arp static ip-address mac-address vni vni-id interface interface-type interface-number** command to configure static ARP entries.

----End

Checking the Configuration

Run the **display arp [all | brief]** command to view all ARP entries.

14.5.5 (Optional) Configuring a Static MAC Address Entry

Context

To reduce broadcast traffic and improve network security, you can configure static MAC address entries on the VXLAN-enabled device to specify the forwarding path of BUM packets. This prevents unauthorized users from intercepting data of authorized users.

For packet forwarding in different directions, static MAC address entries are configured in two directions:

- Configure static MAC address entries on the VXLAN side to specify the VXLAN encapsulation direction when traffic is forwarded from the LAN side to the VXLAN side.
- Configure static MAC address entries on the LAN side to specify the outbound interface when traffic is forwarded from the VXLAN side to the LAN side.

Procedure

- Configure a static MAC address entry on the VXLAN side.
 - a. Run **system-view**
The system view is displayed.
 - b. Run **mac-address static mac-address bridge-domain bd-id source source-ip-address peer peer-ip-address vni vni-id**
A static MAC address entry of a VXLAN tunnel is configured.
By default, a static MAC address entry of a VXLAN tunnel is not configured.
- Configure a static MAC address entry on the LAN side.

- a. Run **system-view**
The system view is displayed.
- b. Run **mac-address static** *mac-address interface-type interface-number.subnum*
bridge-domain *bd-id* { **default** | **untag** | **vid** *vid* }
A static MAC address entry based on a BD is configured.
By default, no static MAC address entry based on a BD is configured.

----End

Checking the Configuration

Run the **display mac-address** [*mac-address*] **bridge-domain** *bd-id* command to check MAC address entries in a BD.

14.5.6 Verifying the VXLAN Configuration in Centralized Gateway Mode Using Static Mode

Context

After you complete configuring VXLAN service access points and VXLAN tunnels, run the following commands to verify the VXLAN configuration.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to view the BD configuration.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to view VXLAN tunnel information.
- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to view VXLAN configuration of a specified VNI or all VNIs.
- Run the **display vxlan peer** [**vni** *vni-id*] command to view the destination VTEP IP address with a specified VNI.

----End

14.6 Configuring VXLAN (in BGP EVPN Mode)

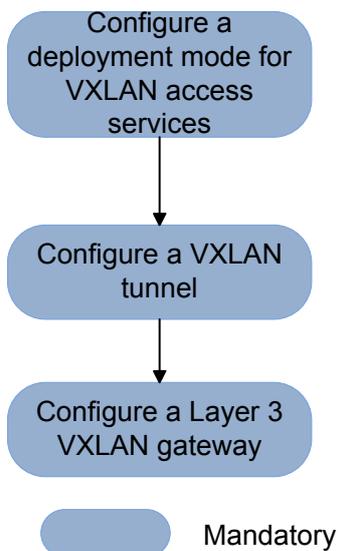
Pre-configuration Tasks

Before configuring user communication over VXLAN tunnels, ensure Layer 3 route reachability.

Configuration Process

Figure 14-17 shows the process of configuring communication between end users in different VXLAN network segments.

Figure 14-17 Configuring communication between end users in different VXLAN network segments



14.6.1 Configuring Deployment Mode for VXLAN Access Service

Context

When configuring VXLAN on a device, you need to select a deployment mode for the VXLAN access service on the downlink interface.

At the access side, two methods are available for deploying VXLAN services:

- Based on VLAN: You can associate one or multiple VLANs with a BD to add users in these VLANs to the BD. This VLAN-based mode implements larger-granularity control, but is easy to configure. It applies to VXLAN deployment on a live network.
- Based on encapsulation mode: The device sends packets of different encapsulation modes to different Layer 2 sub-interfaces based on the VLAN tags contained in the packets. You can bind a Layer 2 sub-interface to a BD to add specified users to the BD. This mode implements refined and flexible control but requires more complex configuration. It applies to VXLAN deployment on a new network.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bridge-domain** *bd-id*

A BD is created and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run **description** *description*

The description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run **quit**

Exit from the BD view and return to the system view.

Step 5 Configure a service access point.

- Based on VLAN:
 - a. Run **vlan** *vlan-id*
A VLAN is created and the VLAN view is displayed.
 - b. Run **quit**
Exit from the VLAN view and return to the system view.
 - c. Run **bridge-domain** *bd-id*
The view of an existing BD is displayed.
 - d. Run **I2 binding vlan** *vlan-id*
A VLAN is associated with the BD so that data packets can be forwarded in the BD.
By default, a VLAN is not associated with a BD.

NOTE

- The VLANs to be bound to the BD have been created.
 - One VLAN can be associated with only one BD, but one BD can be associated with multiple VLANs.
 - After a global VLAN is associated with a BD, you need to add corresponding interfaces to the VLAN. An Eth-Trunk cannot be added to the VLAN.
- Based on encapsulation mode:
 - a. Run **interface** *interface-type interface-number.subnum mode I2*
A Layer 2 sub-interface is created, and the sub-interface view is displayed.
By default, no Layer 2 sub-interface is created.
 - b. Run **encapsulation** { **dot1q** { **vid** *pe-vid* } | **default** | **untag** | **qinq** { **vid** *vlan-vid ce-vid* } }
An encapsulation mode is configured for a Layer 2 sub-interface to specify the type of packets that can pass through the sub-interface.
By default, the encapsulation mode of packets allowed to pass a Layer 2 sub-interface is not configured.
 - c. Run **bridge-domain** *bd-id*
A specified Layer 2 sub-interface is associated with a BD so that data packets can be forwarded in the BD.
By default, a Layer 2 sub-interface is not associated with a BD.

 **NOTE**

When configuring an encapsulation mode on a Layer 2 sub-interface, pay attention to the following points:

- The VLAN ID in **dot1q** mode or outer VLAN ID in **qinq** mode cannot be the same as the allowed VLAN of the corresponding main interface or the global VLAN.
- On the same main interface, the VLAN ID in **dot1q** mode and the outer VLAN ID in **qinq** mode must be different.
- When the encapsulation mode of a Layer 2 sub-interface is **default**, the corresponding main interface cannot be added to any VLAN, including VLAN 1.
- Before the encapsulation mode of a Layer 2 sub-interface is set to **default**, the main interface has only one sub-interface.
- After the encapsulation mode of a Layer 2 sub-interface is set to **default**, no other sub-interface can be created on the main interface.
- When the encapsulation mode of a Layer 2 sub-interface is set to **untag**, other sub-interfaces of the main interface cannot be set to **untag**.
- When the device functions as a Layer 3 VXLAN gateway, the traffic encapsulation type on a Layer 2 sub-interface cannot be set to **default**.

----End

14.6.2 Configuring a VXLAN Tunnel

Context

When deploying a VXLAN network, you need to configure the uplink interfaces of the devices for VXLAN tunnel establishment.

A VXLAN tunnel is established between IP addresses of two virtual tunnel end points (VTEPs). Border Gateway Protocol (BGP) Ethernet VPN (EVPN) can be used to dynamically establish VXLAN tunnels by establishing a BGP EVPN peer relationship between two VTEPs and using BGP EVPN routes to transmit VXLAN network identifiers (VNIs) and VTEP IP addresses between the peer.

 **NOTE**

A VXLAN tunnel is specified by a pair of VTEP IP addresses. When a local VTEP receives the same remote VTEP IP address repeatedly, only one VXLAN tunnel can be established, but packets are encapsulated with different VNIs before being forwarded through the tunnel.

Procedure

Step 1 Establish a BGP EVPN peer relationship.

1. Run **system-view**

The system view is displayed.

2. Run **bgp as-number**

The BGP view is displayed.

3. Run **peer ipv4-address as-number { as-number-plain | as-number-dot }**

The remote PE is configured as a BGP EVPN peer.

4. Run **peer ipv4-address connect-interface loopback interface-number**

An interface is specified for setting up a TCP connection with the BGP EVPN peer.

5. Run **l2vpn-family evpn**

The BGP-EVPN address family view is displayed.

6. Run **peer { ipv4-address | group-name } enable**

The ability to exchange EVPN routes with the peer or in a group is enabled.

7. (Optional) Run **peer ipv4-address group group-name**

The BGP EVPN peer is added to a peer group.

Adding the BGP EVPN peer to a peer group simplifies configuration and management of the BGP network.

8. (Optional) Run **peer { group-name | ipv4-address } route-policy route-policy-name export**

The BGP EVPN peer (group) is configured to advertise only specified routes.

To strictly control EVPN route advertisement, you need to configure an export routing policy. It can filter routes to be advertised to other EVPN peers (group).

9. (Optional) Run **peer { ipv4-address | group-name } route-policy route-policy-name import**

The BGP EVPN peer (group) is configured to receive only specified routes.

To strictly control EVPN route acceptance, you need to configure an import routing policy. It can filter routes received from other EVPN peers (group).

10. (Optional) Run **undo policy vpn-target**

The device is disabled from filtering received EVPN routes by the VPN target.

11. (Optional) Run **peer { group-name | ipv4-address } mac-limit mac-limit [idle-forever | idle-timeout times]**

The maximum number of MAC advertisement routes received from a peer is specified.

An EVPN instance may import many unused MAC advertisement routes from some peers. It is recommended that you run this command when the number of received MAC advertisement routes from the peers occupies a large percentage of the total number of MAC advertisement routes on the device.

12. Run **quit**

Exit from the BGP-EVPN address family view and return to the BGP view.

13. Run **quit**

Exit from the BGP view and return to the system view.

Step 2 Configure an IP address for the source VTEP.

1. Run **interface nve nve-number**

An NVE interface is created, and the NVE interface view is displayed.

By default, no NVE interface is created.

2. Run **source ip-address**

An IP address is configured for the source VTEP.

By default, no IP address is configured for a source VTEP.

----End

14.6.3 Configuring a Layer 3 VXLAN Gateway

Context

A VBDIF interface is configured on a Layer 3 VXLAN gateway to forward packets across network segments. You do not need to create a VBDIF interface for communication between users in the same network segment.

If end users in a VXLAN site need to access the Internet or communicate with end users in another VXLAN site, a Layer 3 VXLAN gateway needs to be deployed to provide end users with Layer 3 services.

After you create a logical Layer 3 VBDIF interface and configure an IP address for the VBDIF interface, the VBDIF interface functions as the gateway for tenants in the BD to forward packets at Layer 3 based on the IP address. Each BD has only one VBDIF interface.

To ensure that users in different network segments can communicate with each other, ensure that the default gateway address is the IP address of the VBDIF interface on the Layer 3 VXLAN gateway.

Procedure

Step 1 Configure a VPN instance.

1. Run **system-view**

The system view is displayed.

2. Run **ip vpn-instance** *vpn-instance-name*

A VPN instance is created, and the VPN instance view is displayed.

3. Run **ipv4-family**

The IPv4 address family is enabled for the VPN instance and the VPN instance IPv4 address family view is displayed.

By default, the IPv4 address family is disabled for the VPN instance.

4. Run **route-distinguisher** *route-distinguisher*

A route distinguisher (RD) is configured for the VPN instance.

By default, no RD is configured for the VPN instance.

5. Run **vpn-target** *vpn-target* &<1-8> [**both** | **export-extcommunity** | **import-extcommunity**] **evpn**

The VPN target is configured for the VPN instance.

A VPN target is a BGP extended community attribute. It is used to control the receiving and advertisement of EVPN routing information. A maximum of eight VPN targets can be configured using a **vpn-target** command. If you want to configure more VPN targets in the EVPN instance address family, run the **vpn-target evpn** command multiple times.

6. (Optional) Run **export route-policy** *policy-name* **evpn**

The VPN instance IPv4 address family is associated with an export routing policy to filter EVPN routes to be advertised to the EVPN address family.

To strictly control EVPN route advertisement, you need to configure an export routing policy. It can filter routes to be advertised to the EVPN address family.

7. (Optional) Run **import route-policy** *policy-name evpn*

The VPN instance IPv4 address family is associated with an import routing policy to filter EVPN routes received from the EVPN address family.

To strictly control EVPN route acceptance, you need to configure an import routing policy. It can filter routes received from the EVPN address family.

8. Run **quit**

Exit from the VPN instance IPv4 address family view and return to the VPN instance view.

9. Run **vlan vni** *vni-id*

A VNI is bound to the VPN instance.

By default, no VNI is bound to the VPN instance.

10. Run **quit**

Exit from the VPN instance view and return to the system view.

Step 2 Configure a Layer 3 gateway and bind the VPN instance to it.

1. Run **interface vbdif** *bd-id*

A VBDIF interface is created and the VBDIF interface view is displayed.

 **NOTE**

The number of the VBDIF interface must match an existing BD ID.

2. Run **ip binding vpn-instance** *vpn-instance-name*

The VBDIF interface is bound to the VPN instance.

 **NOTE**

- Using the **ip binding vpn-instance** command will delete Layer 3 configurations such as the IP address and routing protocol on the VBDIF interface. Reconfigure them if needed.
- An interface cannot be bound to a VPN instance that is not enabled with an address family.

3. Run **ip address** *ip-address { mask | mask-length } [sub]*

An IP address is configured for the VBDIF interface to implement Layer 3 communication.

By default, no IP address is configured for a VBDIF interface.

4. (Optional) Run **mac-address** *mac-address*

A MAC address is configured for the VBDIF interface.

By default, the MAC address of a VBDIF interface is the system MAC address.

5. Run **quit**

Exit from the VBDIF interface view and return to the system view.

Step 3 Configure VXLAN gateways to advertise IP prefix routes to each other.

1. Run **bgp** *as-number*

The BGP view is displayed.

2. Run **ipv4-family vpn-instance** *vpn-instance-name*

The BGP-VPN instance IPv4 address family view is displayed.

3. Run **import-route** *protocol* [*process-id*] [**med** *med* | **route-policy** *route-policy-name*]
*

Routes of other protocols are imported to the BGP-VPN instance IPv4 address family view.

To enable advertisement of host IP routes, you only need to configure the device to import direct routes. To enable advertisement of network segment routes, advertise these routes using a dynamic routing protocol such as OSPF and enable the device to import routes of dynamic routing protocols.

4. Run **advertise l2vpn evpn**

The VPN instance is enabled to advertise IP routes to the BGP-EVPN address family.

By default, the VPN instance is disabled from advertising IP routes to the BGP-EVPN address family.

---End

14.6.4 Checking the Configuration

Context

After you complete configuring VXLAN service access points and VXLAN tunnels, run the following commands to check the VXLAN configuration.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to view the BD configuration.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to view VXLAN tunnel information.
- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to view VXLAN configuration of a specified VNI or all VNIs.
- Run the **display bgp evpn routing-table** command to view EVPN route information.

---End

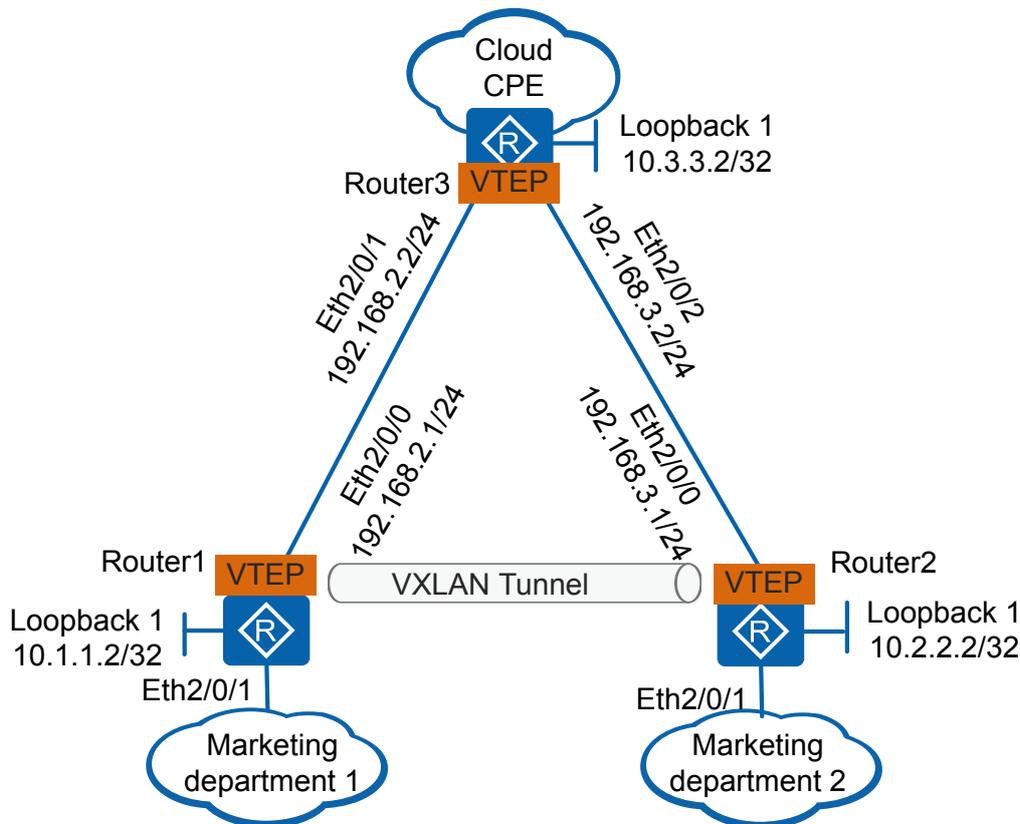
14.7 Configuration Examples for VXLANs

14.7.1 Example for Configuring Communication Within a Network Segment Through a VXLAN Tunnel

Networking Requirements

In [Figure 14-18](#), an enterprise has two departments scattered in different geographical locations. As the two departments have the same service requirements, they are planned in the same network segment. End users in both departments belong to VLAN 10. They need to communicate over the VXLAN tunnel.

Figure 14-18 Configuring communication within a network segment through a VXLAN tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Router1, Router2, and Router3 to ensure Layer 3 connectivity.
2. Configure a deployment mode for the VXLAN access service on Router1 and Router2.
3. Configure information for VXLAN tunnel establishment on Router1 and Router2.

Procedure

Step 1 Configure a routing protocol.

Configure IP addresses for interfaces on Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here. When OSPF is used, the 32-bit loopback address of each router must be advertised.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface loopback 1
[Router1-LoopBack1] ip address 10.1.1.2 32
[Router1-LoopBack1] quit
[Router1] interface ethernet 2/0/0
[Router1-Ethernet2/0/0] undo portswitch
[Router1-Ethernet2/0/0] ip address 192.168.2.1 24
[Router1-Ethernet2/0/0] quit
[Router1] ospf
```

```
[Router1-ospf-1] area 0
[Router1-ospf-1-area-0.0.0.0] network 10.1.1.2 0.0.0.0
[Router1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Router1-ospf-1-area-0.0.0.0] quit
[Router1-ospf-1] quit
```

After OSPF is configured, the Routers can learn the loopback interface address of each other and successfully ping each other. The following shows the ping result from Router1 to Router2.

```
[Router1] ping 10.2.2.2
PING 10.2.2.2: 56 data bytes, press CTRL_C to break
  Reply from 10.2.2.2: bytes=56 Sequence=1 ttl=255 time=240 ms
  Reply from 10.2.2.2: bytes=56 Sequence=2 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=4 ttl=255 time=14 ms
  Reply from 10.2.2.2: bytes=56 Sequence=5 ttl=255 time=5 ms

--- 10.2.2.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 5/53/240 ms
```

Step 2 Configure a service access point on Router1 and Router2, respectively.

Configure Router1.

```
[Router1] bridge-domain 10
[Router1-bd10] quit
[Router1] interface ethernet 2/0/1.1 mode l2
[Router1-Ethernet2/0/1.1] encapsulation dot1q vid 10
[Router1-Ethernet2/0/1.1] bridge-domain 10
[Router1-Ethernet2/0/1.1] quit
```

Configure Router2.

```
[Router2] bridge-domain 10
[Router2-bd10] quit
[Router2] interface ethernet 2/0/1.1 mode l2
[Router2-Ethernet2/0/1.1] encapsulation dot1q vid 10
[Router2-Ethernet2/0/1.1] bridge-domain 10
[Router2-Ethernet2/0/1.1] quit
```

Step 3 Configure information for VXLAN tunnel establishment on Router1 and Router2.

Configure Router1.

```
[Router1] bridge-domain 10
[Router1-bd10] vxlan vni 2010
[Router1-bd10] quit
[Router1] interface nve 1
[Router1-Nve1] source 10.1.1.2
[Router1-Nve1] vni 2010 head-end peer-list 10.2.2.2
[Router1-Nve1] quit
```

Configure Router2.

```
[Router2] bridge-domain 10
[Router2-bd10] vxlan vni 2010
[Router2-bd10] quit
[Router2] interface nve 1
[Router2-Nve1] source 10.2.2.2
[Router2-Nve1] vni 2010 head-end peer-list 10.1.1.2
[Router2-Nve1] quit
```

Step 4 Verify the configuration.

After the configuration is complete, run the **display vxlan vni** command on Router1 and Router2. The command output shows that the VNI status is up. Run the **display vxlan tunnel**

command, and you can see VXLAN tunnel information. The command output on Router1 is used as an example.

```
[Router1] display vxlan vni
VNI          BD-ID          State
-----
2010         10             up
-----
Number of vxlan vni bound to BD is : 2

VNI          VRF-ID
-----
-----
Number of vxlan vni bound to VPN is : 0

[Router1] display vxlan tunnel
Tunnel ID    Source          Destination     State    Type
-----
4026531841   10.1.1.2       10.2.2.2       up      static
-----
Number of vxlan tunnel : 1
```

After the configuration is complete, users in the same network segment can communicate over a VXLAN tunnel.

---End

Configuration Files

- Router1 configuration file

```
#
sysname Router1
#
bridge-domain
10
  vxlan vni 2010
#

interface
Ethernet2/0/0
  undo
  portswitch
  ip address 192.168.2.1
255.255.255.0
#

interface Ethernet2/0/1.1 mode
l2
  encapsulation dot1q vid
10
  bridge-domain 10
#

interface
LoopBack1
  ip address 10.1.1.2 255.255.255.255
#

interface
Nve1
  source
10.1.1.2
  vni 2010 head-end peer-list
10.2.2.2
#
ospf
1
```

```
area
0.0.0.0
 network 10.1.1.2
0.0.0.0
 network 192.168.2.0 0.0.0.255
#
return
```

- Router2 configuration file

```
#
sysname Router2
#
bridge-domain
10
 vxlan vni 2010
#
interface
Ethernet2/0/0
 undo
 portswitch
 ip address 192.168.3.1
255.255.255.0
#
interface Ethernet2/0/1.1 mode
12
 encapsulation dot1q vid
10
 bridge-domain 10
#
interface
LoopBack1
 ip address 10.2.2.2 255.255.255.255
#
interface
Nve1
 source
10.2.2.2
 vni 2010 head-end peer-list
10.1.1.2
#
ospf
1
 area
0.0.0.0
 network 10.2.2.2
0.0.0.0
 network 192.168.3.0 0.0.0.255
#
return
```

- Router3 configuration file

```
#
sysname Router3
#
interface
Ethernet2/0/1
 undo
 portswitch
 ip address 192.168.2.2
255.255.255.0
#
interface
```

```
Ethernet2/0/2
 undo
 portswitch
 ip address 192.168.3.2
 255.255.255.0
 #

 interface
 LoopBack1
 ip address 10.3.3.2 255.255.255.255
 #
 ospf
 1
 area
 0.0.0.0
 network 10.3.3.2
 0.0.0.0
 network 192.168.2.0 0.0.0.255
 network 192.168.3.0 0.0.0.255
 #

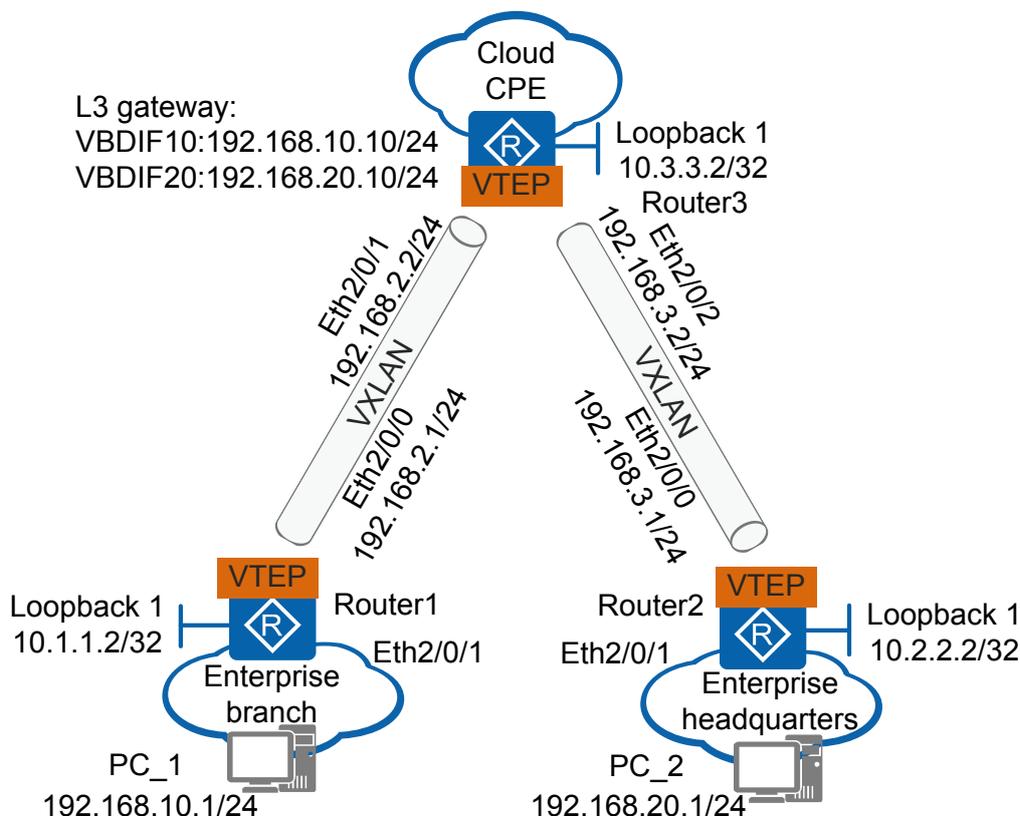
 return
```

14.7.2 Example for Configuring a Layer 3 VXLAN Gateway to Enable Communication Between Users in Different Network Segments

Networking Requirements

In [Figure 14-19](#), Router1 and Router2 are the branch and headquarters gateways of an enterprise. As users in the headquarters and branch have different service requirements, they are planned in different network segments. PC_1 in the branch and PC_2 in the headquarters belong to VLAN 10 and VLAN 20, respectively. The enterprise requires that users in the headquarters and branch can communicate using a Layer 3 VXLAN gateway.

Figure 14-19 Configuring a Layer 3 VXLAN gateway to enable communication between users in different network segments



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Router1, Router2, and Router3 to ensure Layer 3 network connectivity.
2. Configure a deployment mode for the VXLAN access service on Router1 and Router2.
3. Configure information for VXLAN tunnel establishment on Router1, Router2, and Router3.
4. Configure a Layer 3 gateway on Router3.

Procedure

Step 1 Configure a routing protocol.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here. When OSPF is used, the 32-bit loopback address of each router must be advertised.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface loopback 1
[Router1-LoopBack1] ip address 10.1.1.2 32
[Router1-LoopBack1] quit
[Router1] interface ethernet 2/0/0
[Router1-Ethernet2/0/0] undo portswitch
```

```
[Router1-Ethernet2/0/0] ip address 192.168.2.1 24
[Router1-Ethernet2/0/0] quit
[Router1] ospf
[Router1-ospf-1] area 0
[Router1-ospf-1-area-0.0.0.0] network 10.1.1.2 0.0.0.0
[Router1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Router1-ospf-1-area-0.0.0.0] quit
[Router1-ospf-1] quit
```

After OSPF is configured, the routers can learn the loopback interface address of each other and successfully ping each other. The following shows the ping result from Router1 to Router2.

```
[Router1] ping 10.2.2.2
PING 10.2.2.2: 56 data bytes, press CTRL_C to break
Reply from 10.2.2.2: bytes=56 Sequence=1 ttl=255 time=1 ms
Reply from 10.2.2.2: bytes=56 Sequence=2 ttl=255 time=5 ms
Reply from 10.2.2.2: bytes=56 Sequence=3 ttl=255 time=5 ms
Reply from 10.2.2.2: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 10.2.2.2: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.2.2.2 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/3/5 ms
```

Step 2 Configure a service access point on Router1 and Router2, respectively.

Configure Router1.

```
[Router1] bridge-domain 10
[Router1-bd10] quit
[Router1] interface ethernet 2/0/1.1 mode l2
[Router1-Ethernet2/0/1.1] encapsulation dot1q vid 10
[Router1-Ethernet2/0/1.1] bridge-domain 10
[Router1-Ethernet2/0/1.1] quit
```

Configure Router2.

```
[Router2] bridge-domain 20
[Router2-bd20] quit
[Router2] interface ethernet 2/0/1.1 mode l2
[Router2-Ethernet2/0/1.1] encapsulation dot1q vid 20
[Router2-Ethernet2/0/1.1] bridge-domain 20
[Router2-Ethernet2/0/1.1] quit
```

Step 3 Configure information for VXLAN tunnel establishment on Router1, Router2, and Router3.

Configure Router1.

```
[Router1] bridge-domain 10
[Router1-bd10] vxlan vni 2010
[Router1-bd10] quit
[Router1] interface nve 1
[Router1-Nve1] source 10.1.1.2
[Router1-Nve1] vni 2010 head-end peer-list 10.3.3.2
[Router1-Nve1] quit
```

Configure Router2.

```
[Router2] bridge-domain 20
[Router2-bd20] vxlan vni 2020
[Router2-bd20] quit
[Router2] interface nve 1
[Router2-Nve1] source 10.2.2.2
[Router2-Nve1] vni 2020 head-end peer-list 10.3.3.2
[Router2-Nve1] quit
```

Configure Router3.

```
[Router3] bridge-domain 10
[Router3-bd10] vxlan vni 2010
[Router3-bd10] quit
[Router3] interface nve 1
[Router3-Nve1] source 10.3.3.2
[Router3-Nve1] vni 2010 head-end peer-list 10.1.1.2
[Router3-Nve1] quit
[Router3] bridge-domain 20
[Router3-bd20] vxlan vni 2020
[Router3-bd20] quit
[Router3] interface nve 1
[Router3-Nve1] source 10.3.3.2
[Router3-Nve1] vni 2020 head-end peer-list 10.2.2.2
[Router3-Nve1] quit
```

Step 4 Configure a Layer 3 VXLAN gateway on Router3.

```
[Router3] interface vbdif 10
[Router3-Vbdif10] ip address 192.168.10.10 24
[Router3-Vbdif10] quit
[Router3] interface vbdif 20
[Router3-Vbdif20] ip address 192.168.20.10 24
[Router3-Vbdif20] quit
```

Step 5 Verify the configuration.

After the preceding configuration, run the **display vxlan vni** and **display vxlan tunnel** commands on Router1, Router2, and Router3. You can find that the VNI status is Up and VXLAN tunnel information is displayed. The command output on Router3 is used as an example.

```
[Router3] display vxlan vni
VNI          BD-ID          State
-----
2010         10             up
2020         20             up
-----
Number of vxlan vni bound to BD is : 2

VNI          VRF-ID
-----
Number of vxlan vni bound to VPN is : 0

[Router3] display vxlan tunnel
Tunnel ID    Source          Destination     State    Type
-----
4026531842   10.3.3.2        10.1.1.2        up       static
4026531841   10.3.3.2        10.2.2.2        up       static
-----
Number of vxlan tunnel : 2
```

---End

Configuration Files

- Router1 configuration file

```
#
sysname Router1
#
bridge-domain
10
vxlan vni 2010
#
interface
```

```
Ethernet2/0/0
 undo
 portswitch
 ip address 192.168.2.1
 255.255.255.0
 #

interface Ethernet2/0/1.1 mode
 l2
 encapsulation dot1q vid
 10
 bridge-domain 10
 #

interface
 LoopBack1
 ip address 10.1.1.2 255.255.255.255
 #

interface
 Nve1
 source
 10.1.1.2
 vni 2010 head-end peer-list
 10.3.3.2
 #
 ospf
 1
 area
 0.0.0.0
 network 10.1.1.2
 0.0.0.0
 network 192.168.2.0 0.0.0.255
 #

return
```

● Router2 configuration file

```
#
 sysname Router2
 #
 bridge-domain
 20
 vxlan vni 2020
 #

interface
 Ethernet2/0/0
 undo
 portswitch
 ip address 192.168.3.1
 255.255.255.0
 #

interface Ethernet2/0/1.1 mode
 l2
 encapsulation dot1q vid
 20
 bridge-domain 20
 #

interface
 LoopBack1
 ip address 10.2.2.2 255.255.255.255
 #

interface
 Nve1
 source
 10.2.2.2
```

```
vni 2020 head-end peer-list
10.3.3.2
#
ospf
1
area
0.0.0.0
network 10.2.2.2
0.0.0.0
network 192.168.3.0 0.0.0.255
#
return
```

● Router3 configuration file

```
#
sysname Router3
#
bridge-domain
10
vxlan vni 2010
bridge-domain
20
vxlan vni 2020
#
interface Ethernet2/0/1
undo
portswitch
ip address 192.168.2.2
255.255.255.0
#
interface Ethernet2/0/2
undo
portswitch
ip address 192.168.3.2
255.255.255.0
#
interface
LoopBack1
ip address 10.3.3.2 255.255.255.255
#
interface
Vbdf10
ip address 192.168.10.10
255.255.255.0
#
interface
Vbdf20
ip address 192.168.20.10 255.255.255.0
#
interface
Nve1
source
10.3.3.2
vni 2010 head-end peer-list
10.1.1.2
vni 2020 head-end peer-list
10.2.2.2
#
ospf
1
area
0.0.0.0
network 10.3.3.2
```

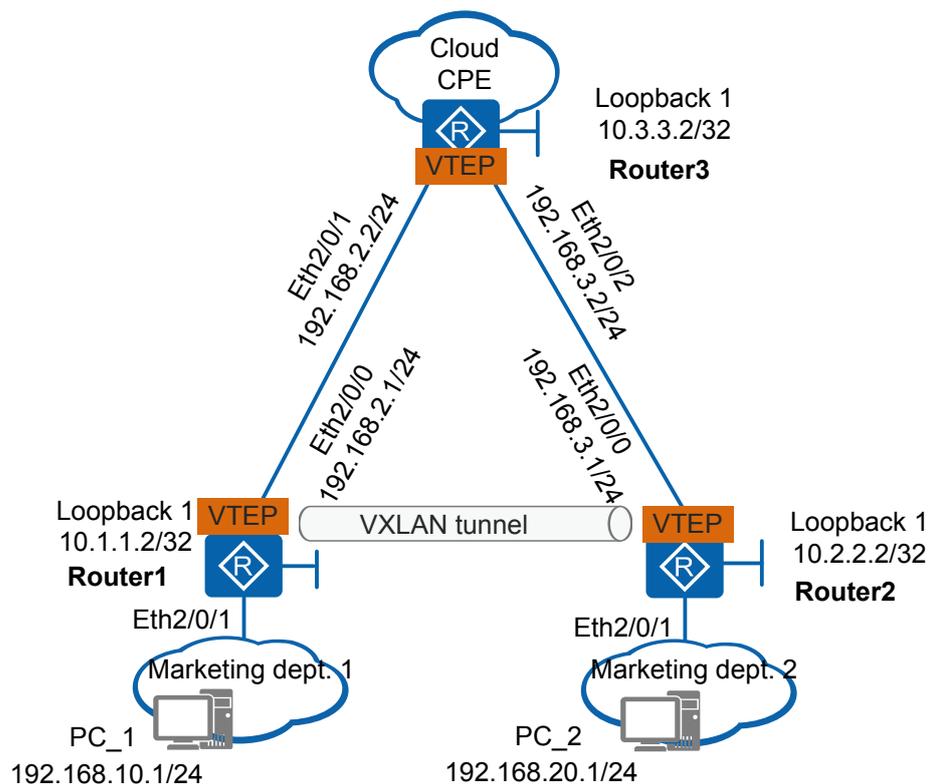
```
0.0.0.0
network 192.168.2.0 0.0.0.255
network 192.168.3.0 0.0.0.255
#
return
```

14.7.3 Example for Dynamically Establishing a VXLAN Tunnel in BGP EVPN Mode to Implement Communication Between Users in Different Network Segments

Networking Requirements

In **Figure 14-20**, Router1 and Router2 are the branch and headquarters gateways of an enterprise. As users in the headquarters and branch have different service requirements, they are planned in different network segments. PC_1 in the branch and PC_2 in the headquarters belong to VLAN 10 and VLAN 20, respectively. The enterprise requires that users in the headquarters and branch can communicate over a VXLAN tunnel dynamically established using BGP EVPN.

Figure 14-20 Configuring communication between different network segments through a Layer 3 VXLAN gateway



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Router1, Router2, and Router3 to ensure Layer 3 network connectivity.
2. Configure a deployment mode for the VXLAN access service on Router1 and Router2.
3. Establish a BGP EVPN peer relationship.
4. Configure an IP address for the source VTEP on Router1 and Router2.
5. Configure a VPN instance on Router1 and Router2.
6. Configure a Layer 3 gateway on Router1 and Router2.
7. Configure Router1, Router2, and Router3 to advertise IP prefix routes to the BGP peer.

Procedure

Step 1 Configure a routing protocol.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here. When OSPF is used, the 32-bit loopback address of each router must be advertised.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface loopback 1
[Router1-LoopBack1] ip address 10.1.1.2 32
[Router1-LoopBack1] quit
[Router1] interface ethernet 2/0/0
[Router1-Ethernet2/0/0] undo portswitch
[Router1-Ethernet2/0/0] ip address 192.168.2.1 24
[Router1-Ethernet2/0/0] quit
[Router1] ospf
[Router1-ospf-1] area 0
[Router1-ospf-1-area-0.0.0.0] network 10.1.1.2 0.0.0.0
[Router1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Router1-ospf-1-area-0.0.0.0] quit
[Router1-ospf-1] quit
```

After OSPF is configured, the routers can learn the loopback interface address of each other and successfully ping each other. The following shows the ping result from Router1 to Router2.

```
[Router1] ping 10.2.2.2
PING 10.2.2.2: 56 data bytes, press CTRL_C to break
  Reply from 10.2.2.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.2.2.2: bytes=56 Sequence=2 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=3 ttl=255 time=5 ms
  Reply from 10.2.2.2: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.2.2.2: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.2.2.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/3/5 ms
```

Step 2 Configure a service access point on Router1 and Router2, respectively.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] bridge-domain 10
[Router1-bd10] quit
[Router1] interface ethernet 2/0/1.1 mode 12
[Router1-Ethernet2/0/1.1] encapsulation dot1q vid 10
```

```
[Router1-Ethernet2/0/1.1] bridge-domain 10  
[Router1-Ethernet2/0/1.1] quit
```

Step 3 Establish a BGP EVPN peer relationship.

Establish a BGP EVPN peer relationship on Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] bgp 100  
[Router1-bgp] peer 10.3.3.2 as-number 100  
[Router1-bgp] peer 10.3.3.2 connect-interface LoopBack1  
[Router1-bgp] l2vpn-family evpn  
[Router1-bgp-af-evpn] peer 10.3.3.2 enable  
[Router1-bgp-af-evpn] quit  
[Router1-bgp] quit  
[Router1] interface nve 1  
[Router1-Nve1] source 10.1.1.2  
[Router1-Nve1] quit
```

On Router3, establish a BGP EVPN peer relationship with Router1 and Router2.

```
[Router3] bgp 100  
[Router3-bgp] peer 10.1.1.2 as-number 100  
[Router3-bgp] peer 10.1.1.2 connect-interface LoopBack1  
[Router3-bgp] peer 10.2.2.2 as-number 100  
[Router3-bgp] peer 10.2.2.2 connect-interface LoopBack1  
[Router3-bgp] l2vpn-family evpn  
[Router3-bgp-af-evpn] peer 10.1.1.2 enable  
[Router3-bgp-af-evpn] peer 10.2.2.2 enable  
[Router3-bgp-af-evpn] quit  
[Router3-bgp] quit
```

Step 4 Configure a VPN instance on Router1 and Router2.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] ip vpn-instance vpn1  
[Router1-vpn-instance-vpn1] ipv4-family  
[Router1-vpn-instance-vpn1-af-ipv4] route-distinguisher 100:1  
[Router1-vpn-instance-vpn1-af-ipv4] vpn-target 1:1 evpn  
[Router1-vpn-instance-vpn1-af-ipv4] quit  
[Router1-vpn-instance-vpn1] vxlan vni 5010  
[Router1-vpn-instance-vpn1] quit  
[Router1] bridge-domain 10  
[Router1-bd10] vxlan vni 2010  
[Router1-bd10] quit
```

Step 5 Configure a Layer 3 VXLAN gateway on Router1 and Router2 and bind the VPN instance to the gateway.

Configure Router1. The configuration of Router2 is similar to the configuration of Router1, and is not mentioned here.

```
[Router1] interface vbdif 10  
[Router1-Vbdif10] ip binding vpn-instance vpn1  
[Router1-Vbdif10] ip address 192.168.10.10 24  
[Router1-Vbdif10] quit
```

Step 6 Configure Router1, Router2, and Router3 to advertise IP prefix routes to the BGP peer.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here.

```
[Router1] bgp 100  
[Router1-bgp] ipv4-family vpn-instance vpn1  
[Router1-bgp-vpn1] import-route direct  
[Router1-bgp-vpn1] advertise l2vpn evpn
```

```
[Router1-bgp-vpn1] quit  
[Router1-bgp] quit
```

Step 7 Verify the configuration.

After the configuration is complete, run the **display vxlan tunnel** command on Router1, Router2, and Router3. You can view VXLAN tunnel information. The command output on Router3 is used as an example.

```
[Router3] display vxlan tunnel  
Tunnel ID      Source          Destination     State    Type  
-----  
4026531842     10.3.3.2       10.1.1.2       up       dynamic  
4026531841     10.3.3.2       10.2.2.2       up       dynamic  
-----  
Number of vxlan tunnel : 2
```

---End

Configuration Files

- Router1 configuration file

```
#  
sysname Router1  
#  
  
ip vpn-instance  
vpn1  
  ipv4-  
  family  
    route-distinguisher  
    100:1  
    vpn-target 1:1 export-extcommunity  
  evpn  
    vpn-target 1:1 import-extcommunity  
  evpn  
    vxlan vni  
    5010  
  #  
  bridge-domain  
  10  
    vxlan vni 2010  
  #  
  
interface  
Ethernet2/0/0  
  undo  
  portswitch  
  ip address 192.168.2.1  
  255.255.255.0  
  #  
  
interface Ethernet2/0/1.1 mode  
l2  
  encapsulation dot1q vid  
  10  
  bridge-domain 10  
  #  
  
interface  
LoopBack1  
  ip address 10.1.1.2 255.255.255.255  
  #  
  
interface  
Vbdf10  
  ip binding vpn-instance  
  vpn1
```

```
ip address 192.168.10.10
255.255.255.0
#

interface
Nve1
 source
10.1.1.2
#

bgp
100
 peer 10.3.3.2 as-number
100
 peer 10.3.3.2 connect-interface
LoopBack1
#

 ipv4-family
unicast
 undo
synchronization
 peer 10.3.3.2
enable
#

 l2vpn-family
evpn
 policy vpn-
target
 peer 10.3.3.2
enable
#

 ipv4-family vpn-instance
vpn1
 import-route
direct
 advertise l2vpn
evpn
#
ospf
1
 area
0.0.0.0
 network 10.1.1.2
0.0.0.0
 network 192.168.2.0 0.0.0.255
#

return
```

● Router2 configuration file

```
#
sysname Router2
#

ip vpn-instance
vpn1
 ipv4-
family
 route-distinguisher
100:1
 vpn-target 1:1 export-extcommunity
evpn
 vpn-target 1:1 import-extcommunity
evpn
```

```
vxlan vni
5020
#
bridge-domain
20
  vxlan vni 2020
#

interface
Ethernet2/0/0
  undo
  portswitch
  ip address 192.168.3.1
  255.255.255.0
#

interface Ethernet2/0/1.1 mode
12
  encapsulation dot1q vid
20
  bridge-domain 20
#

interface
LoopBack1
  ip address 10.2.2.2 255.255.255.255
#

interface
Vbdif20
  ip binding vpn-instance
vpn1
  ip address 192.168.20.10
  255.255.255.0
#

interface
Nve1
  source
10.2.2.2
#

bgp
100
  peer 10.3.3.2 as-number
100
  peer 10.3.3.2 connect-interface
LoopBack1
#

  ipv4-family
unicast
  undo
synchronization
  peer 10.3.3.2
enable
#

  l2vpn-family
evpn
  policy vpn-
target
  peer 10.3.3.2
enable
#
```

```
ipv4-family vpn-instance
vpn1
import-route
direct
advertise l2vpn
evpn
#
ospf
1
area
0.0.0.0
network 10.2.2.2
0.0.0.0
network 192.168.3.0 0.0.0.255
#
return
```

● Router3 configuration file

```
#
sysname Router3
#
interface Ethernet2/0/1
undo
portswitch
ip address 192.168.2.2
255.255.255.0
#
interface Ethernet2/0/2
undo
portswitch
ip address 192.168.3.2
255.255.255.0
#
interface
LoopBack1
ip address 10.3.3.2 255.255.255.255
#
bgp
100
peer 10.1.1.2 as-number
100
peer 10.1.1.2 connect-interface
LoopBack1
peer 10.2.2.2 as-number
100
peer 10.2.2.2 connect-interface
LoopBack1
#
ipv4-family
unicast
undo
synchronization
peer 10.1.1.2
enable
peer 10.2.2.2
enable
#
l2vpn-family
evpn
policy vpn-
target
```

```
peer 10.1.1.2
enable
peer 10.2.2.2
enable

#

ipv4-family vpn-instance
vpn1
import-route
direct
advertise l2vpn
evpn
#
ospf
1
area
0.0.0.0
network 10.3.3.2
0.0.0.0
network 192.168.2.0 0.0.0.255
network 192.168.3.0 0.0.0.255
#
return
```

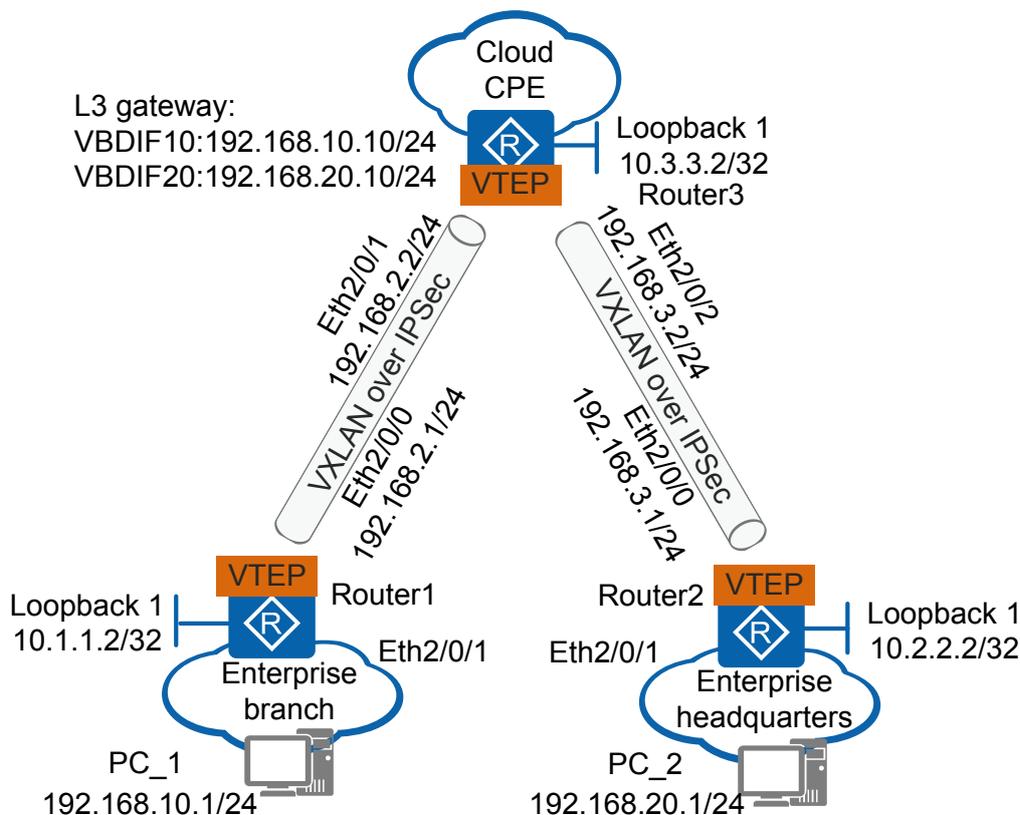
14.7.4 Example for Configuring the Headquarters and Branch to Communicate Using VXLAN over IPsec Tunnels

Networking Requirements

In [Figure 14-21](#), Router1 and Router2 are the branch and headquarters gateways of an enterprise. VXLAN tunnels are established to enable communication between the headquarters and branch.

The enterprise requires that services transmitted over VXLAN tunnels be protected by IPsec to prevent eavesdropping and tampering. To meet this requirement, VXLAN over IPsec can be configured to encrypt service packets transmitted between the headquarters and branch.

Figure 14-21 Configuring the headquarters and branch to communicate using VXLAN over IPsec tunnels



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Router1, Router2, and Router3 to ensure Layer 3 network connectivity.
2. Configure a deployment mode for the VXLAN access service on Router1 and Router2.
3. Configure information for VXLAN tunnel establishment on Router1, Router2, and Router3.
4. Configure a Layer 3 gateway on Router3.
5. Configure ACLs on Router1, Router2, and Router3 to define the data flows to be protected by IPsec.
6. Configure an IPsec proposal on Router1, Router2, and Router3 to define the traffic protection method.
7. Configure an IKE peer on Router1, Router2, and Router3 and define the attributes used for IKE negotiation.
8. Configure IPsec policies on Router1, Router2, and Router3 and apply the ACLs, IPsec proposal, and IKE peer to define the data flows to be protected and protection method.
9. Apply the IPsec policy groups to the interfaces on Router1, Router2, and Router3 to enable the IPsec protection function on the interfaces.

Procedure

Step 1 Configure a routing protocol.

Configure Router1. The configurations of Router2 and Router3 are similar to the configuration of Router1, and are not mentioned here. When OSPF is used, the 32-bit loopback address of each router must be advertised.

```
<Huawei> system-view
[Huawei] sysname Router1
[Router1] interface loopback 1
[Router1-LoopBack1] ip address 10.1.1.2 32
[Router1-LoopBack1] quit
[Router1] interface ethernet 2/0/0
[Router1-Ethernet2/0/0] undo portswitch
[Router1-Ethernet2/0/0] ip address 192.168.2.1 24
[Router1-Ethernet2/0/0] quit
[Router1] ospf
[Router1-ospf-1] area 0
[Router1-ospf-1-area-0.0.0.0] network 10.1.1.2 0.0.0.0
[Router1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[Router1-ospf-1-area-0.0.0.0] quit
[Router1-ospf-1] quit
```

After OSPF is configured, the routers can learn the loopback interface address of each other and successfully ping each other. The following shows the ping result from Router1 to Router3.

```
[Router1] ping 10.3.3.2
PING 10.3.3.2: 56 data bytes, press CTRL_C to break
  Reply from 10.3.3.2: bytes=56 Sequence=1 ttl=255 time=1 ms
  Reply from 10.3.3.2: bytes=56 Sequence=2 ttl=255 time=1 ms
  Reply from 10.3.3.2: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 10.3.3.2: bytes=56 Sequence=4 ttl=255 time=1 ms
  Reply from 10.3.3.2: bytes=56 Sequence=5 ttl=255 time=44 ms

--- 10.3.3.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/9/44 ms
```

Step 2 Configure a service access point on Router1.

Configure a service access point on Router1. The configuration on Router2 is similar to that on Router1, and is not mentioned here.

```
[Router1] bridge-domain 10
[Router1-bd10] quit
[Router1] interface ethernet 2/0/1.1 mode l2
[Router1-Ethernet2/0/1.1] encapsulation dot1q vid 10
[Router1-Ethernet2/0/1.1] bridge-domain 10
[Router1-Ethernet2/0/1.1] quit
```

Step 3 Configure information for VXLAN tunnel establishment on Router1, Router2, and Router3.

Configure Router1.

```
[Router1] bridge-domain 10
[Router1-bd10] vxlan vni 2010
[Router1-bd10] quit
[Router1] interface nve 1
[Router1-Nve1] source 10.1.1.2
[Router1-Nve1] vni 2010 head-end peer-list 10.3.3.2
[Router1-Nve1] quit
```

Configure Router2.

```
[Router2] bridge-domain 20
[Router2-bd20] vxlan vni 2020
[Router2-bd20] quit
[Router2] interface nve 1
[Router2-Nve1] source 10.2.2.2
[Router2-Nve1] vni 2020 head-end peer-list 10.3.3.2
[Router2-Nve1] quit
```

Configure Router3.

```
[Router3] bridge-domain 10
[Router3-bd10] vxlan vni 2010
[Router3-bd10] quit
[Router3] interface nve 1
[Router3-Nve1] source 10.3.3.2
[Router3-Nve1] vni 2010 head-end peer-list 10.1.1.2
[Router3-Nve1] quit
[Router3] bridge-domain 20
[Router3-bd20] vxlan vni 2020
[Router3-bd20] quit
[Router3] interface nve 1
[Router3-Nve1] source 10.3.3.2
[Router3-Nve1] vni 2020 head-end peer-list 10.2.2.2
[Router3-Nve1] quit
```

Step 4 Configure a Layer 3 VXLAN gateway on Router3.

```
[Router3] interface vbdif 10
[Router3-Vbdif10] ip address 192.168.10.10 24
[Router3-Vbdif10] quit
[Router3] interface vbdif 20
[Router3-Vbdif20] ip address 192.168.20.10 24
[Router3-Vbdif20] quit
```

Step 5 Configure ACLs on Router1, Router2, and Router3 to define the data flows to be protected by IPsec.

Configure an ACL on Router1.

```
[Router1] acl number 3000
[Router1-acl-adv-3000] rule permit ip source 10.1.1.2 0.0.0.0 destination
10.3.3.2 0.0.0.0
[Router1-acl-adv-3000] quit
```

Configure an ACL on Router2.

```
[Router2] acl number 3001
[Router2-acl-adv-3001] rule permit ip source 10.2.2.2 0.0.0.0 destination
10.3.3.2 0.0.0.0
[Router2-acl-adv-3001] quit
```

Configure two ACLs on Router3.

```
[Router3] acl number 3000
[Router3-acl-adv-3000] rule permit ip source 10.3.3.2 0.0.0.0 destination
10.1.1.2 0.0.0.0
[Router3-acl-adv-3000] quit
[Router3] acl number 3001
[Router3-acl-adv-3001] rule permit ip source 10.3.3.2 0.0.0.0 destination
10.2.2.2 0.0.0.0
[Router3-acl-adv-3001] quit
```

Step 6 Configure an IPsec proposal on Router1, Router2, and Router3.

Configure an IPsec proposal on Router1.

```
[Router1] ipsec proposal s1
[Router1-ipsec-proposal-s1] esp authentication-algorithm sha2-256
[Router1-ipsec-proposal-s1] esp encryption-algorithm aes-256
[Router1-ipsec-proposal-s1] quit
```

Configure an IPsec proposal on Router2.

```
[Router2] ipsec proposal s1
[Router2-ipsec-proposal-s1] esp authentication-algorithm sha2-256
[Router2-ipsec-proposal-s1] esp encryption-algorithm aes-256
[Router2-ipsec-proposal-s1] quit
```

Configure an IPsec proposal on Router3.

```
[Router3] ipsec proposal s1
[Router3-ipsec-proposal-s1] esp authentication-algorithm sha2-256
[Router3-ipsec-proposal-s1] esp encryption-algorithm aes-256
[Router3-ipsec-proposal-s1] quit
```

Step 7 Create an IKE peer on Router1, Router2, and Router3.

Configure an IKE proposal on Router1.

```
[Router1] ike proposal 1
[Router1-ike-proposal-1] encryption-algorithm aes-256
[Router1-ike-proposal-1] authentication-algorithm sha2-256
[Router1-ike-proposal-1] dh group2
[Router1-ike-proposal-1] quit
```

Create an IKE peer on Router1 and configure the pre-shared key and remote ID based on default settings.

```
[Router1] ike peer 23
[Router1-ike-peer-23] ike-proposal 1
[Router1-ike-peer-23] pre-shared-key cipher Huawei@123
[Router1-ike-peer-23] remote-address 192.168.2.2
[Router1-ike-peer-23] quit
```

Configure an IKE proposal on Router2.

```
[Router2] ike proposal 1
[Router2-ike-proposal-1] encryption-algorithm aes-256
[Router2-ike-proposal-1] authentication-algorithm sha2-256
[Router2-ike-proposal-1] dh group2
[Router2-ike-proposal-1] quit
```

Create an IKE peer on Router2 and configure the pre-shared key and remote ID based on default settings.

```
[Router2] ike peer 24
[Router2-ike-peer-24] ike-proposal 1
[Router2-ike-peer-24] pre-shared-key cipher Huawei@123
[Router2-ike-peer-24] remote-address 192.168.3.2
[Router2-ike-peer-24] quit
```

Configure an IKE proposal on Router3.

```
[Router3] ike proposal 1
[Router3-ike-proposal-1] encryption-algorithm aes-256
[Router3-ike-proposal-1] authentication-algorithm sha2-256
[Router3-ike-proposal-1] dh group2
[Router3-ike-proposal-1] quit
```

Create an IKE peer on Router3 and configure the pre-shared key and remote ID based on default settings.

```
[Router3] ike peer 21
[Router3-ike-peer-21] ike-proposal 1
[Router3-ike-peer-21] pre-shared-key cipher Huawei@123
[Router3-ike-peer-21] remote-address 192.168.2.1
[Router3-ike-peer-21] quit
[Router3] ike peer 22
[Router3-ike-peer-22] ike-proposal 1
[Router3-ike-peer-22] pre-shared-key cipher Huawei@123
```

```
[Router3-ike-peer-22] remote-address 192.168.3.1  
[Router3-ike-peer-22] quit
```

Step 8 Create IPsec policies on Router1, Router2, and Router3.

Create an IPsec policy on Router1.

```
[Router1] ipsec policy map1 2 isakmp  
[Router1-ipsec-policy-isakmp-map1-2] ike-peer 23  
[Router1-ipsec-policy-isakmp-map1-2] proposal s1  
[Router1-ipsec-policy-isakmp-map1-2] security acl 3000  
[Router1-ipsec-policy-isakmp-map1-2] quit
```

Create an IPsec policy on Router2.

```
[Router2] ipsec policy user1 2 isakmp  
[Router2-ipsec-policy-isakmp-user1-2] ike-peer 24  
[Router2-ipsec-policy-isakmp-user1-2] proposal s1  
[Router2-ipsec-policy-isakmp-user1-2] security acl 3001  
[Router2-ipsec-policy-isakmp-user1-2] quit
```

Create an IPsec policy on Router3.

```
[Router3] ipsec policy map1 2 isakmp  
[Router3-ipsec-policy-isakmp-map1-2] ike-peer 21  
[Router3-ipsec-policy-isakmp-map1-2] proposal s1  
[Router3-ipsec-policy-isakmp-map1-2] security acl 3000  
[Router3-ipsec-policy-isakmp-map1-2] quit  
[Router3] ipsec policy user1 2 isakmp  
[Router3-ipsec-policy-isakmp-user1-2] ike-peer 22  
[Router3-ipsec-policy-isakmp-user1-2] proposal s1  
[Router3-ipsec-policy-isakmp-user1-2] security acl 3001  
[Router3-ipsec-policy-isakmp-user1-2] quit
```

Step 9 Apply the IPsec policy groups to the interfaces on Router1, Router2, and Router3.

Apply the IPsec policy to an interface on Router1.

```
[Router1] interface ethernet 2/0/0  
[Router1-Ethernet2/0/0] ipsec policy map1  
[Router1-Ethernet2/0/0] quit
```

Apply the IPsec policy to an interface on Router2.

```
[Router2] interface ethernet 2/0/0  
[Router2-Ethernet2/0/0] ipsec policy user1  
[Router2-Ethernet2/0/0] quit
```

Apply the IPsec policies to the interfaces on Router3.

```
[Router3] interface ethernet 2/0/1  
[Router3-Ethernet2/0/1] ipsec policy map1  
[Router3-Ethernet2/0/1] quit  
[Router3] interface ethernet 2/0/2  
[Router3-Ethernet2/0/2] ipsec policy user1  
[Router3-Ethernet2/0/2] quit
```

Step 10 Verify the configuration.

After the preceding configuration, PC_1 can still ping PC_2 and the data transmitted between them is encrypted.

Run the **display ike sa** command on Router3. You can find the established IKE SAs.

```
[Router3] display ike sa
```

Conn-ID	Peer	VPN	Flag(s)	Phase
2189	192.168.2.1:500		RD A	v2:2
2188	192.168.2.1:500		RD A	v2:1

```

2183      192.168.3.1:500      RD|A      v2:2
2178      192.168.3.1:500      RD|A      v2:1

Number of IKE SA : 4
-----

Flag Description:
RD--READY  ST--STAYALIVE  RL--REPLACED  FD--FADING  TO--TIMEOUT
HRT--HEARTBEAT  LKG--LAST KNOWN GOOD SEQ NO.  BCK--BACKED UP
M--ACTIVE  S--STANDBY  A--ALONE  NEG--NEGOTIATING
  
```

Run the **display vxlan vni** and **display vxlan tunnel** commands on Router1, Router2, and Router3. You can find that the VNI status is Up and VXLAN tunnel information is displayed. The command output on Router3 is used as an example.

```

[Router3] display vxlan vni
VNI          BD-ID          State
-----
2010         10             up
2020         20             up
-----
Number of vxlan vni bound to BD is : 2

VNI          VRF-ID
-----
-----
Number of vxlan vni bound to VPN is : 0

[Router3] display vxlan tunnel
Tunnel ID    Source          Destination     State    Type
-----
4026531842   10.3.3.2        10.1.1.2        up       static
4026531841   10.3.3.2        10.2.2.2        up       static
-----
Number of vxlan tunnel : 2
  
```

---End

Configuration Files

- Router1 configuration file

```

#
sysname Router1
#

acl number
3000
 rule 5 permit ip source 10.1.1.2 0 destination 10.3.3.2 0
#

ipsec proposal
s1
 esp authentication-algorithm
 sha2-256
 esp encryption-algorithm aes-256
#

ike proposal
1
 encryption-algorithm
 aes-256
 dh
 group2
 authentication-algorithm
 sha2-256
 authentication-method pre-
 share
  
```

```
integrity-algorithm hmac-  
sha2-256  
prf hmac-sha2-256  
#  
  
ike peer  
23  
pre-shared-key cipher %^%#I:TE+I3nvA"|a6GX){:*][TI2!r-EJ&,Ck*+)N(N%^  
%#  
ike-proposal  
1  
remote-address 192.168.2.2  
#  
  
ipsec policy map1 2  
isakmp  
security acl  
3000  
ike-peer  
23  
proposal s1  
#  
bridge-domain  
10  
vxlan vni 2010  
#  
  
interface  
Ethernet2/0/0  
undo  
portswitch  
ip address 192.168.2.1  
255.255.255.0  
ipsec policy map1  
#  
  
interface Ethernet2/0/1.1 mode  
12  
encapsulation dot1q vid  
10  
bridge-domain 10  
#  
  
interface  
LoopBack1  
ip address 10.1.1.2 255.255.255.255  
#  
  
interface  
Nve1  
source  
10.1.1.2  
vni 2010 head-end peer-list  
10.3.3.2  
#  
ospf  
1  
area  
0.0.0.0  
network 10.1.1.2  
0.0.0.0  
network 192.168.2.0 0.0.0.255  
#  
return
```

● Router2 configuration file

```
#  
sysname Router2  
#
```

```
acl number
3000
 rule 5 permit ip source 10.2.2.2 0 destination 10.3.3.2 0
#

ipsec proposal
s1
 esp authentication-algorithm
sha2-256
 esp encryption-algorithm aes-256
#

ike proposal
1
 encryption-algorithm
aes-256
 dh
group2
 authentication-algorithm
sha2-256
 authentication-method pre-
share
 integrity-algorithm hmac-
sha2-256
 prf hmac-sha2-256
#

ike peer
24
 pre-shared-key cipher %^%#%40zAxZ^A~Q}]@EPm$41CLh8A{Adv*G16\}G=GiM%^
%#
 ike-proposal
1
 remote-address 192.168.3.2
#

ipsec policy user1 2
isakmp
 security acl
3001
 ike-peer
24
 proposal s1
#

bridge-domain
20
 vxlan vni 2020
#

interface
Ethernet2/0/0
 undo
portswitch
 ip address 192.168.3.1
255.255.255.0
 ipsec policy user1
#

interface Ethernet2/0/1.1 mode
12
 encapsulation dot1q vid
20
 bridge-domain 20
#

interface
LoopBack1
 ip address 10.2.2.2 255.255.255.255
```

```
#
interface
Nve1
 source
10.2.2.2
 vni 2020 head-end peer-list
10.3.3.2
#
ospf
1
 area
0.0.0.0
 network 10.2.2.2
0.0.0.0
 network 192.168.3.0 0.0.0.255
#
return
```

● Router3 configuration file

```
#
sysname Router3
#
acl number 3000
 rule 5 permit ip source 10.3.3.2 0 destination 10.1.1.2 0
acl number 3001
 rule 5 permit ip source 10.3.3.2 0 destination 10.2.2.2 0
#
ipsec proposal
s1
 esp authentication-algorithm
sha2-256
 esp encryption-algorithm aes-256
#
ike proposal
1
 encryption-algorithm
aes-256
 dh
group2
 authentication-algorithm
sha2-256
 authentication-method pre-
share
 integrity-algorithm hmac-
sha2-256
 prf hmac-sha2-256
#
ike peer
21
 pre-shared-key cipher %^%#T*hBB{Pci9Xmp=+|}{(.@/2ki4h1G6N$`@`Ldj`+S%^
%#
 ike-proposal
1
 remote-address
192.168.2.1
ike peer
22
 pre-shared-key cipher %^%#$giTMpBP{PPF^c%!K.^>`!z4Tw>qFX>kX`(\|xhI%^
%#
 ike-proposal
1
 remote-address
192.168.3.1
#
```

```
ipsec policy map1 2
isakmp
 security acl
3000
 ike-peer
21
 proposal s1
ipsec policy user1 2
isakmp
 security acl
3001
 ike-peer
22
 proposal s1
#

bridge-domain
10
 vxlan vni 2010
bridge-domain
20
 vxlan vni 2020
#

interface
Ethernet2/0/1
 undo
portswitch
 ip address 192.168.2.2
255.255.255.0
 ipsec policy
map1
#

interface
Ethernet2/0/2
 undo
portswitch
 ip address 192.168.3.2
255.255.255.0
 ipsec policy user1
#

interface
LoopBack1
 ip address 10.3.3.2 255.255.255.255
#

interface
Vbdif10
 ip address 192.168.10.10
255.255.255.0
#
interface
Vbdif20
 ip address 192.168.20.10 255.255.255.0
#

interface
Nve1
 source
10.3.3.2
 vni 2010 head-end peer-list
10.1.1.2
 vni 2020 head-end peer-list
10.2.2.2
#
ospf
1
```

```
area
0.0.0.0
 network 10.3.3.2
0.0.0.0
 network 192.168.2.0 0.0.0.255
 network 192.168.3.0 0.0.0.255
#
return
```

14.8 References for VXLANs

The following table lists the references for VXLANs.

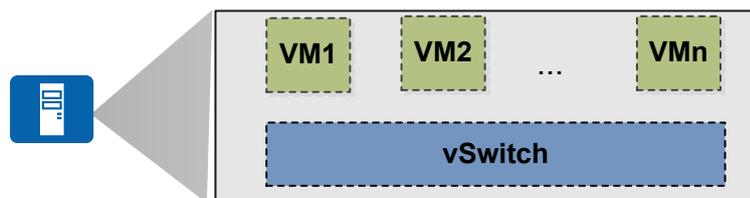
Document	Description
RFC 7348	Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks
RFC 7432	BGP MPLS-Based Ethernet VPN

14.9 Further Reading

14.9.1 Server Virtualization

Server virtualization virtualizes one physical server into multiple logical servers, that is virtual machines (VMs), as shown in [Figure 14-22](#).

Figure 14-22 Basic architecture of server virtualization



- VM
Each VM has its own operating system and application software, and has an independent MAC address and IP address. VMs can run independently.
- vSwitch
A vSwitch provides Layer 2 communication, isolation, and QoS capabilities for VMs.

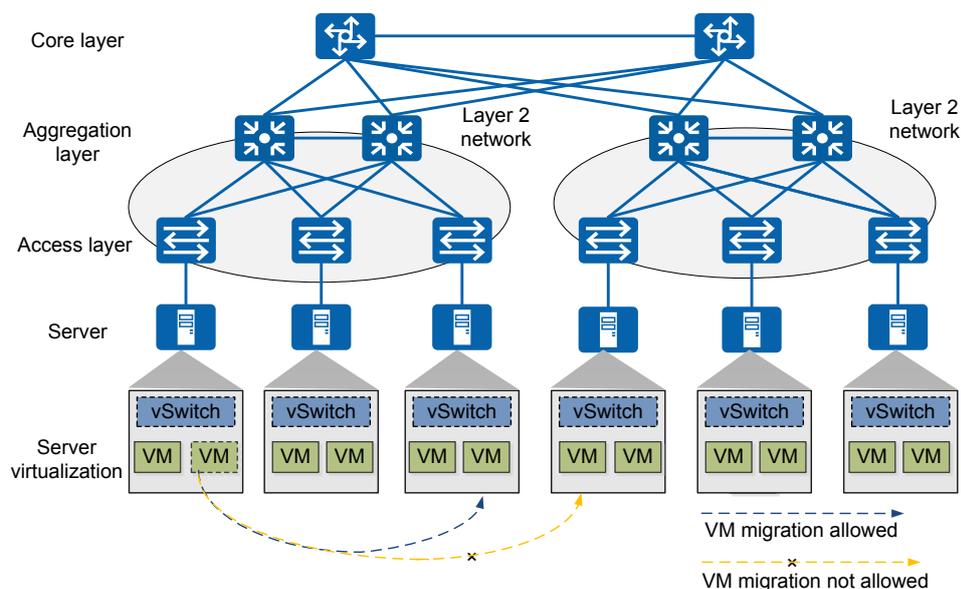
Server virtualization has the following advantages:

- Effectively improves server utilization.
- Provides services and resources on demand.
- Reduces energy consumption.
- Lowers customers' operations and maintenance (O&M) costs.

14.9.2 Large Layer 2 Network

Dynamic VM migration becomes a critical issue to meet flexible service changes. Dynamic VM migration is a process of moving VMs from one physical server to another, while ensuring normal running of the VMs. End users are unaware of this process, so administrators can flexibly allocate server resources or maintain and upgrade servers without affecting server usage by end users. The key of dynamic VM migration is to ensure uninterrupted services during the migration, so the IP and MAC addresses of VMs must remain unchanged. To meet this requirement, VM migration must occur within a Layer 2 domain but not across Layer 2 domains, as shown in [Figure 14-23](#).

Figure 14-23 VM migration on a traditional network



In the traditional data center network architecture, the Layer 2 network uses redundant devices and links to improve reliability. This will inevitably result in physical loops during VM migration.

To prevent broadcast storms caused by physical loops, a loop prevention protocol such as Spanning Tree Protocol (STP) is required to block redundant links. Due to STP limitations, an STP-enabled Layer 2 network can contain no more than 50 network nodes, so dynamic VM migration can only occur in a limited scope.

To enable VM migration in a large scope or across domains, servers involved must be on the same Layer 2 network, which is called large Layer 2 network.

Generally, the following technologies can be used to provide a large Layer 2 network:

- Network device virtualization
- Transparent Interconnection of Lots of Links (TRILL)
- VXLAN
- Ethernet Virtual Network (EVN)

Network device virtualization, TRILL, and EVN technologies can construct a physical large Layer 2 network to enlarge the VM migration scope. However, a physical large Layer 2 network requires huge changes to the existing network structure, and still has many restrictions on the VM migration scope. VXLAN can solve the preceding problems.

A virtual large Layer 2 network can solve the problem and enable VM migration in a larger scope, as shown in [Figure 14-24](#).

Figure 14-24 VM migration on a large Layer 2 network

