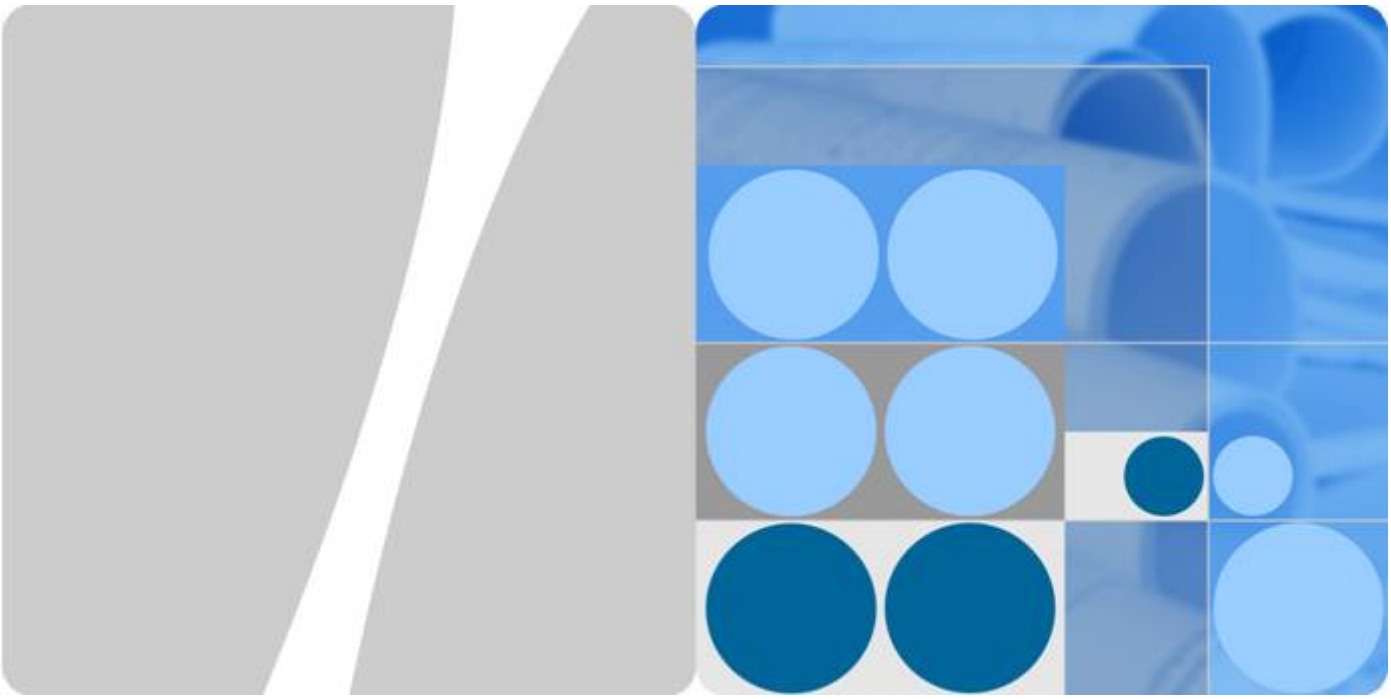


S7700&S9700 Smart&Core Routing Switch

V200R003C00

Configuration Guide - IP Service

Issue 02

Date 2013-07-25

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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://enterprise.huawei.com

About This Document

Intended Audience

This document describes the concepts and configuration procedures of IP Service features on the device, and provides the 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
DANGER	Indicates a hazard with a high level or medium level of risk which, if not avoided, could result in death or serious injury.
WARNING	Indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
CAUTION	Indicates a potentially hazardous situation that, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results.
TIP	Provides a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points in the main text.

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**.
Italic	Command arguments are in *italics*.
[ ]	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. You can select one or several items, or select no item.
&<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.

Password Setting Conventions

- If a password is set in plain text mode, the password is saved as the plain text in the configuration file, which brings security risks. Therefore, the cipher text mode is recommended for password setting. You are advised to change passwords regularly to ensure device security.
- If a password is set to a valid cipher text (can be decrypted on the device) string that starts and ends both with **%$%$**, the same cipher text is displayed when you check the configuration file on the device. Therefore, this password setting method is not recommended.

Change History

Updates between document issues are cumulative. Therefore, the latest document issue contains all updates made in previous issues.

Changes in Issue 02 (2013-07-25)

This version has the following updates:

The documentation is updated according to product feature updates.

Changes in Issue 01 (2013-05-30)

Initial commercial release.

Contents

1 IP Address Configuration

About This Chapter

Network devices can communicate at the network layer only after they are configured with IP addresses.

1.1 IP Address Overview
The Internet Protocol (IP) is the core protocol in the TCP/IP protocol suite. Data of TCP, UDP, ICMP and IGMP protocols is transmitted in IP packets. Devices on different network segments communicate with each other using IP addresses.

1.2 IP Addressing Features Supported by the Switch
You can configure a primary IP address and several secondary IP addresses for an interface. An interface can also borrow an IP address from another interface.

1.3 Configuring IP Addresses for Interfaces
To enable network devices to communicate at the network layer, configure interface IP addresses on the network devices.

1.4 Configuring an IP Unnumbered Interface
An IP unnumbered interface can borrow the IP address from another interface.

1.5 Configuration Examples
This section provides examples to explain how to configure the primary IP address, secondary IP addresses, and IP unnumbered on an interface.

1.6 Common Configuration Errors
This section describes common errors that may occur in IP address configuration. Learning this section helps you avoid faults caused incorrect IP address configuration.

1.1 IP Address Overview

The Internet Protocol (IP) is the core protocol in the TCP/IP protocol suite. Data of TCP, UDP, ICMP and IGMP protocols is transmitted in IP packets. Devices on different network segments communicate with each other using IP addresses.

An IP address is a 32-bit address used on the Internet. Each device on an IP network must have an IP address.

An IP address consists of a network ID and a host ID. The network ID identifies a network and the host ID identifies a specific network device on the network. Network devices with the same network ID are located on the same network, regardless of their physical locations.

1.2 IP Addressing Features Supported by the Switch

You can configure a primary IP address and several secondary IP addresses for an interface. An interface can also borrow an IP address from another interface.

You can configure the following IP address features on the switch:

- Manually configure the IP address for an interface
- IP unnumbered interface

Interfaces of the S7700&S9700 can be assigned overlapping network segments to save the address space.

- Interfaces on the same switch can be assigned IP addresses on overlapping network segments, but the IP addresses cannot be located on the same network segment. For example, an interface has been assigned 20.1.1.1/16. If you assign 20.1.1.2/24 to another interface on the same switch, the system displays a warning message but the configuration succeeds. If you assign 20.1.1.2/16 to another interface, the system displays an error message, indicating that the configuration fails because of an IP address conflict.
- The primary and secondary IP addresses of an interface can be located on overlapping network segments but not the same network segment. For example, if an interface has been assigned a primary IP address 20.1.1.1/24 and you assign secondary IP address 20.1.1.2/16 sub to this interface, the system displays a warning message but the configuration succeeds.
- The primary IP address of one interface and secondary IP address of another interface on the same switch can be located on overlapping network segments but not on the same network segment. For example, if an interface has been assigned a primary IP address 20.1.1.1/16 and you assign secondary IP address 20.1.1.2/24 sub to another interface on the switch, the system displays a warning message but the configuration succeeds.

1.3 Configuring IP Addresses for Interfaces

To enable network devices to communicate at the network layer, configure interface IP addresses on the network devices.

Pre-configuration Tasks

Before configuring IP addresses for interfaces, complete the following tasks:

- Setting link layer parameters for the interfaces to ensure that the link layer protocol status of the interfaces is Up

1.3.1 Configuring a Primary IP Address for an Interface

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 address ip-address { mask | mask-length }
```

A primary IP address is configured for the interface.

Each interface has only one primary IP address. If you configure multiple primary IP addresses for an interface, the last configured IP address becomes the primary IP address of the interface.

----End

1.3.2 (Optional) Configuring a Secondary IP Address for an Interface

Context

Generally, an interface needs only a primary IP address. In some special scenarios, you need to configure secondary IP addresses for an interface. For example, a switch connects to a physical network through an interface, and hosts on this network belong to two network segments. To enable the switch to communicate with all hosts on the physical network, configure a primary IP address and a secondary IP address for this interface. You can configure multiple IP address for a Layer 3 interface on a switch, one as the primary IP address, and the others as secondary IP addresses. Each Layer 3 interface can have a maximum of 255 secondary IP addresses.

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 address ip-address { mask | mask-length } sub
```

A secondary IP address is configured for the interface.

----End

1.3.3 Checking the Configuration

Procedure

- Run the **display ip interface** [**brief**] [*interface-type* [*interface-number*]] or **display ip interface** [*interface-type* *interface-number*] command to check the IP address configuration of an interface.
- Run the **display interface** [*interface-type* [*interface-number*]] command to check information about an interface.

----End

1.4 Configuring an IP Unnumbered Interface

An IP unnumbered interface can borrow the IP address from another interface.

Pre-configuration Tasks

Before configuring an IP unnumbered interface, complete the following tasks:

- Setting link layer parameters for the interfaces to ensure that the link layer protocol status of the interfaces is Up

1.4.1 (Optional) Configuring a Primary IP Address for the IP Numbered Interface

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

IP unnumbered interfaces can borrow IP addresses from interfaces including Ethernet, Loopback, VLANIF interfaces.

Step 3 Run:

```
ip address ip-address { mask | mask-length }
```

A primary IP address is configured for the interface.

Each interface has only one primary IP address. If you configure multiple primary IP addresses for an interface, the last configured IP address becomes the primary IP address of the interface.

----End

1.4.2 Configuring IP Unnumbered on an Interface

Context

In some scenarios, an interface can borrow an IP address from another interface to conserve IP addresses. For example, if an interface is seldom used, configure IP unnumbered on this interface so that the interface can share an IP address with another interface.

An IP unnumbered interface cannot run dynamic routing protocols because it does not have an IP address itself. To enable the interface to communicate with a peer network segment, configure a static route to the network segment.

The restrictions for IP unnumbered interface configuration are as follows:

- Ethernet interfaces do not support this configuration.
- One IP unnumbered interface cannot borrow the IP address from another IP unnumbered interface.
- The IP address of an interface can be shared by multiple IP unnumbered interfaces.
- If an interface has more than one IP address, IP unnumbered interfaces can borrow only the primary IP address.
- If the interface assigned to an IP unnumbered interface has no IP address, the IP unnumbered interface obtains IP address 0.0.0.0.
- Other interfaces can borrow IP addresses from loopback interfaces, but loopback interfaces cannot borrow IP addresses from other interfaces.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

The interface can be Tunnel, POS or Mtunnel interface.

Step 3 Run:

```
ip address unnumbered interface interface-type interface-number
```

The tunnel interface is configured to borrow the IP address from a specified interface.

----End

1.4.3 Checking the Configuration

Procedure

- Run the **display ip interface brief** [*interface-type* [*interface-number*]] or **display ip interface** [*interface-type interface-number*] command to check the IP address configuration of an interface.

- Run the **display interface** [*interface-type* [*interface-number*]] command to check information about an interface.

----End

1.5 Configuration Examples

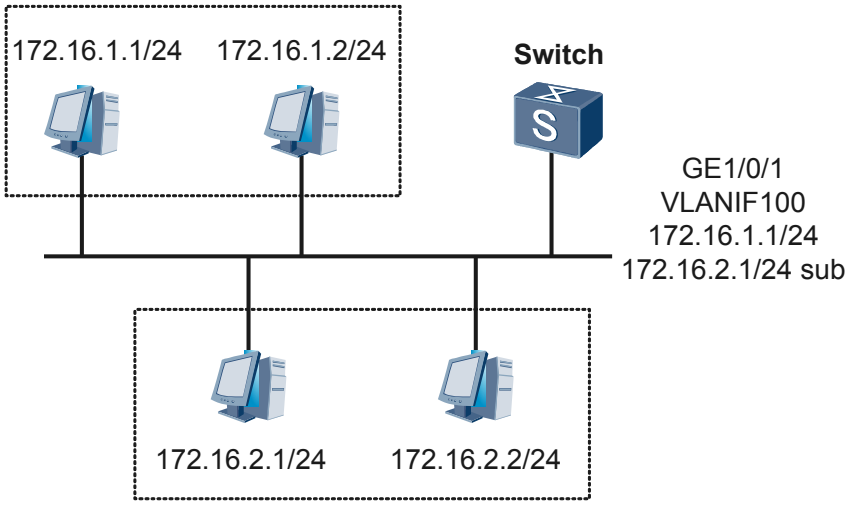
This section provides examples to explain how to configure the primary IP address, secondary IP addresses, and IP unnumbered on an interface.

1.5.1 Example for Configuring IP Addresses for an Interface

Networking Requirements

As shown in **Figure 1-1**, the Switch has only one idle interface GE1/0/1 to connect to a LAN. The hosts on the LAN are located on two network segments: 172.16.1.0/24 and 172.16.2.0/24. The interface must be configured with two interfaces to provide access for hosts on the two network segments.

Figure 1-1 Network diagram for IP addresses configuration



Configuration Roadmap

The configuration roadmap is as follows:

Configure a primary IP address and a secondary IP address for the interface.

NOTE

IP addresses of the same interface must be on different network segments.

Procedure

Step 1 Add GE1/0/1 to VLAN 100, and configure a primary IP address and a secondary IP address for VLANIF100.

```
<Quidway> system-view
[Quidway] vlan 100
[Quidway-Vlan100] quit
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[Quidway-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface vlanif 100
[Quidway-Vlanif100] ip address 172.16.1.1 24
[Quidway-Vlanif100] ip address 172.16.2.1 24 sub
```

Step 2 Verify the configuration.

Ping a host on network segment 172.16.1.0 from the Switch. The ping operation succeeds.

```
<Quidway> ping 172.16.1.2
PING 172.16.1.2: 56  data bytes, press CTRL_C to break
    Reply from 172.16.1.2: bytes=56 Sequence=1 ttl=128 time=25 ms
    Reply from 172.16.1.2: bytes=56 Sequence=2 ttl=128 time=27 ms
    Reply from 172.16.1.2: bytes=56 Sequence=3 ttl=128 time=26 ms
    Reply from 172.16.1.2: bytes=56 Sequence=4 ttl=128 time=26 ms
    Reply from 172.16.1.2: bytes=56 Sequence=5 ttl=128 time=26 ms
  --- 172.16.1.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 25/26/27 ms
```

Ping a host on network segment 172.16.2.0 from the Switch. The ping operation succeeds.

```
<Quidway> ping 172.16.2.2
  PING 172.16.2.2: 56  data bytes, press CTRL_C to break
    Reply from 172.16.2.2: bytes=56 Sequence=1 ttl=128 time=25 ms
    Reply from 172.16.2.2: bytes=56 Sequence=2 ttl=128 time=26 ms
    Reply from 172.16.2.2: bytes=56 Sequence=3 ttl=128 time=26 ms
    Reply from 172.16.2.2: bytes=56 Sequence=4 ttl=128 time=26 ms
    Reply from 172.16.2.2: bytes=56 Sequence=5 ttl=128 time=26 ms
  --- 172.16.2.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 25/25/26 ms
```

----End

Configuration Files

Configuration file of the Switch

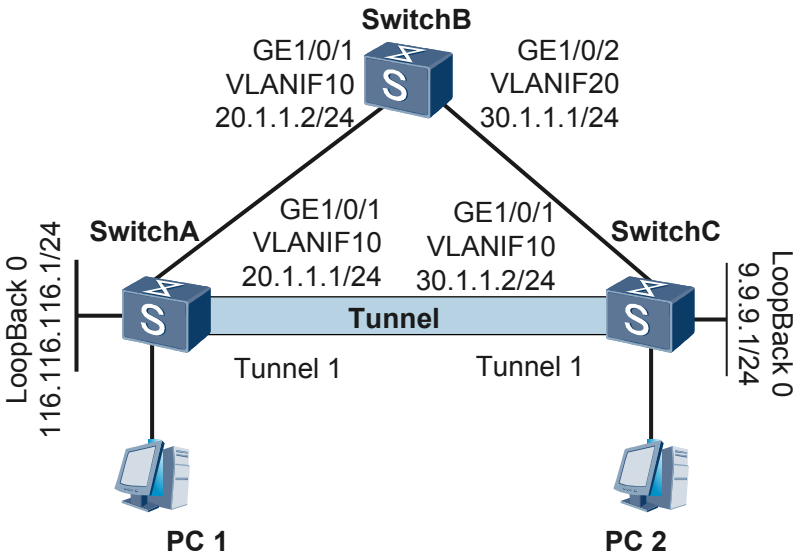
```
#
 sysname Quidway
#
 vlan batch 100
#
interface Vlanif100
 ip address 172.16.1.1 255.255.255.0
 ip address 172.16.2.1 255.255.255.0 sub
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
return
```

1.5.2 Example for Configuring an IP Unnumbered Interface

Networking Requirements

As shown in **Figure 1-2**, Tunnel interfaces (Tunnel1) of SwitchA and SwitchC are seldom used, so they have no IP address configured. IP unnumbered need to be configured on the tunnel interfaces so that the two switches can communicate through the tunnel.

Figure 1-2 Network diagram for IP unnumbered interface configuration



Configuration Roadmap

The configuration roadmap is as follows:

1. Create tunnel interfaces on SwitchA and SwitchC, set up a GRE tunnel between them, and specify the source and destination addresses of the tunnel interfaces.
2. On SwitchA and SwitchC, configure an IP address for a loopback interface and configure the tunnel interface to borrow the IP address from this loopback interface.

Procedure

Step 1 Configure public IP and the IP address of interface Loopback0

Configure SwitchA.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port link-type access
[SwitchA-GigabitEthernet1/0/1] port default vlan 10
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] ip address 20.1.1.1 24
```

```
[SwitchA-Vlanif10] quit
[SwitchA] interface loopback 0
[SwitchA-LoopBack0] ip address 116.116.116.1 24
[SwitchA-LoopBack0] quit
```

Configure SwitchB.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] vlan batch 10 20
[SwitchB] interface gigabitethernet 1/0/1
[SwitchB-GigabitEthernet1/0/1] port link-type access
[SwitchB-GigabitEthernet1/0/1] port default vlan 10
[SwitchB-GigabitEthernet1/0/1] quit
[SwitchB] interface gigabitethernet 1/0/2
[SwitchB-GigabitEthernet1/0/2] port link-type access
[SwitchB-GigabitEthernet1/0/2] port default vlan 20
[SwitchB-GigabitEthernet1/0/2] quit
[SwitchB] interface vlanif 10
[SwitchB-Vlanif10] ip address 20.1.1.2 24
[SwitchB-Vlanif10] quit
[SwitchB] interface vlanif 20
[SwitchB-Vlanif20] ip address 30.1.1.1 24
[SwitchB-Vlanif20] quit
```

Configure SwitchC.

```
<Quidway> system-view
[Quidway] sysname SwitchC
[SwitchC] vlan 10
[SwitchC-vlan10] quit
[SwitchC] interface gigabitethernet 1/0/1
[SwitchC-GigabitEthernet1/0/1] port link-type access
[SwitchC-GigabitEthernet1/0/1] port default vlan 10
[SwitchC-GigabitEthernet1/0/1] quit
[SwitchC] interface vlanif 10
[SwitchC-Vlanif10] ip address 30.1.1.2 24
[SwitchC-Vlanif10] quit
[SwitchC] interface loopback 0
[SwitchC-LoopBack0] ip address 9.9.9.1 24
[SwitchC-LoopBack0] quit
```

Step 2 Configure OSPF on the devices

Configure SwitchA.

```
[SwitchA] ospf 1
[SwitchA-ospf-1] area 0
[SwitchA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[SwitchA-ospf-1-area-0.0.0.0] quit
[SwitchA-ospf-1] quit
```

Configure SwitchB.

```
[SwitchB] ospf 1
[SwitchB-ospf-1] area 0
[SwitchB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[SwitchB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[SwitchB-ospf-1-area-0.0.0.0] quit
[SwitchB-ospf-1] quit
```

Configure SwitchC.

```
[SwitchC] ospf 1
[SwitchC-ospf-1] area 0
[SwitchC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
```

```
[SwitchC-ospf-1-area-0.0.0.0] quit
[SwitchC-ospf-1] quit
```

Step 3 Configure Tunnel1 to borrow the IP address from Loopback0 and configure the gre tunnel.

Configure SwitchA.

```
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] tunnel-protocol gre
[SwitchA-Tunnel1] ip address unnumbered interface loopback 0
[SwitchA-Tunnel1] source 20.1.1.1
[SwitchA-Tunnel1] destination 30.1.1.2
[SwitchA-Tunnel1] quit
```

Configure SwitchC.

```
[SwitchC] interface tunnel 1
[SwitchC-Tunnel1] tunnel-protocol gre
[SwitchC-Tunnel1] ip address unnumbered interface loopback 0
[SwitchC-Tunnel1] source 30.1.1.2
[SwitchC-Tunnel1] destination 20.1.1.1
[SwitchC-Tunnel1] quit
```

Step 4 Configure static routes.

Configure SwitchA.

```
[SwitchA] ip route-static 9.9.9.0 255.255.255.0 tunnel 1
```

Configure SwitchC.

```
[SwitchC] ip route-static 116.116.116.0 255.255.255.0 tunnel 1
```

Step 5 Verify the configuration.

Ping 9.9.9.1 from SwitchA. The ping operation succeeds.

```
[SwitchA] ping 9.9.9.1
  PING 9.9.9.1: 56  data bytes, press CTRL_C to break
    Reply from 9.9.9.1: bytes=56 Sequence=1 ttl=255 time=2 ms
    Reply from 9.9.9.1: bytes=56 Sequence=2 ttl=255 time=3 ms
    Reply from 9.9.9.1: bytes=56 Sequence=3 ttl=255 time=3 ms
    Reply from 9.9.9.1: bytes=56 Sequence=4 ttl=255 time=3 ms
    Reply from 9.9.9.1: bytes=56 Sequence=5 ttl=255 time=3 ms

  --- 9.9.9.1 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

- Configuration file of SwitchA

```
#
 sysname SwitchA
#
vlan batch 10
#
interface LoopBack0
 ip address 116.116.116.1 255.255.225.0
#
interface Vlanif10
 ip address 20.1.1.1 255.255.255.0
```

```
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 10
#
interface Tunnel1
 ip address unnumbered interface LoopBack0
 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 9.9.9.0 255.255.255.0 Tunnel1
#
return
```

- Configuration file of SwitchB

```
#
sysname SwitchB
#
vlan batch 10 20
#
interface Vlanif10
 ip address 20.1.1.2 255.255.255.0
#
interface Vlanif20
 ip address 30.1.1.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 10
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 20
#
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 SwitchC

```
#
 sysname SwitchC
#
vlan batch 10
#
interface LoopBack0
 ip address 9.9.9.1 255.255.225.0
#
interface Vlanif10
 ip address 30.1.1.2 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 10
#
interface Tunnel1
 ip address unnumbered interface LoopBack0
 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 116.116.116.0 255.255.255.0 Tunnel1
#
return
```

1.6 Common Configuration Errors

This section describes common errors that may occur in IP address configuration. Learning this section helps you avoid faults caused incorrect IP address configuration.

1.6.1 IP Address Configuration Fails on an Interface

Fault Analysis

An error occurs in IP address configuration, so the configuration fails.

Procedure

Step 1 Check the error message and rectify the fault according to **Table 1-1**.

Table 1-1 Error messages and ways to rectify faults

Error Message	Description	Troubleshooting Method
Error: The specified IP address is invalid.	The IP address or subnet mask is incorrect.	Configure the IP address or subnet mask correctly. ● The IP address must be a Class A, Class B, or Class C IP address. ● The subnet mask must match the IP address.
Error: The specified address conflicts with another address.	The specified IP address is on the same network segment as the IP address of another interface on the local device.	Configure another IP address for the interface.

Error Message	Description	Troubleshooting Method
Error: The specified primary address does not exist.	The primary IP address to be deleted does not exist. **NOTE** Each interface has only one primary IP address. If you configure multiple primary IP addresses for an interface, the last configured IP address becomes the primary IP address of the interface.	You do not need to delete the IP address.
Error: Please configure the primary address in the interface view first.	● The secondary IP address cannot be configured because the primary IP address has not been configured for the interface. ● If IP unnumbered is configured on the interface, no secondary IP address can be configured on the interface.	Configure a primary IP address for the interface first.
Error: The number of addresses of the specified interface reached the upper limit (256).	The number of secondary IP addresses on the interface exceeds the maximum; therefore, no more secondary IP address can be configured. **NOTE** Each interface can have a maximum of 255 IP addresses.	-
Error: Please delete the sub address in the interface view first.	The primary IP address cannot be deleted because the interface has secondary IP addresses.	Delete all the secondary IP addresses from the interface, and then delete the primary IP address.
Error: The specified address cannot be deleted because it is not the primary address of this interface.	The command used to delete a primary IP address cannot delete a secondary IP address.	Run the **undo ip address** *ip-address* { *mask* | *mask-length* } **sub** command to delete the secondary IP address.
Error: The specified sub address does not exist.	The secondary IP address to be deleted does not exist.	You do not need to delete the IP address.

Error Message	Description	Troubleshooting Method
Error: The address already exists.	The interface has been configured with the same IP address.	Configure a different IP address for the interface.

----End

1.6.2 An Interface Fails to Communicate with the Peer Device After IP Unnumbered Is Configured

Fault Analysis

The possible causes are:

- The IP unnumbered interface is not a tunnel interface.
- The specified numbered interface has no IP address configured.
- The IP unnumbered interface has no static route to the peer device.

Procedure

Step 1 Check whether the IP unnumbered interface is a tunnel interface.

NOTE

Currently, only tunnel, POS or MTunnel interfaces on the device can borrow IP addresses from other interfaces. Ethernet interfaces cannot borrow IP addresses from other interfaces.

If IP unnumbered is configured in the tunnel interface view, go to step 2.

Step 2 Check whether the numbered interface specified in the IP unnumbered interface configuration has an IP address.

Run the **display this** command in this interface view to check whether the interface has an IP address.

- If **ip address x.x.x.x** is not displayed, run the **ip address** *ip-address* { *mask* | *mask-length* } command in this interface view to configure an IP address.
- If **ip address x.x.x.x** is displayed, go to step 3.

Step 3 Check whether a static route to the peer device is configured on the local device.

NOTE

The IP unnumbered interface cannot run dynamic routing protocols because it does not have an IP address itself. To enable the interface to communicate with the peer device, configure a static route to the peer device.

Run the **display ip routing-table** command to check whether a static route to the peer device is configured. In the routing table, static routes are identified by **Static** in the **Proto** field.

- If no static route in the routing table matches the destination IP address on the peer device, run the **ip route-static** *ip-address* { *mask* | *mask-length* } *interface-type interface-number* [*nexthop-address*] command to configure a static route.

- If the routing table contains a static route matching the destination IP address on the peer device and the outbound interface is the unnumbered interface, go to step 4.

Step 4 Collect the following information and contact Huawei technical support personnel.

- Results of the preceding troubleshooting procedure
- Configuration file, logs, and alarms of the local device

----End

2 UDP Helper Configuration

About This Chapter

This chapter describes the principle and configuration of UDP helper, and provides configuration examples.

2.1 UDP Helper Overview
The UDP helper function relays the UDP broadcast packets destined for specified ports.

2.2 Configuring UDP Helper
The UDP helper function relays the UDP broadcast packets destined for specified ports.

2.3 Maintaining UDP Helper
UDP helper maintenance includes displaying and clearing UDP helper statistics.

2.4 Configuration Examples
This example describes how to configure UDP helper on a switch.

2.1 UDP Helper Overview

The UDP helper function relays the UDP broadcast packets destined for specified ports.

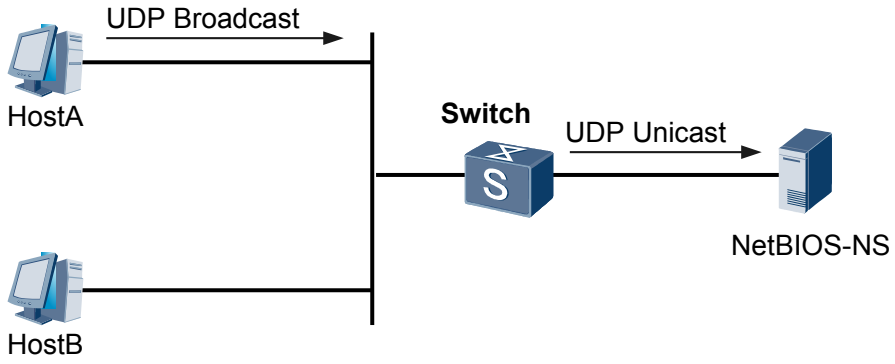
Background

Hosts on a network may need to obtain the network configuration or resolve host names by sending UDP broadcast packets to the server. If the hosts and server are located in different broadcast domains, broadcast packets cannot reach the server and the hosts cannot obtain the required information from the server.

The switch provides the UDP helper function to solve this problem. UDP helper can relay the UDP broadcast packets with specified destination ports. It converts the broadcast packets into unicast packets and sends the unicast packets to the specified destination servers.

As shown in **Figure 2-1**, HostA uses a host name to access HostB, and the NetBIOS Name Service (NetBIOS-NS) server resolves the host name of HostB. The NetBIOS-NS server and HostA are in different broadcast domains, so the UDP broadcast packet with destination port UDP 137 sent by HostA cannot reach the NetBIOS-NS server. After UDP helper is enabled on the Switch, the Switch can forward the packet with destination port UDP 137 to the NetBIOS-NS server through unicast so that the NetBIOS-NS server can resolve the host name of HostB.

Figure 2-1 UDP helper relays broadcast packets



Packets Forwarded by UDP Helper

The packets that can be forwarded by UDP helper must meet the following requirements:

- The destination MAC address is the broadcast MAC address (ffff-ffff-ffff).
- The Time-to-Live (TTL) is larger than 1.
- The protocol type is UDP.
- The destination port is a specified UDP port.

UDP Helper Ports

After UDP helper is enabled on the switch, the switch relays the UDP packets with six specified destination ports by default. Manual configuration is required if the switch needs to relay the

UDP packets with other destination ports. A maximum of 40 UDP port numbers can be configured on the switch.

Table 2-1 lists the default UDP ports.

Table 2-1 Default UDP ports supported by UDP helper

Protocol	UDP Port Number
Trivial File Transfer Protocol (TFTP)	69
Domain Name System (DNS)	53
Time Service	37
NetBIOS Name Service (NetBIOS-NS)	137
NetBIOS Datagram Service (NetBIOS-DS)	138
Terminal Access Controller Access Control System (TACACS)	49

NOTE

UDP helper does not relay DHCP packets. That is, the destination port number cannot be 67 or 68. To relay DHCP packets, enable the DHCP relay function on the switch. For details about DHCP relay, see **Configuring DHCP Relay Agent**.

2.2 Configuring UDP Helper

The UDP helper function relays the UDP broadcast packets destined for specified ports.

Pre-configuration Tasks

Before configuring UDP helper, complete the following task:

- Configuring a reachable route from the switch to the destination server

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
udp-helper enable
```

UDP helper is enabled.

Step 3 (Optional) Run:

```
udp-helper port { port-number | dns | netbios-ds | netbios-ns | tacacs | tftp | time }
```

The UDP destination port to which UDP broadcast packets are relayed is specified.

NOTE

- After UDP helper is enabled, the switch relays the UDP packets with the following destination ports by default: Time (37), TACACS (49), DNS (53), TFTP (69), NetBIOS-NS (137), and NetBIOS-DS (138). If the UDP destination port you want to specify is among the six ports, skip this step.

Step 4 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

The interface must be a VLANIF interface.

Step 5 Run:

```
udp-helper server ip-address
```

The destination server for UDP helper is specified.

----End

Checking the Configuration

- Run the **display udp-helper port** command to check the UDP port numbers of the packets that need to be relayed.

2.3 Maintaining UDP Helper

UDP helper maintenance includes displaying and clearing UDP helper statistics.

2.3.1 Displaying UDP Helper Statistics

Procedure

- Run the **display udp-helper server** command to display the packet relay interface, destination server address, and number of forwarded packets.

----End

2.3.2 Clearing UDP Helper Statistics

Context

CAUTION

UDP helper statistics cannot be restored after being cleared. Exercise caution when you run the **reset udp-helper packet** command.

Procedure

- Run the **reset udp-helper packet** command in the user view to clear UDP helper statistics.

----End

2.4 Configuration Examples

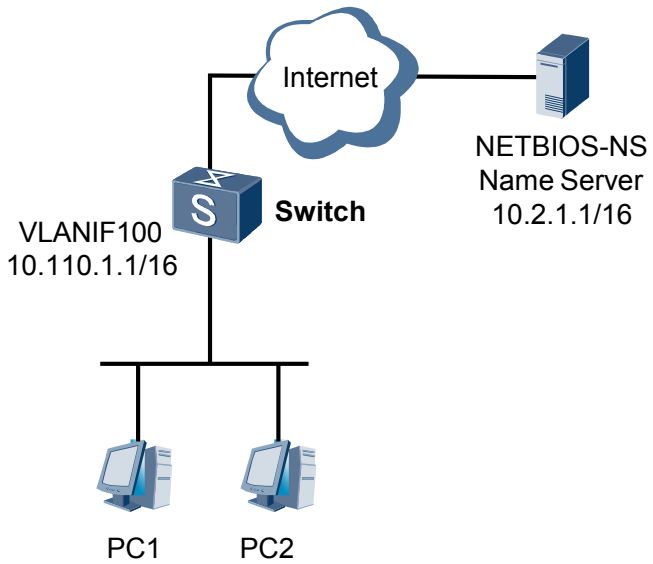
This example describes how to configure UDP helper on a switch.

2.4.1 Example for Configuring UDP Helper

Networking Requirements

As shown in **Figure 2-2**, the IP address of VLANIF100 on the Switch is 10.110.1.1/16, and the IP address of the NetBIOS-NS server is 10.2.1.1/16. The Switch and NetBIOS-NS server are on different network segments, and a reachable route exists between them. The PCs need to access each other using host names.

Figure 2-2 UDP helper network



Configuration Roadmap

The configuration roadmap is as follows:

Relay the UDP packets with destination port 137 and destination address 255.255.255.255 and the UDP packets with destination address 10.110.255.255 to the specified NetBIOS-NS server.

When the Switch receives a broadcast NetBIOS-NS Register packet, it changes the destination IP address in the IP header of the broadcast packet to the IP address of the NetBIOS-NS server and forwards the packet to the NetBIOS-NS server.

NOTE

After UDP helper is enabled on the Switch, the Switch relays the broadcast packets with UDP destination port 137 by default. The UDP port number, therefore, does not need to be configured in this example.

Procedure

Step 1 Enable UDP helper.

```
<Quidway> system-view
[Quidway] udp-helper enable
```

Step 2 Configure a destination server.

```
[Quidway] vlan 100
[Quidway-Vlan100] quit
[Quidway] interface vlanif 100
[Quidway-Vlanif100] ip address 10.110.1.1 16
[Quidway-Vlanif100] udp-helper server 10.2.1.1
[Quidway-Vlanif100] quit
[Quidway] quit
```

Step 3 Verify the configuration.

The destination server configured on VLANIF100 is the NetBIOS-NS server.

```
<Quidway> display udp-helper server interface vlanif 100
vlan-interface        Server-Ip        packet-num
Vlanif100               10.2.1.1               0
```

----End

Configuration Files

Configuration file of the Switch

```
#
 sysname Quidway
#
 vlan batch 100
#
 udp-helper enable
#
interface Vlanif100
 ip address 10.110.1.1 255.255.0.0
 udp-helper server 10.2.1.1
#
return
```

3 ARP Configuration

About This Chapter

The Address Resolution Protocol (ARP) maps IP addresses to MAC addresses so that Ethernet frames can be transmitted on a physical network.

3.1 ARP Overview
As the basis of Ethernet network communication, ARP maps IP addresses to MAC addresses.

3.2 ARP Features Supported by the Device
ARP can be a dynamic ARP or a static ARP. ARP provides some extended functions, such as proxy ARP, ARP-Ping, Egress ARP Inspection (EAI), ARP packet forwarding after port isolation and so on.

3.3 Default Configuration
This section describes default ARP configurations.

3.4 Configuring Static ARP
Static ARP entries improve communication security.

3.5 Optimizing Dynamic ARP
By default, hosts and switches dynamically learn ARP entries. You can adjust parameters of dynamic ARP entries based on network requirements.

3.6 Configuring Proxy ARP
The switch can function as a proxy of the destination host to reply an ARP Request message.

3.7 Configuring ARP-Ping
ARP-Ping includes ARP-Ping IP and ARP-Ping MAC. ARP-Ping sends ARP Request packets or ICMP Echo Request packets to check whether a specified IP address or MAC address is used.

3.8 Enabling a Device to Learn Multicast MAC Addresses and Generate ARP Entries
If a device is enabled to learn multicast MAC addresses, it can generate ARP entries after receiving ARP packets carrying multicast MAC addresses as source MAC addresses. This section describes how to enable a device to learn multicast MAC addresses and generate ARP entries.

3.9 Configuring ARP Automatic Scanning and Fixed ARP
ARP automatic scanning and fixed ARP enable a device to generate dynamic ARP entries and convert the dynamic ARP entries to static ARP entries.

3.10 Configuring the function to detect IP address conflicts

After the function to detect IP address conflicts is enabled, the device will detect the IP address conflicts by ARP packets.

3.11 Configuring Egress ARP Inspection

Egress ARP inspection enables the switch to restrict the scope of ARP packet forwarding. This function prevents broadcast of ARP packets in a VLAN and reduces the traffic volume in the VLAN.

3.12 Configuring a Static Unicast or Multicast MAC Address

After a static unicast or multicast MAC address is configured on an interface, unicast or multicast packets destined for the unicast or multicast MAC address are forwarded only to the interface.

3.13 Configuring ARP Packet Forwarding Between Isolated Interfaces

After ARP packet forwarding between isolated interfaces is configured, the device that has EAI enabled forwards packets to trusted interfaces when port isolation is enabled on both inbound and outbound interfaces. This allows isolated users to communicate.

3.14 Maintaining ARP

Maintaining ARP includes clearing ARP entries and monitoring ARP running status.

3.15 Configuration Examples

This section provides configuration examples including networking requirements and configuration roadmap.

3.1 ARP Overview

As the basis of Ethernet network communication, ARP maps IP addresses to MAC addresses.

On a local area network (LAN), a host or a network device must learn the IP address of the destination host or device before sending data to it. Additionally, the host or network device must learn the physical address of the destination host or device because IP packets must be encapsulated into frames for transmission over a physical network. Therefore, the mapping from an IP address into a physical address is required. ARP is used to map IP addresses into physical addresses.

3.2 ARP Features Supported by the Device

ARP can be a dynamic ARP or a static ARP. ARP provides some extended functions, such as proxy ARP, ARP-Ping, Egress ARP Inspection (EAI), ARP packet forwarding after port isolation and so on.

Comparison Between Dynamic ARP and Static ARP

ARP entries are classified into static and dynamic ARP entries.**Table 3-1** describes dynamic ARP and static ARP, including entry generation , maintenance, and application scenarios.

Table 3-1 Comparison between dynamic ARP and static ARP

Item	Concept	Entry Generation and Maintenance	Usage Scenario
Dynamic ARP	Dynamic ARP entries are generated and maintained automatically using the ARP protocol.	● They can be aged, updated, or overridden by static ARP entries. ● By default, ARP entries are dynamically learned and maintained.	Dynamic ARP entries are generated and maintained dynamically by the ARP protocol.

Item	Concept	Entry Generation and Maintenance	Usage Scenario
Static ARP	Static ARP entries are manually configured. Mappings between IP addresses and MAC addresses are fixed.	Static ARP entries are manually configured and maintained. They cannot be aged and overridden by dynamic ARP entries. **NOTE** Static ARP entries improve communication security. However, a large number of ARP entries increase configuration and maintenance costs.	Static ARP is configured to: ● Direct the packets whose destination IP addresses are not on the local network segment to a gateway on the local network segment so that the packets can be forwarded by the gateway. ● Bind destination IP addresses of illegal packets to a nonexistent MAC address so that illegal packets are filtered out. Static ARP entries can be configured on important network devices such as servers to specify member devices that they can communicate with. In this way, mappings between IP addresses and MAC addresses of these member devices cannot be modified by forged ARP packets and illegal ARP replies can be prevented. This protects servers against network attacks.

ARP Extended Functions

ARP provides extended functions in the following table.

Table 3-2 ARP extended functions and their usage scenarios

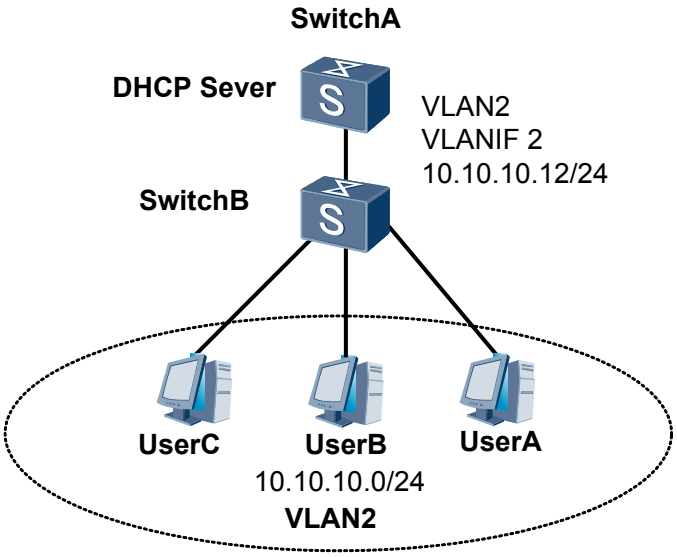
Function	Concept	Usage Scenario
Proxy ARP	The switch can function as a proxy of the destination host to reply an ARP Request message.	Proxy ARP is classified into the following three types: 1. Routed Proxy ARP: Routed Proxy ARP enables network devices on the same network segment but on different physical networks to communicate. 2. Intra-VLAN Proxy ARP: Intra-VLAN Proxy ARP enables isolated network devices in a VLAN to communicate. 3. Inter-VLAN Proxy ARP: Inter-VLAN Proxy ARP enables network devices in different VLANs or network devices in different sub-VLANs but on the same network segment to communicate.
ARP-Ping	ARP-Ping includes ARP-Ping IP and ARP-Ping MAC. ARP-Ping sends ARP Request packets or broadcasts ICMP Echo Request packets to check whether a specified IP address or MAC address is used.	● The ARP-Ping IP function checks whether an IP address is used by another device on the network. ● The ARP-Ping MAC function checks whether a MAC address is used or queries the IP address mapping the MAC address.

Egress ARP Inspection

As shown in **Figure 3-1**, SwitchB is located between a Layer 3 switch SwitchA (DHCP server) and user hosts. All the user hosts belong to VLAN 2 and obtain IP addresses through DHCP.

This network has the following problem: When SwitchB receives an ARP Request packet, it broadcasts the packet in the VLAN. This increases the traffic volume in the VLAN.

Figure 3-1 EAI networking



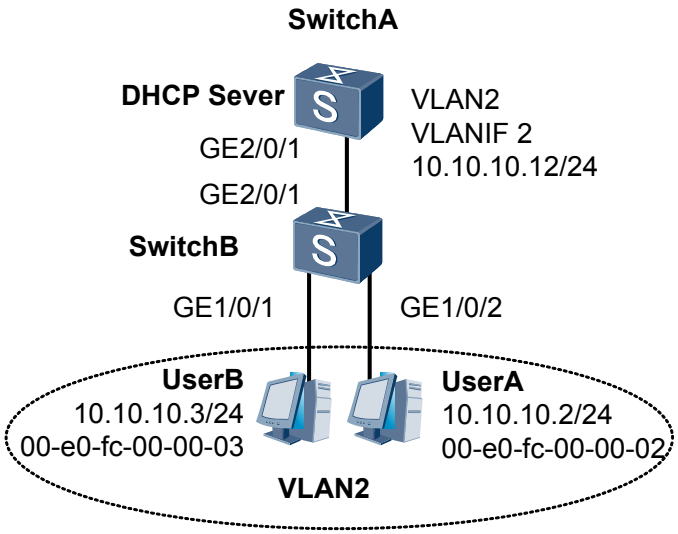
The EAI function can work with DHCP snooping to solve the preceding problem. After EAI is enabled in VLAN 2 on SwitchB, SwitchB matches the destination IP addresses of received ARP Request packets with the dynamic binding entries generated by DHCP snooping to determine the outbound interfaces. If the destination IP address of an ARP Request packet matches a dynamic binding entry, SwitchB sends the packet to the outbound interface specified in the binding entry. If the destination IP address matches a static binding entry and no outbound interface is specified in the static binding entry, SwitchB searches the dynamic MAC address table for the outbound interface according to the MAC address in the static binding entry. If the ARP Request packet matches no binding entry, SwitchB forwards the packet to the trusted interface or drops the packet.

Configuring ARP Packet Forwarding Between Isolated Interfaces

As shown in **Figure 3-2**, SwitchB is located between a Layer 3 switch SwitchA (DHCP server) and user hosts. All the user hosts belong to VLAN2 and obtain IP addresses through DHCP. Port isolation is configured on interfaces GE1/0/1 and GE1/0/2 that connect UserA and UserB to the network respectively. Intra-VLAN proxy ARP is enabled on VLANIF2 of SwitchA to enable UserA and UserB in a VLAN to be isolated at Layer 2 and to communicate at Layer 3.

After receiving ARP Request packets sent from UserA to UserB, SwitchB that has EAI enabled matches destination IP addresses of received ARP Request packets with dynamic binding entries generated by DHCP snooping to determine outbound interfaces for the packets. If the destination IP address of an ARP Request packet (IP address of UserB) matches a dynamic binding entry, SwitchB sends the packet to the destination interface, that is GE1/0/1 of UserB. However, the destination interface and the inbound interface of the ARP Request packet (GE1/0/2 of UserA) are configured with port isolation. ARP Request packets are discarded. UserA and UserB cannot communicate.

Figure 3-2 Configuring ARP Packet Forwarding Between Isolated Interfaces



To address this problem, the device provides the function of ARP packet forwarding between isolated interfaces.

After ARP packet forwarding between isolated interfaces is enabled in VLAN2 on SwitchB, SwitchB forwards ARP Request packets to a trusted interface (GE2/0/1 connecting SwitchB to SwitchA). Enabling intra-VLAN ARP proxy on SwitchA allows isolated users to communicate with each other.

Precautions for Configuring ARP-CPCAR

The switch has a default CIR value for each type of protocol packet. You can adjust CIR values for specified types of protocol packets based on services and network environment.

NOTE

The CIR values listed in the following tables are for reference only. Adjust CIR values based on services and network environment to prevent high CPU usage.

When switches need to learn a lot of ARP entries, the default CIR value for ARP packets cannot meet requirements for sending ARP packets so that ARP entries are learned slowly. When a large number ARP entries exist and these entries are aged out simultaneously, the default CIR value for ARP Reply packets cannot meet requirements for sending ARP Reply packets. This may lead to the loss of ARP Reply packets and deletion of some ARP entries because their aging time cannot be updated. To avoid these problems, run the **display arp statistics all** command to check statistics on ARP entries and adjust CIR values for ARP Request and ARP Reply packets based on the ARP entry quantity. For details, see **Table 3-3**.

Table 3-3 Recommended CIR values for ARP Request and ARP Reply packets

Number of ARP Entries	Recommended CIR (kbit/s)
<1k	128
1k to 3k	256

Number of ARP Entries	Recommended CIR (kbit/s)
3k to 4k	512
4k to 5k	768

3.3 Default Configuration

This section describes default ARP configurations.

Table 3-4 describes the default configuration of ARP.

Table 3-4 Default ARP configuration

Parameter	Default Configuration
Aging time of dynamic ARP entries	1200 seconds
Maximum number of probes for aging dynamic ARP entries	3 times
Aging detection mode of dynamic ARP entries	An interface sends ARP aging probe packets in broadcast mode.
Layer 2 topology detection	Layer 2 topology detection is disabled.
ARP proxy	ARP proxy is disabled.

3.4 Configuring Static ARP

Static ARP entries improve communication security.

Context

Static ARP entries are manually configured and maintained. They cannot be aged and overridden by dynamic ARP entries. Therefore, static ARP entries 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.

NOTE

Static ARP entries cannot be modified. However, the configuration workload is heavy. Static ARP entries cannot apply to a network where IP addresses of hosts may change or a small-sized network.

Pre-configuration Tasks

Before configuring static ARP entries, complete the following tasks:

- Setting link layer protocol parameters for interfaces to ensure that the link layer protocol status of the interfaces is Up

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
arp static ip-address mac-address [ vpn-instance vpn-instance-name ] or arp static ip-address mac-address vid vlan-id [ cevid ce-vid ] interface interface-type interface-number [ .subinterface-number ]
```

A static ARP entry is configured.

----End

Checking the Configuration

After configuring the static ARP entries is complete, run the following commands to check the configuration.

- Run the **display arp vpn-instance** *vpn-instance-name* [**dynamic** | **static**] command to check ARP mapping entries of a specified VPN instance.
- Run the **display arp static** command to check static ARP mapping entries.
- Run the **display arp track** command to check changes of outbound interfaces in ARP entries learned by a VLANIF interface.

3.5 Optimizing Dynamic ARP

By default, hosts and switches dynamically learn ARP entries. You can adjust parameters of dynamic ARP entries based on network requirements.

Pre-configuration Tasks

Before optimizing dynamic ARP, complete the following tasks:

- Setting link layer protocol parameters for interfaces to ensure that the link layer protocol status of the interfaces is Up

NOTE

When a switch uses the default CPCAR values (128 kbit/s on the MPU and 64 kbit/s on LPUs) to limit the rate of ARP request packets or ARP reply packets and the CPU usage is below 50%, the switch processes ARP packets at a rate of 100 pps if the minimum length of ARP request packets is 60 bytes. After CPCAR is canceled, the device learns 1000 ARP entries or so every second.

3.5.1 Adjusting Aging Parameters of Dynamic ARP Entries

Context

Aging parameters of ARP entries include the aging time, the number of probes, and detection modes. Proper adjustment of aging parameters improves network reliability.

You can adjust the following parameters of dynamic ARP entries:

- Aging time of dynamic ARP entries: When the aging time of a dynamic ARP entry is reached, the device sends an ARP Request packet to the corresponding outbound interface and starts ARP aging detection.If the ARP aging time is too short, the ARP request packets are sent frequently. The default aging time 20 minutes is recommended.
- Number of probes to dynamic ARP entries: Before aging a dynamic ARP entry, the system first performs probes. If no answer is received after the times of probes reach the upper limit, the ARP entry is deleted.
- Aging detection modes of dynamic ARP entries: Before an ARP entry is aged, an interface sends an ARP aging probe packet.

  NOTE

  - If the IP address of the peer device remains the same but the MAC address changes frequently, it is recommended that you configure ARP aging probe packets to be broadcast.
  - If the MAC address of the peer device remains the same, the network bandwidth is insufficient, and the aging time of ARP entries is short, it is recommended that you configure ARP aging probe packets to be unicast.

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:

```
arp expire-time expire-time
```

The aging time of dynamic ARP entries is set.

By default, the aging time of dynamic ARP entries is 1200 seconds, that is, 20 minutes.

Step 4 Run:

```
arp detect-times detect-times
```

The number of probes to dynamic ARP entries is set.

By default, the number of ARP probes is 3.

Step 5 Run:

```
arp detect-mode unicast
```

An interface is configured to send ARP aging probe packets in unicast mode.

By default, an interface sends ARP aging probe packets in broadcast mode.

----End

3.5.2 Enabling ARP Suppression Function

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
arp-suppress enable
```

ARP suppression is enabled on the current device.

By default, ARP suppression is disabled but is enabled on VLANIF interfaces.

----End

3.5.3 Enabling Layer 2 Topology Detection

Context

Layer 2 topology detection enables the system to update all the ARP entries in the VLAN that a Layer 2 interface belongs to when the Layer 2 interface status changes from Down to Up.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
l2-topology detect enable
```

Layer 2 topology detection is enabled.

By default, Layer 2 topology detection is disabled.

----End

3.5.4 Checking the Configuration

Procedure

- Run the **display arp interface** *interface-type interface-number* [*.subinterface-number*] [**vid** *vlan-id* [**cevid** *cevlan-id*]] command to check ARP mapping entries of a specified interface.
- Run the **display arp vpn-instance** *vpn-instance-name* [**dynamic** | **static**] command to check ARP mapping entries of a specified VPN instance.
- Run the **display arp statistics** { **all** | **interface** *interface-type interface-number* [*.subinterface-number*] } command to check statistics on ARP entries.

3.6 Configuring Proxy ARP

The switch can function as a proxy of the destination host to reply an ARP Request message.

Pre-configuration Tasks

Before configuring proxy ARP, complete the following task:

- Setting link layer protocol parameters for interfaces to ensure that the link layer protocol status of the interfaces is Up

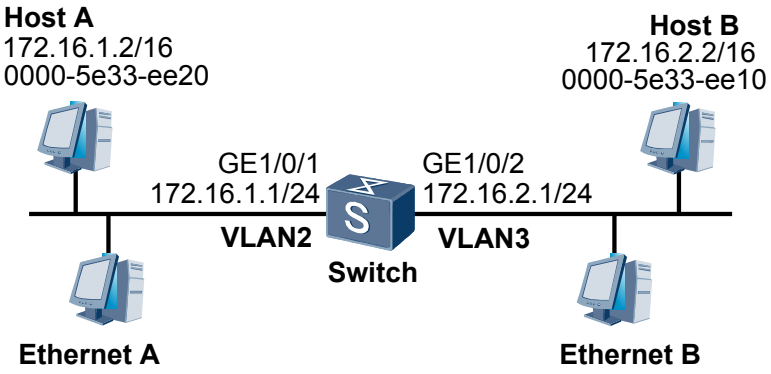
3.6.1 Configuring Routed Proxy ARP

Context

Proxy ARP enables PCs or switches on the same network segment but on different physical networks to communicate. In actual applications, if the current host connected to the switch is not configured with a default gateway address (that is, the host does not know how to reach the intermediate system of the network), the host cannot forward data packets. Routed proxy ARP solves this problem.

Figure 3-3 shows the routed proxy ARP networking. SwitchA uses GE1/0/1 and GE1/0/2 to connect two networks. IP addresses of the two GE interfaces are on different network segments. However, the masks make Host A and VLANIF10 on the same network segment, Host BSTA2 and VLANIF20 on the same network segment, and Host A and Host BSTA2 on the same network segment.

Figure 3-3 Networking diagram for configuring routed proxy ARP



HOSTA sends an ARP Request packet, requesting the MAC address of HOSTBSTA2. After receiving the packet, SwitchA uses its MAC address to reply the Request packet. HOSTA then forwards data using the MAC address of SwitchA.

CAUTION

IP addresses of the STAhosts on a subnet have the same network ID. Therefore, the default gateway address does not need to be configured on the STAhosts.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

On the device, routing proxy ARP can only be enabled on VLANIF interfaces.

Step 3 Run:

```
ip address ip-address { mask | mask-length }
```

IP addresses are configured for interfaces.

The IP address configured for the interface enabled with routed proxy ARP must be on the same network segment as the IP address of the connected hostserver on a LAN.

Step 4 Run:

```
arp-proxy enable
```

Routed proxy ARP is enabled on the interface.

After proxy ARP is enabled, the aging time of ARP entries on hosts should be shortened so that invalid ARP entries can be deleted as soon as possible. The number of packets received but cannot be forwarded by the device is decreased. To set ARP aging time, run the **arp expire-time** *expire-time* command.

----End

Checking the Configuration

After configuring routed proxy ARP is complete, run the following commands to check the configuration.

- Run the **display arp interface** *interface-type interface-number* [*.subinterface-number*] [**vid** *vlan-id* [**cevid** *cevlan-id*]] command to check ARP mapping entries of a specified interface.
- Run the **display arp vpn-instance** *vpn-instance-name* [**dynamic** | **static**] command to check ARP mapping entries of a specified VPN instance.
- Run the **display arp statistics** { **all** | **interface** *interface-type interface-number* [*.subinterface-number*] } command to check statistics on ARP entries.

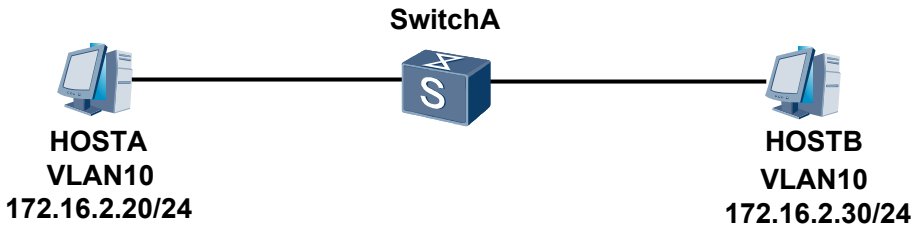
3.6.2 Configuring Intra-VLAN Proxy ARP

Context

If two hosts belong to the same VLAN but are isolated, enable intra-VLAN proxy ARP on an interface associated with the VLAN to allow the hosts to communicate.

As shown in **Figure 3-4**, HOSTA and HOSTB connect to SwitchA. The two interfaces that connect HOSTA and HOSTB to SwitchA belong to VLAN10.

Figure 3-4 Intra-VLAN proxy ARP application



HOSTA and HOSTB cannot communicate at Layer 2 because interface isolation in a VLAN is configured on SwitchA.

To solve this problem, enable intra-VLAN proxy ARP on the interfaces of SwitchA. After an interface of SwitchA receives an ARP Request packet whose destination address is HOSTB and source address is HOSTA, SwitchA does not discard the packet but searches for the ARP entry. If the ARP entry matching HOSTB exists, SwitchA sends its MAC address to HOSTA and forwards packets sent from HOSTA to HOSTB. SwitchA functions as the proxy of HOSTB.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

On the device, Intra-VLAN Proxy ARP can only be enabled on VLANIF interfaces.

Step 3 Run:

```
arp-proxy inner-sub-vlan-proxy enable
```

Intra-VLAN proxy ARP is enabled.

----End

Checking the Configuration

After configuring intra-VLAN proxy ARP is complete, run the following commands to check the configuration.

- Run the **display arp interface** *interface-type interface-number* [*.subinterface-number*] [**vid** *vlan-id* [**cevid** *cevlan-id*]] command to check ARP mapping entries of a specified interface.
- Run the **display arp vpn-instance** *vpn-instance-name* [**dynamic** | **static**] command to check ARP mapping entries of a specified VPN instance.
- Run the **display arp statistics** { **all** | **interface** *interface-type interface-number* [*.subinterface-number*] } command to check statistics on ARP entries.

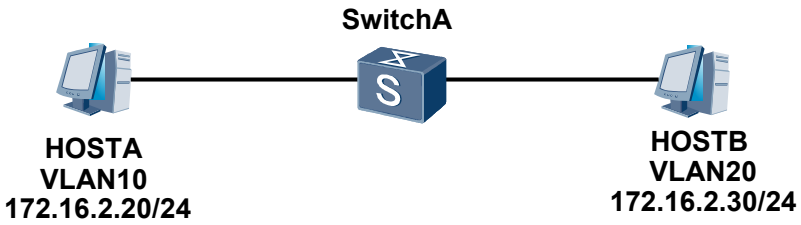
3.6.3 Configuring Inter-VLAN Proxy ARP

Context

If two hosts belong to different VLANs, enable inter-VLAN proxy ARP on interfaces associated with the VLANs to implement Layer 3 communication between the two hosts.

As shown in **Figure 3-5**, HOSTA and HOSTB connect to SwitchA. Interfaces that connect HOSTA and HOSTB to SwitchA belong to VLAN10 and VLAN20 respectively.

Figure 3-5 Inter-VLAN proxy ARP application



Interfaces connecting HOSTA and HOSTB to SwitchA belong to different VLANs. Therefore, HOSTA and HOSTB cannot communicate at Layer 2.

To solve this problem, inter-VLAN proxy ARP needs to be enabled on interfaces of SwitchA. After an interface of SwitchA receives an ARP Request packet whose destination address is HOSTB and source address is HOSTA, SwitchA does not discard the packet but searches for the ARP entry. If the ARP entry matching HOSTB exists, SwitchA sends its MAC address to HOSTA and forwards packets sent from HOSTA to HOSTB. SwitchA functions as the proxy of HOSTB.

Inter-VLAN proxy ARP implements the following functions:

- Allows users in different VLANs to communicate at Layer 3.
- Allows users in different sub-VLANs to communicate. You need to enable inter-VLAN proxy ARP on the VLANIF interface of the super-VLAN.

Procedure

Step 1 Run:

```
system-view
```

Enter the system view.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

On the device, Inter-VLAN Proxy ARP can only be enabled on VLANIF interfaces.

Step 3 Run:

```
arp-proxy inter-sub-vlan-proxy enable
```

Inter-VLAN proxy ARP is enabled.

----End

Checking the Configuration

After configuring inter-VLAN proxy ARP is complete, run the following commands to check the configuration.

- Run the **display arp interface** *interface-type interface-number* [*.subinterface-number*] [**vid** *vlan-id* [**cevid** *cevlan-id*]] command to check ARP mapping entries of a specified interface.
- Run the **display arp vpn-instance** *vpn-instance-name* [**dynamic** | **static**] command to check ARP mapping entries of a specified VPN instance.
- Run the **display arp statistics** { **all** | **interface** *interface-type interface-number* [*.subinterface-number*] } command to check statistics on ARP entries.

3.7 Configuring ARP-Ping

ARP-Ping includes ARP-Ping IP and ARP-Ping MAC. ARP-Ping sends ARP Request packets or ICMP Echo Request packets to check whether a specified IP address or MAC address is used.

Pre-configuration Tasks

Before configuring ARP-Ping, complete the following task:

- Configuring link layer protocol parameters and IP addresses for interfaces to ensure that the link layer protocol status of the interfaces is Up.

3.7.1 Configuring ARP-Ping IP

Context

Before configuring an IP address for a device on a LAN, run the **arp-ping ip** command to check whether the IP address is used by other network devices.

The **ping** command can also check whether an IP address is in use. If the destination host or the switch configured with the firewall function are configured not to reply to ping packets, there is no response to the ping packet. Consequently, the IP address is considered unused. ARP is a Layer 2 protocol. In most cases, ARP packets can pass through the firewall that is disabled from replying to the Ping packets to prevent the preceding situation.

Procedure

Step 1 Run:

```
arp-ping ip ip-address [ interface interface-type interface-number ]
```

Check whether the IP address is used.

- If the following information is displayed, the IP address is not used.

```
[Quidway] arp-ping ip 110.1.1.2
 ARP-Pinging 110.1.1.2:

Error: Request timed out.
Error: Request timed out.
Error: Request timed out.
Info: The IP address is not used by anyone!
```

- If the following information is displayed, the IP address is used.

```
[Quidway] arp-ping ip 128.1.1.1
 ARP-Pinging 128.1.1.1:
128.1.1.1 is used by 00e0-517d-f202
```

----End

3.7.2 Configuring ARP-Ping MAC

Context

When you know a specific MAC address but not the corresponding IP address on a network segment, you can obtain the corresponding IP address using the **arp-ping mac** command to send ICMP packets. In this way, you can obtain the IP address mapping the MAC address.

Procedure

Step 1 Run:

```
arp-ping mac mac-address { ip-address [ vpn-instance vpn-instance-name ] |
interface interface-type interface-number }
```

Check whether the MAC address is used. If the MAC address is in use, query the IP address mapping the MAC address.

- If the following information is displayed, the MAC address is not used.

```
[Quidway] arp-ping mac 00e0-517d-f201 interface gigabitethernet 1/0/1
  OutInterface: GigabitEthernet1/0/0 MAC[00-E0-51-7D-F2-01], press CTRL_C to
break
Error: Request timed out.
Error: Request timed out.
Error: Request timed out.

    ----- ARP-Ping MAC statistics -----
    3 packet(s) transmitted
```

```
    0 packet(s) received
    MAC[00-E0-51-7D-F2-01]  not be used
```

- If the following information is displayed, it means that the MAC address is used.

```
[Quidway] arp-ping mac 00e0-517d-f202 interface gigabitethernet 1/0/1
  OutInterface: GigabitEthernet1/0/0 MAC[00-E0-51-7D-F2-02], press CTRL_C to
break
    ----- ARP-Ping MAC statistics -----
    1 packet(s) transmitted
    1 packet(s) received
    IP ADDRESS               MAC ADDRESS
    128.1.1.1                00-E0-51-7D-F2-02
```

----End

3.8 Enabling a Device to Learn Multicast MAC Addresses and Generate ARP Entries

If a device is enabled to learn multicast MAC addresses, it can generate ARP entries after receiving ARP packets carrying multicast MAC addresses as source MAC addresses. This section describes how to enable a device to learn multicast MAC addresses and generate ARP entries.

Background Information

A MAC address corresponding to an IP address may be a multicast MAC address. In this case, a network administrator has to configure a static ARP entry. A device can generate dynamic ARP entries if enabled to learn multicast MAC addresses. This way reduces a network administrator's workload of configuring static ARP entries and reduces network operation and maintenance costs.

Procedure

- Globally enable a device to learn multicast MAC addresses and generate dynamic ARP entries.
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     arp learning multicast enable

     The device is globally enabled to learn multicast MAC addresses.

     By default, a device is globally disabled from learning multicast MAC addresses.

     NOTE

     If a device is globally enabled to learn multicast MAC addresses, the interfaces of this device are enabled to learn multicast MAC addresses.
- Enable an interface to learn multicast MAC addresses.
  1. Run:
     system-view

The system view is displayed.

2. Run:
   interface vlanif *interface-number*

   The interface view is displayed.

3. Run:
   arp learning multicast enable

   The interface is enabled to learn multicast MAC addresses.

   By default, if a device is globally enabled to learn multicast MAC addresses, all the device interfaces are enabled to learn multicast MAC addresses. If a device is globally disabled from learning multicast MAC addresses, all the device interfaces are disabled from learning multicast MAC addresses.

   NOTE

   If the **undo arp learning multicast enable** command is run on a specific interface, the interface is disabled from learning multicast MAC addresses but uses the global configuration in the configuration file. If the device is globally enabled to learn multicast MAC addresses, the interface still has this function enabled. If the device is globally disabled from learning multicast MAC addresses, the interface has this function disabled.

- Disable an interface from learning multicast MAC addresses after a device has been globally enabled to learn multicast MAC addresses. The interface does not use the global configuration.

  1. Run:
     system-view

     The system view is displayed.

  2. Run:
     interface vlanif *interface-number*

     The interface view is displayed.

  3. Run:
     arp learning multicast disable

     The interface is disabled from learning multicast MAC addresses. This configuration is saved in the configuration file.

----End

3.9 Configuring ARP Automatic Scanning and Fixed ARP

ARP automatic scanning and fixed ARP enable a device to generate dynamic ARP entries and convert the dynamic ARP entries to static ARP entries.

Background Information

To improve communications security, network administrators generally configure static ARP entries on a small-sized LAN. However, if a gateway has multiple users attached, a network administrator has to configure static ARP entries for each user. Current networks use dynamic ARP for communication.

Dynamic ARP helps reduce a network administrator's workload but has its own limitations. Dynamic ARP entries can be overwritten by subsequent ARP entries and are vulnerable to network attacks. Therefore, dynamic ARP cannot provide reliability for network communications.

ARP automatic scanning is generally used with fixed ARP to defend against network attacks:

- After ARP automatic scanning is configured, a device automatically scans all its neighbor devices on a LAN. The device sends ARP Request packets to its neighbor devices, obtains the MAC addresses of its neighbor devices, and generates dynamic ARP entries.
- After fixed ARP is configured, the device converts these dynamic ARP entries to static ARP entries.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vlanif interface-number
```

The VLANIF interface view is displayed.

Step 3 Run:

```
arp scan [ start-ip-address to end-ip-address ]
```

ARP automatic scanning is configured.

Step 4 Run:

```
arp fixup
```

Fixed ARP is configured.

----End

3.10 Configuring the function to detect IP address conflicts

After the function to detect IP address conflicts is enabled, the device will detect the IP address conflicts by ARP packets.

Context

When an IP address conflict occurs between network devices, it causes high CPU usage and route flapping. User services will be affected and even interrupted.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

arp ip-conflict-detect enable

The function to detect IP address conflicts is enabled.

----End

Checking the Configuration

Run the **display arp ip-conflict track** command, you can view the recorded information about the detected IP address conflict.

3.11 Configuring Egress ARP Inspection

Egress ARP inspection enables the switch to restrict the scope of ARP packet forwarding. This function prevents broadcast of ARP packets in a VLAN and reduces the traffic volume in the VLAN.

Context

Egress ARP inspection (EAI) applies to the following scenario: a switch is deployed between an upstream Layer 3 switch and user hosts. The user hosts belong to the same VLAN and connect to the network through user-side interfaces of the switch. Users obtain IP addresses through DHCP.

If the switch broadcasts ARP Request packets in the VLAN, the traffic volume in the VLAN increases. To reduce network loads in the VLAN, enable EAI in this VLAN on the switch. Before enabling EAI in a VLAN, run the **dhcp snooping enable** command to enable DHCP snooping globally.

After EAI is enabled, the switch matches destination IP addresses of received ARP Request packets with dynamic binding entries generated by DHCP snooping to determine outbound interfaces for the packets. If the destination IP address of an ARP Request packet matches a dynamic binding entry, the switch sends the packet to the outbound interface specified in the binding entry. If the destination IP address matches a static binding entry and no outbound interface is specified in the static binding entry, the switch searches the dynamic MAC address table for the outbound interface according to the MAC address in the static binding entry. If the ARP Request packet matches no binding entry, the switch processes the packet as follows:

- If the ARP Request packet is sent from the Layer 3 switch, the switch drops the packet.
- If the ARP Request packet is sent from a user host, the switch forwards the packet to the trusted interface.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

vlan *vlan-id*

The VLAN view is displayed.

Step 3 Run:

```
dhcp snooping arp security enable
```

EAI is enabled in the VLAN.

By default, EAI is disabled.

----End

3.12 Configuring a Static Unicast or Multicast MAC Address

After a static unicast or multicast MAC address is configured on an interface, unicast or multicast packets destined for the unicast or multicast MAC address are forwarded only to the interface.

Pre-configuration Tasks

Before configuring a static unicast or multicast MAC address, complete the following task:

- Creating a VLAN and adding the interface that needs to have a static unicast or multicast MAC address configured to the VLAN

Context

When an NLB cluster works in unicast or multicast mode, the Layer 3 gateway of NLB servers needs to send unicast packets to multiple outbound interfaces. In this situation, you need to run the **mac-address multiport interface** command to configure static MAC entries for the multiple outbound interfaces and then run the **arp static** command to configure static ARP entries for the multiple outbound interfaces. After static ARP entries are configured, you can determine the MAC address and VLAN of a host IP address and then search for the MAC address table based on the MAC address and VLAN to determine the outbound interface so that the host can connect to the servers in the NLB cluster.

After static unicast or multicast MAC address entries are configured, multicast packets destined for a multicast MAC address are forwarded only to the interfaces that have this unicast or multicast MAC address configured in a VLAN.

NOTE

The VLAN cannot be a MAC VLAN, super-VLAN, leased line VLAN, control VLAN of a Smart Ethernet Protocol (SEP) segment, or control VLAN of a Rapid Ring Protection Protocol (RRPP) ring.

S7700:ES0D0G24SA00, ES0D0G24CA00, ES0D0X12SA00 cards can only connect to server cluster, but cannot connect to user devices.

S9700:EH1D2G24SSA0, EH1D2S24CSA0, EH1D2X12SSA0 cards can only connect to server cluster, but cannot connect to user devices.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Configure a static unicast or multicast MAC address.

- Configure a static unicast or multicast MAC address on an interface.

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

   The interface view is displayed.
2. Run:
   mac-address multiport *mac-address* **vlan** *vlan-id*

   A static unicast or multicast MAC address is configured on the interface.

- Configure a static unicast or multicast MAC address on multiple interfaces.

  Run:

  mac-address multiport *mac-address* **interface** { **interface-type** *interface-number1* [**to interface-type** *interface-number2*] } &<1-10> **vlan** *vlan-id*

  A static unicast or multicast MAC address is configured on multiple interfaces.

  Interface numbers must be consecutive, and specified interfaces must be on the same board.*interface-number2* must be larger than *interface-number1*.

  NOTE

  The outbound interface cannot be an Eth-Trunk interface.

Step 3 Run:

arp static *ip-address mac-address* [**vpn-instance** *vpn-instance-name*] or **arp static** *ip-address mac-address* **vid** *vlan-id* [**cevid** *ce-vid*] **interface** *interface-type interface-number* [*.subinterface-number*]

A static ARP entry is configured.

mac-address must be the same as the unicast or multicast MAC address configured in step 2.

----End

Checking the Configuration

After the preceding configurations are complete, run the following commands in any view to check the configuration of the static unicast or multicast MAC address.

- Run the **display mac-address multiport** *mac-address* **vlan** *vlan-id* or **display mac-address multiport** [**vlan** *vlan-id*] [**total-number**] command to check configured static unicast or multicast MAC entries.

3.13 Configuring ARP Packet Forwarding Between Isolated Interfaces

After ARP packet forwarding between isolated interfaces is configured, the device that has EAI enabled forwards packets to trusted interfaces when port isolation is enabled on both inbound and outbound interfaces. This allows isolated users to communicate.

Prerequisites

The device has been enabled with **3.11 Configuring Egress ARP Inspection**.

Context

ARP packet forwarding between isolated interfaces applies to the following scenario: a device is deployed between an upstream Layer 3 switch and user hosts. User hosts belong to the same

VLAN and connect to the network through user-side interfaces of the device. Users obtain IP addresses through DHCP. Port isolation is enabled on the user-side interfaces. Intra-VLAN proxy ARP is enabled on the upstream Layer 3 switch to enable users in a VLAN to be isolated at Layer 2 and to communicate at Layer 3.

After receiving ARP Request packets sent from a user host to another user host, the device that has EAI enabled matches destination IP addresses of received ARP Request packets with dynamic binding entries generated by DHCP snooping to determine outbound interfaces for the packets. If the destination IP address of an ARP Request packet matches a dynamic binding entry, the device sends the packet to the destination interface which is the interface connecting the requested user host to the network. If the destination interface and the inbound interface of the ARP Request packet are configured with port isolation, the ARP Request packet is discarded. Users that are isolated cannot communicate.

To address this problem, the function of ARP packet forwarding between isolated interfaces is enabled on the device. The device then forwards ARP Request packets to a trusted interface which is the interface connecting the device to an upstream Layer 3 switch. Enabling intra-VLAN ARP proxy on the Layer 3 switch allows isolated users to communicate with each other.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
vlan vlan-id
```

The VLAN view is displayed.

Step 3 Run:

```
dhcp snooping arp security isolate-forwarding-trust
```

The device is enabled to forward ARP packets to trusted interfaces when port isolation is enabled on both inbound and outbound interfaces.

By default, the device does not forward ARP packets to trusted interfaces when port isolation is enabled on both inbound and outbound interfaces.

----End

3.14 Maintaining ARP

Maintaining ARP includes clearing ARP entries and monitoring ARP running status.

3.14.1 Clearing ARP Entries

Context

CAUTION

- After ARP entries are cleared, mappings between IP addresses and MAC addresses are deleted. As a result, users may not access specified nodes. Exercise caution when you clear ARP entries.
- Static ARP entries cannot be restored after being cleared. Exercise caution when you clear static ARP entries.

Procedure

- Run the **reset arp** { **all** | **dynamic** [**ip** *ip-address* [**vpn-instance** *vpn-instance-name*]] | **interface** *interface-type interface-number* [*.subinterface-number*] [**ip** *ip-address*] | **static** } command to clear ARP entries in the ARP mapping table and clear the statistics information of related packets.
- Run the **reset arp packet statistics** command to clear the statistics on ARP packets.

----End

3.14.2 Monitoring the ARP Running Status

Context

Monitoring the ARP running status includes checking ARP mapping entries, strict ARP entry learning, ARP packet statistics, ARP packet processing rate, and maximum number of ARP entries learnt by an interface.

Procedure

- Run the **display arp packet statistics** command in any view to check ARP packet statistics.
- Run the **display arp interface** *interface-type interface-number* [*.subinterface-number*] [**vid** *vlan-id* [**cevid** *cevlan-id*]] command in any view to check the information about the ARP mapping table based on interfaces.
- Run the **display arp vpn-instance** *vpn-instance-name*[**dynamic** | **static**] command in any view to check the information about ARP mapping tables based on VPN instances.
- Run the **display arp error packet** command in any view to check the latest ten ARP error packets.
- Run the **display arp dynamic** command in any view to check all dynamic ARP entries.
- Run the **display arp dynamic vlan** command in any view to check dynamic ARP entries that a device has generated on a specified VLAN.
- Run the **display arp network** command in any view to check ARP mapping entries of a specified network segment.
- Run the **display arp static** command in any view to check all static ARP mapping entries.

- Run the **display arp status** command in any view to check the delivery state of ARP entries on the LPU.
- Run the **display arp track** command in any view to check changes of outbound interfaces in ARP entries learned by a VLANIF interface.
- Run the **display arp all** command in any view to check ARP mapping entries of all interface boards.

----End

3.15 Configuration Examples

This section provides configuration examples including networking requirements and configuration roadmap.

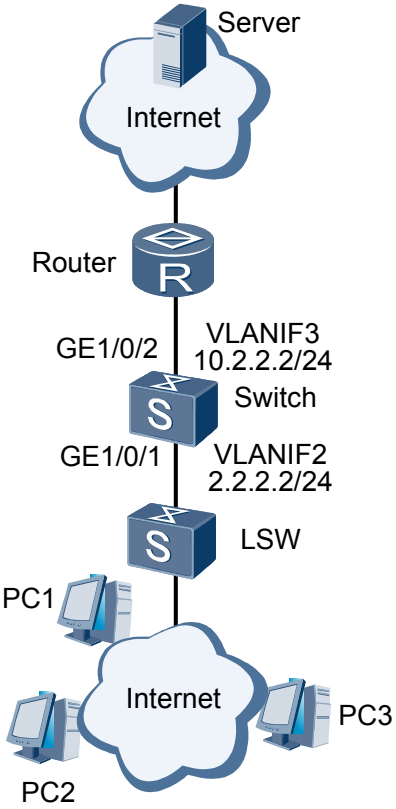
3.15.1 Example for Configuring ARP

Networking Requirements

As shown in **Figure 3-6**, GE1/0/1 on the switch connects to hosts through the LAN Switch (LSW). GE1/0/2 connects to a server through the Router. Requirements are as follows:

- GE1/0/1 belongs to VLAN2 and GE1/0/2 belongs to VLAN3.
- Dynamic ARP parameters should be configured for VLANIF2 of the switch so that packets are transmitted correctly regardless of network typology change.
- A static ARP entry should be configured on GE1/0/2 of the switch to ensure secure communication with the server and prevent illegal ARP packets. The IP address of the router should be 10.2.2.3 and the corresponding MAC address is 00e0-fc01-0000.

Figure 3-6 Networking diagram for configuring ARP



Configuration Roadmap

The configuration roadmap is as follows:

1. Create VLANs and add interfaces to the VLANs.
2. Set dynamic ARP parameters for the user-side VLANIF interface.
3. Configure a static ARP entry.

Procedure

Step 1 Create VLANs and add interfaces to the VLANs.

Create VLAN2 and VLAN3.

```
<Quidway> system-view
[Quidway] vlan batch 2 3
```

Add GE1/0/1 to VLAN2 and GE1/0/2 to VLAN3.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port hybrid tagged vlan 2
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port hybrid tagged vlan 3
[Quidway-GigabitEthernet1/0/2] quit
```

Step 2 Set dynamic ARP parameters for the VLANIF interface.

Create VLANIF2.

```
[Quidway] interface vlanif 2
```

Configure an IP address for VLANIF2.

```
[Quidway-Vlanif2] ip address 2.2.2.2 255.255.255.0
```

Set the aging time of ARP entries to 60s.

```
[Quidway-Vlanif2] arp expire-time 60
```

Set the number of probes to ARP entries to 2.

```
[Quidway-Vlanif2] arp detect-times 2
[Quidway-Vlanif2] quit
```

Create VLANIF3.

```
[Quidway] interface vlanif 3
```

Configure an IP address for VLANIF3.

```
[Quidway-Vlanif3] ip address 10.2.2.2 255.255.255.0
[Quidway-Vlanif3] quit
```

Step 3 Configure a static ARP entry.

Configure a static ARP entry with IP address 10.2.2.3, MAC address 00e0-fc01-0000, VLAN ID 3, and outbound interface GE1/0/2.

```
[Quidway] arp static 10.2.2.3 00e0-fc01-0000 vid 3 interface gigabitethernet 1/0/2
[Quidway] quit
```

Step 4 Verify the configuration.

Run the **display current-configuration** command to check the aging time, number of probes, and ARP mapping entries.

```
<Quidway> display current-configuration | include arp
 arp detect-times 2
 arp expire-time 60
 arp static 10.2.2.3 00e0-fc01-0000 vid 3 interface GigabitEthernet1/0/2
```

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
vlan batch 2 to 3
#
interface Vlanif2
 arp detect-times 2
 arp expire-time 60
 ip address 2.2.2.2 255.255.255.0
#
interface Vlanif3
 ip address 10.2.2.2 255.255.255.0
#
interface GigabitEthernet1/0/1
```

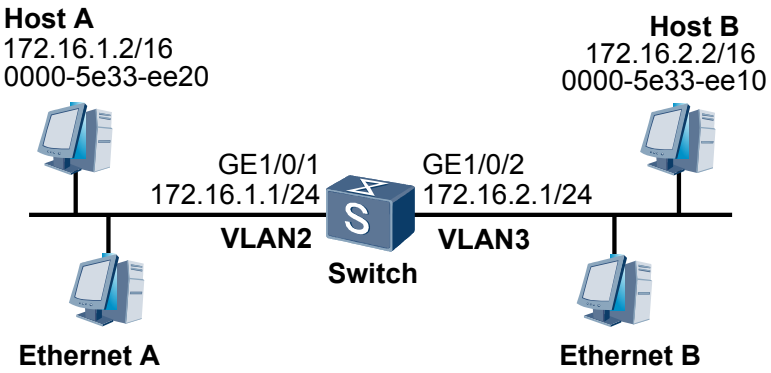
```
 port hybrid tagged vlan 2
#
interface GigabitEthernet1/0/2
 port hybrid tagged vlan 3
#
 arp static 10.2.2.3 00e0-fc01-0000 vid 3 interface GigabitEthernet1/0/2
#
return
```

3.15.2 Example for Configuring Routed Proxy ARP

Networking Requirements

In **Figure 3-7**, Ethernet interfaces GE1/0/1 and GE1/0/2 connect to two LANs respectively. The two LANs are at the same network segment 172.16.0.0/16. HostA and HostB have no default gateway. Routed proxy ARP is required to be configured on the switch so that hosts on two LANs can communicate.

Figure 3-7 Networking diagram for configuring routed proxy ARP



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses for interfaces.
2. Enable routed proxy ARP on interfaces.

Procedure

Step 1 Create VLAN2 and add GE1/0/1 to VLAN2.

```
<Quidway> system-view
[Quidway] vlan 2
[Quidway-vlan2] quit
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port link-type access
[Quidway-GigabitEthernet1/0/1] port default vlan 2
[Quidway-GigabitEthernet1/0/1] quit
```

Step 2 Create and configure VLANIF2.

```
[Quidway] interface vlanif 2
[Quidway-Vlanif2] ip address 172.16.1.1 255.255.255.0
```

Step 3 Enable routed proxy ARP on VLANIF2.

```
[Quidway-Vlanif2] arp-proxy enable
[Quidway-Vlanif2] quit
```

Step 4 Create VLAN3 and add GE1/0/2 to VLAN3.

```
[Quidway] vlan 3
[Quidway-vlan3] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port link-type access
[Quidway-GigabitEthernet1/0/2] port default vlan 3
[Quidway-GigabitEthernet1/0/2] quit
```

Step 5 Create and configure VLANIF3.

```
[Quidway] interface vlanif 3
[Quidway-Vlanif3] ip address 172.16.2.1 255.255.255.0
```

Step 6 Enable routed proxy ARP on VLANIF3.

```
[Quidway-Vlanif3] arp-proxy enable
[Quidway-Vlanif3] quit
```

Step 7 Configure hosts.

Configure IP address 172.16.1.2/16 for HostA.

Configure IP address 172.16.2.2/16 for HostB.

Step 8 Verify the configuration.

Ping Host B from Host A. Host A can ping Host B successfully.

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
 vlan batch 2 to 3
#
interface Vlanif2
 ip address 172.16.1.1 255.255.255.0
 arp-proxy enable
#
interface Vlanif3
 ip address 172.16.2.1 255.255.255.0
 arp-proxy enable
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 2
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 3
#
return
```

3.15.3 Example for Configuring Intra-VLAN Proxy ARP

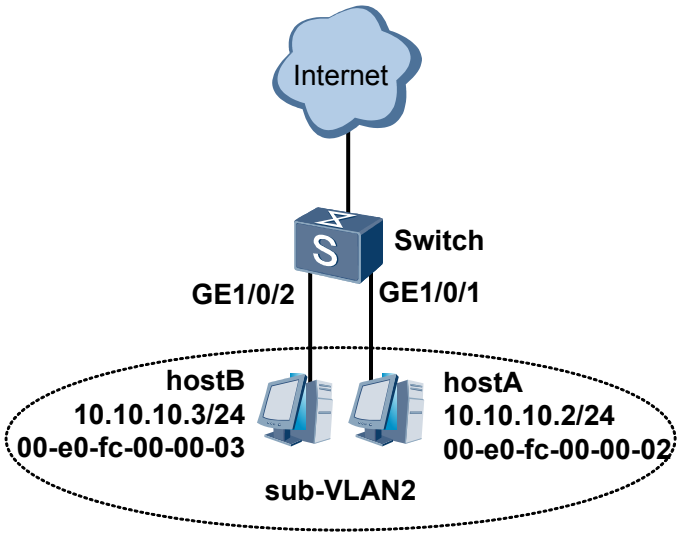
Networking Requirements

As shown in **Figure 3-8**, GE1/0/2 and GE1/0/1 on the switch belong to sub-VLAN2. Sub-VLAN2 belongs to super-VLAN3. Requirements are as follows:

- HostA and HostB in VLAN2 should be isolated at Layer 2.
- HostA and HostB can communicate at Layer 3 using intra-VLAN proxy ARP.

The IP address of the VLANIF interface corresponding to the super-VLAN is 10.10.10.1 and the mask is 255.255.255.0.

Figure 3-8 Networking diagram for configuring intra-VLAN proxy ARP



Configuration Roadmap

The configuration roadmap is as follows:

1. Create and configure a super-VLAN and a sub-VLAN.
2. Add interfaces to the sub-VLAN.
3. Create a VLANIF interface corresponding to the super-VLAN and assign an IP address to the VLANIF interface.
4. Enable intra-VLAN proxy ARP on the VLANIF interface.

Procedure

Step 1 Configure a super-VLAN and a sub-VLAN.

Configure sub-VLAN2.

```
<Quidway> system-view
[Quidway] vlan 2
[Quidway-vlan2] quit
```

Enable interface isolation on GE1/0/1 and GE1/0/2.

```
[Quidway] port-isolate mode l2
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port-isolate enable
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port-isolate enable
[Quidway-GigabitEthernet1/0/2] quit
```

Add GE1/0/1 and GE1/0/2 to sub-VLAN2.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port link-type access
[Quidway-GigabitEthernet1/0/1] port default vlan 2
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port link-type access
[Quidway-GigabitEthernet1/0/2] port default vlan 2
[Quidway-GigabitEthernet1/0/2] quit
```

Configure super-VLAN3 and add sub-VLAN2 to super-VLAN3.

```
[Quidway] vlan 3
[Quidway-vlan3] aggregate-vlan
[Quidway-vlan3] access-vlan 2
[Quidway-vlan3] quit
```

Step 2 Create and configure VLANIF3.

Create VLANIF3.

```
[Quidway] interface vlanif 3
```

Configure an IP address for VLANIF3.

```
[Quidway-Vlanif3] ip address 10.10.10.1 24
```

Step 3 Enable intra-VLAN proxy ARP on VLANIF3.

```
[Quidway-Vlanif3] arp-proxy inner-sub-vlan-proxy enable
[Quidway-Vlanif3] quit
```

Step 4 Verify the configuration.

Run the **display current-configuration** command to check configurations of the super-VLAN, sub-VLAN, and VLANIF interface. The output of the command is displayed in the following configuration file.

hostA and hostB can ping each other.

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
 vlan batch 2 to 3
#
vlan 3
 aggregate-vlan
 access-vlan 2
#
port-isolate mode l2
#
interface Vlanif3
```

```
 ip address 10.10.10.1 255.255.255.0
 arp-proxy inner-sub-vlan-proxy enable
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 2
 port-isolate enable group 1
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 2
 port-isolate enable group 1
#
return
```

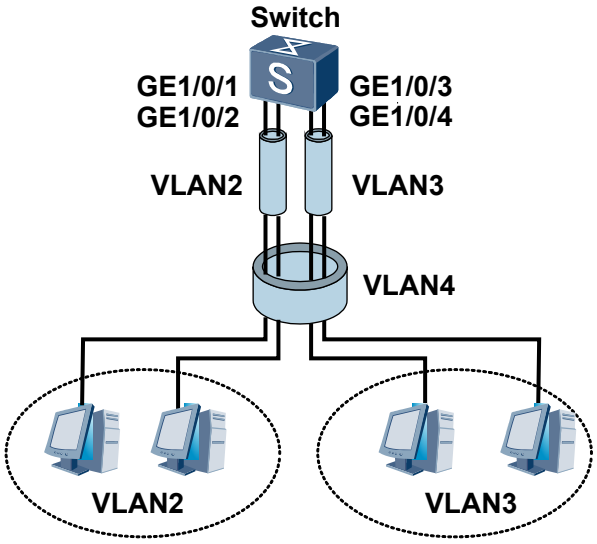
3.15.4 Example for Configuring Inter-VLAN Proxy ARP

Networking Requirements

As shown in **Figure 3-9**, VLAN2 and VLAN3 belong to super-VLAN4. Requirements are as follows:

- Hosts in VLAN2 and VLAN3 cannot ping each other.
- Hosts in VLAN2 and VLAN3 can communicate after inter-VLAN proxy ARP is configured.

Figure 3-9 Networking diagram for configuring inter-VLAN proxy ARP



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a super-VLAN and sub-VLANs.
2. Add interfaces to the sub-VLANs.

3. Create a VLANIF interface corresponding to the super-VLAN and assign an IP address to the VLANIF interface.
4. Enable inter-VLAN proxy ARP.

Procedure

Step 1 Configure a super-VLAN and sub-VLANs.

Configure sub-VLAN2.

```
<Quidway> system-view
[Quidway] vlan 2
[Quidway-vlan2] quit
```

Add GE1/0/1 and GE1/0/2 to sub-VLAN2.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port link-type access
[Quidway-GigabitEthernet1/0/1] port default vlan 2
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port link-type access
[Quidway-GigabitEthernet1/0/2] port default vlan 2
[Quidway-GigabitEthernet1/0/2] quit
```

Configure sub-VLAN3.

```
<Quidway> system-view
[Quidway] vlan 3
[Quidway-vlan3] quit
```

Add GE1/0/3 and GE1/0/4 to sub-VLAN3.

```
[Quidway] interface gigabitethernet 1/0/3
[Quidway-GigabitEthernet1/0/3] port link-type access
[Quidway-GigabitEthernet1/0/3] port default vlan 3
[Quidway-GigabitEthernet1/0/3] quit
[Quidway] interface gigabitethernet 1/0/4
[Quidway-GigabitEthernet1/0/4] port link-type access
[Quidway-GigabitEthernet1/0/4] port default vlan 3
[Quidway-GigabitEthernet1/0/4] quit
```

Configure super-VLAN4, then add sub-VLAN2 and sub-VLAN3 to super-VLAN4.

```
[Quidway] vlan 4
[Quidway-vlan4] aggregate-vlan
[Quidway-vlan4] access-vlan 2
[Quidway-vlan4] access-vlan 3
[Quidway-vlan4] quit
```

Step 2 Create and configure VLANIF4.

Create VLANIF4.

```
[Quidway] interface vlanif 4
```

Configure an IP address for VLANIF4.

```
[Quidway-Vlanif4] ip address 10.10.10.1 24
```

Step 3 Enable inter-VLAN proxy ARP on VLANIF4.

```
[Quidway-Vlanif4] arp-proxy inter-sub-vlan-proxy enable
[Quidway-Vlanif4] quit
```

Step 4 Verify the configuration.

Run the **display current-configuration** command to check configurations of the super-VLAN, sub-VLANs, and VLANIF interface. The output of the command is displayed in the following configuration file.

Hosts in VLAN2 and VLAN3 can communicate after inter-VLAN proxy ARP is configured.

----End

Configuration Files

Configuration file of the switch

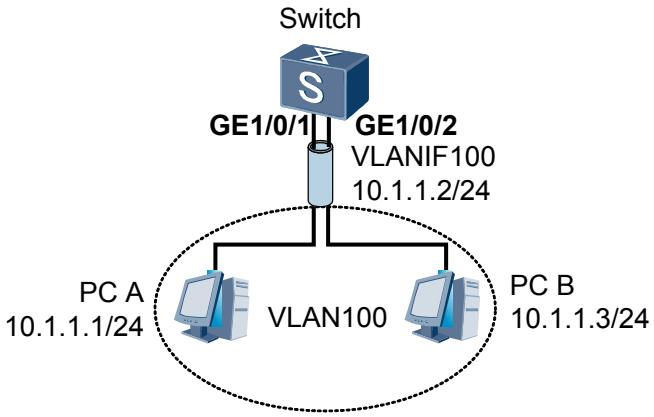
```
#
 sysname Quidway
#
 vlan batch 2 to 4
#
vlan 4
 aggregate-vlan
 access-vlan 2 3
#
interface Vlanif4
 ip address 10.10.10.1 255.255.255.0
 arp-proxy inter-sub-vlan-proxy enable
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 2
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 2
#
interface GigabitEthernet1/0/3
 port link-type access
 port default vlan 3
#
interface GigabitEthernet1/0/4
 port link-type access
 port default vlan 3
#
return
```

3.15.5 Example for Configuring Layer 2 Topology Detection

Networking Requirements

As shown in **Figure 3-10**, two GE interfaces are added to VLAN100. IP addresses of the switch that two GE interfaces connect.

Figure 3-10 Networking diagram for configuring Layer 2 topology detection



Configuration Roadmap

The configuration roadmap is as follows:

1. Add two GE interfaces to VLAN100.
2. Enable Layer 2 topology detection to view changes of ARP entries.

Procedure

Step 1 Create VLAN100 and add two GE interfaces on the switch to VLAN100.

Create VLAN100 and configure an IP address for the VLANIF interface.

```
<Quidway> system-view
[Quidway] vlan 100
[Quidway-vlan100] quit
[Quidway] interface vlanif 100
[Quidway-Vlanif100] ip address 10.1.1.2 24
[Quidway-Vlanif100] quit
```

Add two GE interfaces to VLAN100.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port link-type access
[Quidway-GigabitEthernet1/0/1] port default vlan 100
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port link-type access
[Quidway-GigabitEthernet1/0/2] port default vlan 100
[Quidway-GigabitEthernet1/0/2] quit
```

Step 2 Enable Layer 2 topology detection.

```
[Quidway] l2-topology detect enable
```

Step 3 Restart GE1/0/1 and view changes of ARP entries and aging time.

View ARP entries on the switch. You can find the switch has learnt the MAC address of the PC.

```
[Quidway] display arp all
IP ADDRESS      MAC ADDRESS     EXPIRE(M)  TYPE          INTERFACE   VPN-
INSTANCE
```

```
                                          VLAN/CEVLAN
--------------------------------------------------------------------------------
10.1.1.2        00e0-c01a-4900              I -           Vlanif100
10.1.1.1        00e0-c01a-4901  20          D-0           GE1/0/1
                                            100/-
10.1.1.3        00e0-de24-bf04  20          D-0           GE1/0/2
                                            100/-
--------------------------------------------------------------------------------
Total:3         Dynamic:2       Static:0    Interface:1
```

Run the **shutdown** and **undo shutdown** commands on GE1/0/1 and view the aging time of ARP entries.

- Run the **shutdown** command on GE1/0/1 to view the aging time of ARP entries.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] shutdown
[Quidway-GigabitEthernet1/0/1] display arp all
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE          INTERFACE   VPN-
INSTANCE
                                          VLAN/CEVLAN
--------------------------------------------------------------------------------
10.1.1.2        00e0-c01a-4900              I -
Vlanif100
10.1.1.3        00e0-de24-bf04  18          D-0           GE1/0/2
                                            100/-
--------------------------------------------------------------------------------
Total:2         Dynamic:1       Static:0    Interface:1
```

- Run the **undo shutdown** command on GE1/0/1 to view the aging time of ARP entries.

```
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] undo shutdown
[Quidway-GigabitEthernet1/0/1] display arp all
IP ADDRESS      MAC ADDRESS     EXPIRE(M)  TYPE          INTERFACE   VPN-
INSTANCE
                                          VLAN/CEVLAN
--------------------------------------------------------------------------------
10.1.1.2        00e0-c01a-4900              I -           Vlanif100
10.1.1.1        00e0-c01a-4901  20          D-0           GE1/0/1
                                            100/-
10.1.1.3        00e0-de24-bf04  20          D-0           GE1/0/2
                                            100/-
--------------------------------------------------------------------------------
Total:3         Dynamic:2       Static:0    Interface:1
```

NOTE

The preceding command output shows that the ARP entries learned from GE 1/0/1 are deleted after GE 1/0/1 is shut down. After the undo shutdown command is run on GE 1/0/1 and GE 1/0/1 goes Up, the ARP entry learned from GE 1/0/2 is aged, and then the device sends an ARP probe packet for updating ARP entry. After the entry is updated, the aging time restores the default value, 20 minutes.

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
L2-topology detect enable
#
 vlan batch 100
#
interface Vlanif100
 ip address 10.1.1.2 255.255.255.0
```

```
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 100
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 100
#
return
```

4 DHCP Configuration

About This Chapter

Dynamic Host Configuration Protocol (DHCP) dynamically manages and configures clients in a concentrated manner. It ensures proper IP address allocation and improves IP address use efficiency.

4.9 Common Configuration Errors

This section provides DHCP troubleshooting procedures.

4.1 DHCP Overview

Dynamic Host Configuration Protocol (DHCP) dynamically manages and configures clients in a centralized manner. DHCP uses the client/server model. A client applies to the server for configurations such as the IP address, subnet mask, and default gateway; the server replies with requested configurations based on policies.

As the network expands and becomes complex, the number of hosts often exceeds the number of available IP addresses. As portable computers and wireless networks are widely used, the positions of computers often change, causing IP addresses of the computers to be changed accordingly. As a result, network configurations become increasingly complex. To properly and dynamically assign IP addresses to hosts, DHCP is used.

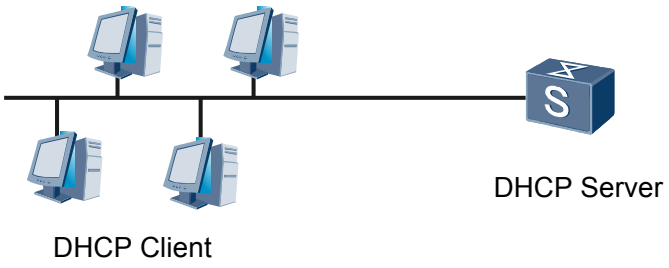
DHCP rapidly and dynamically allocates IP addresses, which improves IP address usage.

4.2 DHCP Features Supported by the switch

The device can function as the DHCP relay agent or DHCP server.

Using the switch as a DHCP Server

Figure 4-1 Networking of the DHCP server



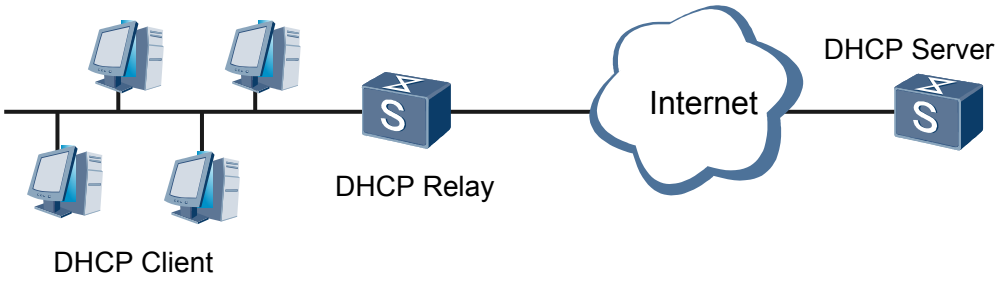
The device can function as the DHCP server to assign IP addresses to clients. After a DHCP client sends a message to the DHCP server to request configuration parameters, the DHCP server responds with a message carrying the requested configurations based on a policy.

When the device functions as the DHCP server, create an address pool on the device to provide IP addresses to DHCP clients. The address pool can be a global address pool or an interface address pool. The device allocates IP addresses to clients by using the global address pool or an interface address pool:

- If a DHCP server based on a global address pool is configured, all online users of the server can obtain IP addresses from this address pool. The global address pool is used when the DHCP server and client are located on different network segments.
- If a DHCP server based on an interface address pool is configured, only users that go online from this interface can obtain IP addresses from this address pool. The interface address pool is used when the DHCP server and client are located on the same network segment.

Using the switch as a DHCP Relay Agent

Figure 4-2 Networking of the DHCP relay agent



The switch supports the DHCP relay function. When the device functions as a DHCP relay agent, the client can communicate with a DHCP server on another network segment through the device, and obtain an IP address and other configuration parameters from the global address pool on the DHCP server. In this manner, DHCP clients on multiple network segments can share one DHCP server. This reduces costs and facilitates centralized management.

NOTE

When the default CPCAR is used,

- The DHCP server can respond to the concurrent login requests of 85 DHCP clients.

4.3 Default Configuration

This section provides default DHCP configurations.

Table 4-1 DHCP default configuration

Parameter	Default Value
Time interval at which the DHCP server waits for the response to ping packets to avoid IP address conflicts	500 ms
IP address lease	1 day
Interval for saving DHCP data to the CF card	7200s
NetBIOS node type of the DHCP client	unspecified

4.4 Configuring a DHCP Server Based on the Global Address Pool

If a DHCP server based on a global address pool is configured, all online users of the server can obtain IP addresses from this address pool.

Pre-configuration Tasks

Before configuring a DHCP server based on the global address pool, complete the following tasks:

- Ensuring that the link between the DHCP client and the device works properly and the DHCP client can communicate with the device
- (Optional) Configuring the DNS service for the DHCP client
- (Optional) Configuring the NetBIOS service for the DHCP client
- Configuring routes from the device to the DNS server and the NetBIOS server (The routes are required only when the servers are configured.)
- (Optional) Configuring the customized DHCP option

4.4.1 Configuring the Global Address Pool

Context

The global address pool attributes include the IP address range, IP address lease, IP addresses not to be automatically allocated, and IP addresses to be statically bound to MAC addresses. IP addresses in the global address pool can be assigned dynamically or bound manually as required.

A maximum of 256 address pools, including global address pools and interface address pools, can be created on the switch.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip pool ip-pool-name
```

A global address pool is created and the global address pool view is displayed.

By default, no global address pool exists on the switch.

Step 3 Run:

```
network ip-address [ mask { mask | mask-length } ]
```

The range of IP addresses that can be allocated dynamically in the global address pool is specified.

By default, no network segment address for a global address pool is specified.

An address pool can contain only one address segment. The address range of the address pool is set by the mask.

NOTE

When configuring the range of dynamically assignable IP addresses in the global address pool, ensure that the range is the same as the network segment on which the DHCP server interface address or the DHCP relay agent interface address resides. This avoids incorrect assignment of IP addresses.

Step 4 (Optional) Run:

```
lease { day day [ hour hour [ minute minute ] ] | unlimited }
```

The IP address lease is set.

By default, the IP address lease is one day.

Different address pools on a DHCP server can be set with different IP address leases, but the IP addresses in one address pool must be configured with the same lease.

Step 5 (Optional) Run:

```
excluded-ip-address start-ip-address [ end-ip-address ]
```

The IP addresses that cannot be automatically allocated in the global address pool are configured.

By default, all IP addresses in the address pool can be automatically assigned to clients.

Some IP addresses in the global address pool are reserved for other services, for example, the IP address of the DNS server cannot be allocated to clients. If you run this command multiple times, you can set multiple IP address ranges that cannot be automatically allocated in the DHCP address pool.

Step 6 Run:

```
gateway-list ip-address &<1-8>
```

The egress gateway address is configured for the DHCP clients.

NOTE

When a DHCP client connects to the DHCP server or host outside the network segment, data must be forwarded through the egress gateway.

To load balance traffic and improve network reliability, configure multiple gateways. An address pool can be configured with a maximum of eight gateway addresses. Gateway addresses cannot be subnet broadcast addresses.

Step 7 (Optional) Run:

```
static-bind ip-address ip-address mac-address mac-address
```

An IP address in the global address pool is statically bound to the MAC address of a DHCP client.

By default, the IP address in a global address pool is not bound to any MAC address.

When a client requires a fixed IP address, bind an idle IP address in the address pool to the client MAC address.

NOTE

When the IP address in the global address pool is statically bound to a MAC address, the IP address must be in the range of IP addresses that can be allocated dynamically.

Step 8 (Optional) Run:

```
force insert option code &<1-254>
```

A DHCP server is configured to forcibly insert an Option field specified in the global address pool to a DHCP Response packet that it sends to a DHCP client.

By default, a DHCP server is not configured to forcibly insert an Option field to a DHCP Response packet that it sends to a DHCP client.

Step 9 (Optional) Run the following commands to configure the DHCP client to automatically obtain the startup configuration file.

1. Run:
   bootfile *bootfile*

   The name of the startup configuration file is configured for the DHCP client.

   By default, the startup configuration file name is not configured for the DHCP client.
2. Run:
   sname *sname*

   The name of the server where the DHCP client obtains the startup configuration file is configured.

   By default, the name of the server where the DHCP client obtains the startup configuration file is not configured.

NOTE

Besides assigning IP addresses, a DHCP server can also provide the required network configuration parameters, such as the startup configuration file name for the DHCP clients. Usually, the startup configuration file is saved on a specified file server. Therefore, the route between the DHCP client and the file server must be reachable.

Step 10 (Optional) Run:
next-server *ip-address*

The server IP address for DHCP clients is configured.

By default, no server IP address is specified.

Step 11 (Optional) Run:
vpn-instance *vpn-instance-name*

The IP address pool is binded to a VPN instance.

By default, an IP address pool is not bound to any VPN instance.

Step 12 (Optional) Run:
lock

The IP address pool is locked.

By default, the IP address pool is unlocked.

----End

4.4.2 Configuring an Interface to Use the Global Address Pool

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
dhcp enable

DHCP is enabled.

Step 3 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces on the switch can be configured to select the global address pool for IP address allocation.

Step 4 Run:

```
ip address ip-address { mask | mask-length }
```

An IP address is assigned to the interface.

When users connected to the interface that has an IP address configured request IP addresses:

- If the switch used as the DHCP server is on the same network segment as DHCP clients, and no relay agent is deployed between them, the switch assigns IP addresses on the same network segment as the interface to users who get online from the interface. If the interface is not configured with an IP address or no address pool is on the same network segment as the interface address, the clients cannot go online.
- If the switch used as the DHCP server and DHCP clients are on different network segments, and a DHCP relay agent is deployed between them, the switch parses the giaddr field of a DHCP Request message to obtain an IP address. If the IP address does not match the corresponding address pool, the user cannot get online.

Step 5 Run:

```
dhcp select global
```

The interface is configured to use the global address pool.

After the configuration is complete, users who get online from this interface can obtain IP addresses and other configuration parameters from the global address pool.

NOTE

If there is a DHCP relay agent between the DHCP client and server, this step is optional. Otherwise, this step is mandatory.

----End

4.4.3 (Optional) Configuring the Static DNS Service on a DHCP Client

Context

When a host connects to the Internet through the domain name, the domain name needs to be resolved to the IP address. This is implemented by the DNS. To ensure that a DHCP client can successfully connect to the Internet, the DHCP server needs to specify the DNS server address when allocating the IP address to the client.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip pool ip-pool-name
```

The IP address pool view is displayed.

Step 3 Run:

```
domain-name domain-name
```

The domain name to be allocated to a DHCP client is configured.

On the DHCP server, you can specify a domain name for each address pool.

Step 4 Run:

```
dns-list ip-address &<1-8>
```

The IP address of the DNS server is configured for a DHCP client.

To load balance the traffic and improve network reliability, configure multiple DNS servers. Each address pool can be configured with a maximum of eight DNS server IP addresses.

----End

4.4.4 (Optional) Configuring the Static NetBIOS Service on a DHCP Client

Context

When a DHCP client uses the Network Basic Input Output System (NetBIOS) protocol for communication, the host names must be mapped to IP addresses. Based on the modes of obtaining mapping, NetBIOS nodes are classified into the following types:

- b-node: indicates a node in broadcast mode. This node obtains mappings in broadcast mode.
- p-node: indicates a node in peer-to-peer mode. This node obtains mappings by communicating with the NetBIOS server.
- m-node: indicates a node in mixed mode. An m-node has some broadcast features.
- h-node: indicates a node in hybrid mode. An h-node is a b-type node enabled with the end-to-end communication mechanism.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip pool ip-pool-name
```

The IP address pool view is displayed.

Step 3 Run:

```
nbns-list ip-address &<1-8>
```

The IP address of the NetBIOS server is configured for a DHCP client.

Each IP address pool can be configured with a maximum of eight NetBIOS server IP addresses.

Step 4 Run:

```
netbios-type { b-node | h-node | m-node | p-node }
```

The NetBIOS node type is configured for a DHCP client.

By default, no NetBIOS node type is specified for a DHCP client.

----End

4.4.5 (Optional) Configuring a Customized DHCP Option for the Global Address Pool

Context

DHCP provides various options. To use these options, add them to the attribute list of the DHCP server manually. If the DHCP server is configured with the Options field, the DHCP client obtains the configuration of the Options field from the DHCP packet replied by the DHCP server when the client requests an IP address from the server.

NOTE

The **option** command configures basic functions, such as the DNS service, NetBIOS service, and IP address lease. The system also provides commands to configure these functions separately. The commands used to configure these functions separately take precedence over the **option** command.

The related commands are as follows:

- DNS service: **domain-name** and **dns-list**
- NetBIOS service: **nbns-list** and **netbios-type**
- Lease: **lease**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip pool ip-pool-name
```

The IP address pool view is displayed.

Step 3 Run:

```
option code [ sub-option sub-code ] { ascii ascii-string | hex hex-string | cipher cipher-string | ip-address ip-address &<1-8> }
```

The customized DHCP option is configured.

After the **option** command is used, the specified option is carried by the DHCP Reply message returned by the DHCP server. Before using this command, ensure that you know the functions of the option to be configured. For details on DHCP options, see RFC 2132.

----End

4.4.6 (Optional) Preventing Repeated IP Address Allocation

Context

Before assigning an address to a client, the switch used as the DHCP server needs to ping the IP address to avoid address conflicts.

After the **dhcp server ping** command is executed, the DHCP server can prevent repeated IP address allocation. The DHCP server pings an IP address to be allocated. If there is no response to the ping packet within a certain period, the DHCP server continues to send ping packets to this IP address until the number of ping packets reaches the maximum value. If there is still no response, this IP address is not in use, and the DHCP server allocates the IP address to a client.

Duplicate IP address detection on the DHCP server should not be too long. Otherwise, the client cannot obtain an IP address. It is recommended that the configured total detection time (Maximum number of send ping packets x Maximum response time) be smaller than 8s.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server ping packet number
```

The maximum number of ping packets to be sent by the switch is set.

By default, the DHCP server sends 0 ping packets, indicating that no ping operation is performed.

Step 3 Run:

```
dhcp server ping timeout milliseconds
```

The period in which the switch waits for the response is set.

By default, the period in which the switch waits for the response is 500 ms.

----End

4.4.7 (Optional) Configuring Automatic Saving of DHCP Data

Context

When the device functions as the DHCP server, you can enable automatic saving of DHCP data so that IP address information is saved to the storage device periodically.

You can configure the device to save DHCP data to the storage device. When a fault occurs, you can restore data from the storage device.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server database enable
```

The function that saves DHCP data to the CF card is enabled.

By default, DHCP data is not saved to the CF card.

After this command is executed, the system generates the **lease.txt** and **conflict.txt** files in the CF card. The two files save the address lease information and address conflict information.

Step 3 Run:

```
dhcp server database write-delay interval
```

The interval for saving DHCP data is set.

After the device is configured to automatically save DHCP data, the device saves data every 7200 seconds by default and the latest data overwrites the previous data.

Step 4 Run:

```
dhcp server database recover
```

The DHCP data in the storage device is restored.

After this command is executed, the device restores DHCP data from the CF card during a restart.

----End

4.4.8 (Optional) Configuring the DHCP Server to trust Option 82

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server trust option82
```

The switch is configured to trust Option 82.

By default, Option 82 is enabled on the DHCP server.

----End

4.4.9 (Optional) Configuring the DHCP Server to Allocate IP Addresses to BOOTP Clients

Context

When the device functions as a DHCP server, the device can allocate IP addresses to BOOTP clients if the BOOTP clients reside on the same network as the DHCP server.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server bootp
```

The DHCP server is configured to respond to BOOTP requests.

By default, a DHCP server does not respond to BOOTP requests.

Step 3 Run:

```
dhcp server bootp automatic
```

The DHCP server is configured to allocate IP addresses to BOOTP clients.

By default, a DHCP server does not allocate IP addresses to BOOTP clients.

----End

4.4.10 Checking the Configuration

Procedure

- Run the **display ip pool** [**name** *ip-pool-name* [*start-ip-address* [*end-ip-address*] | **all** | **conflict** | **expired** | **used**]] command to check information about the specified global address pool.
- Run the **display dhcp server database** command to check information about the DHCP database.

----End

4.5 Configuring a DHCP Server Based on an Interface Address Pool

After a DHCP server based on an interface address pool is configured, only users that go online from this interface can obtain IP addresses from this address pool.

Pre-configuration Tasks

Before configuring a DHCP server based on an interface address pool, complete the following tasks:

- Ensuring that the link between the DHCP client and the device works properly and the DHCP client can communicate with the device
- (Optional) Configuring the DNS server
- (Optional) Configuring the NetBIOS server
- Configuring routes from the device to the DNS server and the NetBIOS server (The routes are required only when the servers are configured.)

4.5.1 Configuring an Interface Address Pool

Context

The interface address pool attributes include the IP address lease, IP addresses not to be automatically allocated, and IP addresses to be statically bound to MAC addresses. IP addresses in the interface address pool can be assigned dynamically or bound manually as required.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled.

Step 3 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces can be configured to select interface address pools for IP address allocation.

Step 4 Run:

```
ip address ip-address { mask | mask-length }
```

An IP address is assigned to the interface.

Step 5 Run:

```
dhcp select interface
```

The interface is configured to use the interface address pool.

The interface address pool is actually the network segment to which the interface belongs, and such an interface address pool only applies to this interface.

Step 6 (Optional) Run:

```
dhcp server lease { day day [ hour hour [ minute minute ] ] | unlimited }
```

The IP address lease is set.

By default, the IP address lease is 1 day.

Step 7 (Optional) Run:

```
dhcp server excluded-ip-address start-ip-address [ end-ip-address ]
```

The IP addresses that cannot be automatically allocated in the interface address pool are configured.

Some IP addresses in the interface address pool are reserved for other services, for example, the IP address of the DNS server cannot be allocated to clients. If you run this command multiple times, you can set multiple IP address ranges that cannot be automatically allocated in the DHCP address pool.

Step 8 (Optional) Run:

```
dhcp server static-bind ip-address ip-address mac-address mac-address
```

An IP address in the interface address pool is statically bound to the MAC address of a DHCP client.

When a client requires a fixed IP address, bind an idle IP address in the address pool to the client MAC address.

NOTE

When the IP address in the global address pool is statically bound to a MAC address, the IP address must be in the range of IP addresses that can be allocated dynamically.

Step 9 (Optional) Run:

```
dhcp server force insert option code &<1-254>
```

A DHCP server is configured to forcibly insert the specified Option field to a DHCP Response packet that it sends to a DHCP client.

By default, no DHCP server forcibly inserts the specified Option field to a DHCP Response packet that it sends to a DHCP client.

Step 10 (Optional) Run the following commands to configure the DHCP client to automatically obtain the startup configuration file.

1. Run:

   ```
   dhcp server bootfile bootfile
   ```

   The name of the startup configuration file is configured for the DHCP client.

   By default, the startup configuration file name is not configured for the DHCP client.

2. Run:

   ```
   dhcp server sname sname
   ```

   The name of the server where the DHCP client obtains the startup configuration file is configured.

   By default, the name of the server where the DHCP client obtains the startup configuration file is not configured.

NOTE

Besides assigning IP addresses, a DHCP server can also provide the required network configuration parameters, such as the startup configuration file name for the DHCP clients. Usually, the startup configuration file is saved on a specified file server. Therefore, the route between the DHCP client and the file server must be reachable.

----End

4.5.2 (Optional) Configuring the Static DNS Service on a DHCP Client

Context

When a host connects to the Internet through the domain name, the domain name needs to be resolved to the IP address. This is implemented by the DNS. To ensure that a DHCP client can successfully connect to the Internet, the DHCP server needs to specify the DNS server address when allocating the IP address to the client.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces can be configured to select interface address pools for IP address allocation.

Step 3 Run:

```
dhcp server domain-name domain-name
```

The domain name to be allocated to a DHCP client is configured.

Step 4 Run:

```
dhcp server dns-list ip-address &<1-8>
```

The IP address of the DNS server is configured for a DHCP client.

To load balance the traffic and improve network reliability, configure multiple DNS servers. Each address pool can be configured with a maximum of eight DNS server IP addresses.

----End

4.5.3 (Optional) Configuring the Static NetBIOS Service on a DHCP Client

Context

When a DHCP client uses the NetBIOS protocol for communication, the host names must be mapped to IP addresses. Based on the modes of obtaining mapping, NetBIOS nodes are classified into the following types:

- b-node: indicates a node in broadcast mode. This node obtains mappings in broadcast mode.
- p-node: indicates a node in peer-to-peer mode. This node obtains mappings by communicating with the NetBIOS server.
- m-node: indicates a node in mixed mode. An m-node is a p-type node with some broadcast features.
- h-node: indicates a node in hybrid mode. An h-node is a b-type node enabled with the end-to-end communication mechanism.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces can be configured to select interface address pools for IP address allocation.

Step 3 Run:

```
dhcp server nbns-list ip-address &<1-8>
```

The IP address of the NetBIOS server is configured for a DHCP client.

Each IP address pool can be configured with a maximum of eight NetBIOS server IP addresses.

Step 4 Run:

```
dhcp server netbios-type { b-node | h-node | m-node | p-node }
```

The NetBIOS node type is configured for a DHCP client.

By default, no NetBIOS node type is specified for a DHCP client.

----End

4.5.4 (Optional) Configuring a Customized DHCP Option for an Interface Address Pool

Context

DHCP provides various options. To use these options, add them to the attribute list of the DHCP server manually.

When a DHCP client requests an IP address from the DHCP server configured with the Options field, the server returns a DHCP Reply message containing the Options field.

NOTE

The **dhcp server option** command configures basic functions, such as the DNS service, NetBIOS service, and IP address lease. The system also provides commands to configure these functions separately. The commands used to configure these functions separately take precedence over the **dhcp server option** command.

The related commands are as follows:

- DNS service: **dhcp server domain-name** and **dhcp server dns-list**
- NetBIOS service: **dhcp server nbns-list** and **dhcp server netbios-type**
- Lease: **dhcp server lease**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces can be configured to select interface address pools for IP address allocation.

Step 3 Run:

```
dhcp server option code [ sub-option sub-code ] { ascii ascii-string | hex hex-string | cipher cipher-string | ip-address ip-address &<1-8> }
```

The customized DHCP option is configured.

After the **dhcp server option** command is run, the specified option is carried by the DHCP Reply message returned by the DHCP server. Before using this command, ensure that you know the functions of the option to be configured. For details on DHCP options, see RFC 2132.

----End

4.5.5 (Optional) Preventing Repeated IP Address Allocation

Context

Before assigning an address to a client, the switch used as the DHCP server needs to ping the IP address to avoid address conflicts.

After the **dhcp server ping** command is executed, the DHCP server can prevent repeated IP address allocation. The DHCP server pings an IP address to be allocated. If there is no response to the ping packet within a certain period, the DHCP server continues to send ping packets to this IP address until the number of ping packets reaches the maximum value. If there is still no response, this IP address is not in use, and the DHCP server allocates the IP address to a client.

Duplicate IP address detection on the DHCP server should not be too long. Otherwise, the client cannot obtain an IP address. It is recommended that the configured total detection time (Maximum number of send ping packets x Maximum response time) be smaller than 8s.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server ping packet number
```

The maximum number of ping packets to be sent by the switch is set.

By default, the DHCP server sends 0 ping packets, indicating that no ping operation is performed.

Step 3 Run:

```
dhcp server ping timeout milliseconds
```

The period in which the switch waits for the response is set.

By default, the period in which the switch waits for the response is 500 ms.

----End

4.5.6 (Optional) Configuring Automatic Saving of DHCP Data

Context

When the device functions as the DHCP server, you can enable automatic saving of DHCP data so that IP address information is saved to the storage device periodically.

You can configure the device to save DHCP data to the storage device. When a fault occurs, you can restore data from the storage device.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server database enable
```

The function that saves DHCP data to the CF card is enabled.

By default, DHCP data is not saved to the CF card.

After this command is executed, the system generates the **lease.txt** and **conflict.txt** files in the CF card. The two files save the address lease information and address conflict information.

Step 3 Run:

```
dhcp server database write-delay interval
```

The interval for saving DHCP data is set.

After the device is configured to automatically save DHCP data, the device saves data every 7200 seconds by default and the latest data overwrites the previous data.

Step 4 Run:

```
dhcp server database recover
```

The DHCP data in the storage device is restored.

After this command is executed, the device restores DHCP data from the CF card during a restart.

----End

4.5.7 (Optional) Configuring the DHCP Server to trust Option 82

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server trust option82
```

The switch is configured to trust Option 82.

By default, Option 82 is enabled on the DHCP server.

----End

4.5.8 (Optional) Configuring the DHCP Server to Allocate IP Addresses to BOOTP Clients

Context

When the device functions as a DHCP server, the device can allocate IP addresses to BOOTP clients if the BOOTP clients reside on the same network as the DHCP server.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server bootp
```

The DHCP server is configured to respond to BOOTP requests.

By default, a DHCP server does not respond to BOOTP requests.

Step 3 Run:

```
dhcp server bootp automatic
```

The DHCP server is configured to allocate IP addresses to BOOTP clients.

By default, a DHCP server does not allocate IP addresses to BOOTP clients.

----End

4.5.9 Checking the Configuration

Procedure

- Run the **display ip pool** [**interface** *interface-pool-name* [*start-ip-address* [*end-ip-address*] | **all** | **conflict** | **expired** | **used**]] command to view information about the IP address pool.

----End

4.6 Configuring a DHCP Relay Agent

By using a DHCP relay agent, a DHCP client can communicate with a DHCP server on another network segment to obtain an IP address and other configuration information.

Pre-configuration Tasks

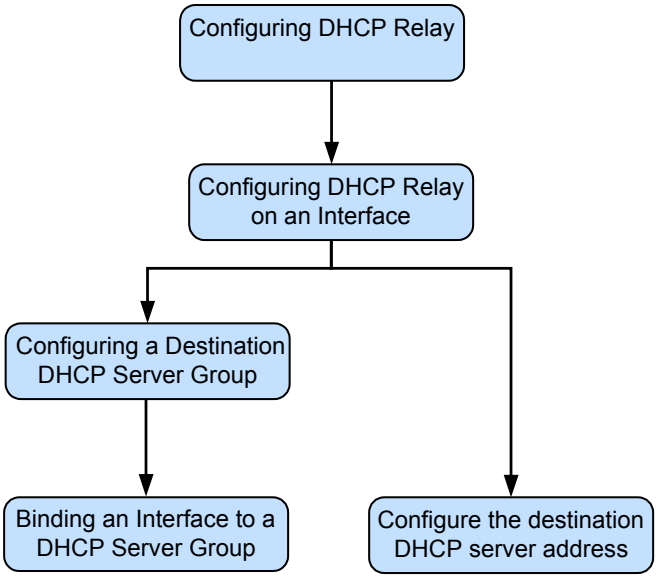
Before configuring a DHCP relay agent, complete the following tasks:

- Configuring a DHCP server
- Configuring a route from the device used as the DHCP relay agent to the DHCP server

Configuration Process

Figure 4-3 shows the configuration process.

Figure 4-3 DHCP relay agent configuration process



4.6.1 Configuring DHCP Relay on an Interface

Context

When the network where a DHCP client resides does not have a DHCP server, a DHCP relay agent can be configured to forward DHCP messages of the client to a DHCP server.

NOTE

A DHCP message is forwarded between a DHCP client and a DHCP server at most 16 times, and then the DHCP message is discarded.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled.

Step 3 (Optional) Run:

```
ip relay address cycle
```

The DHCP server polling function on a DHCP relay agent is enabled.

By default, the DHCP server polling function is disabled on the DHCP relay agent.

Step 4 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Layer 3 GE sub-interfaces, XGE sub-interfaces, 40GE sub-interfaces, Layer 3 Ethernet sub-interfaces, Layer 3 Eth-Trunk sub-interfaces, VLANIF interfaces on the device support the DHCP relay function.

Step 5 Run:

```
ip address ip-address { mask | mask-length }
```

An IP address is assigned to the interface.

NOTE

The interface IP address must be the same as the DHCP client's egress gateway address configured on the DHCP server. If the device functions as the DHCP server, you can run the **gateway-list (IP address pool view)** command to configure the DHCP client's egress gateway address.

Step 6 Run:

```
dhcp select relay
```

The DHCP relay function is enabled on the interface.

NOTE

When the DHCP relay function is enabled on the sub-interface, run **arp broadcast enable** to enable the ARP broadcast function on a sub-interface.

Step 7 Run:

```
quit
```

Return to the system view.

Step 8 (Optional) Run:

```
dhcp relay trust option82
```

The device is configured to trust Option 82.

By default, the device does not discard DHCP messages with Option 82 and giaddr field of the packet is 0.

----End

Follow-up Procedure

When the DHCP relay function is enabled on an interface, specify the DHCP server IP address on the interface in either of the following ways:

- Configure a destination DHCP server group and bind the group to the interface. For details, see **4.6.2 Configuring a Destination DHCP Server Group** and **4.6.3 Binding an Interface to a DHCP Server Group**.

- Run the **dhcp relay server-ip** *ip-address* command in the interface view to configure the destination DHCP server address.

4.6.2 Configuring a Destination DHCP Server Group

Context

After a DHCP server group is created and server IP addresses are added to the group, the switch used as the DHCP relay agent can forward messages to multiple servers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp server group group-name
```

A DHCP server group is created and the DHCP server group view is displayed.

A maximum of 64 DHCP server groups can be configured globally.

Step 3 Run:

```
dhcp-server ip-address [ ip-address-index ]
```

A DHCP server is added to a DHCP server group.

A maximum of 20 DHCP servers can be added to a DHCP server group.

Step 4 (Optional) Run:

```
vpn-instance vpn-instance-name
```

The DHCP server group is bound to the created VPN instance.

----End

4.6.3 Binding an Interface to a DHCP Server Group

Context

After the DHCP relay function is enabled on an interface, bind a DHCP server group to the interface so that DHCP clients can access DHCP servers in the bound server group.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Layer 3 GE sub-interfaces, XGE sub-interfaces, 40GE sub-interfaces, Layer 3 Ethernet sub-interfaces, Layer 3 Eth-Trunk sub-interfaces, VLANIF interfaces on the device support the DHCP relay function.

Step 3 Run:

```
dhcp relay server-select group-name
```

A DHCP server group is bound to the interface.

You can also run the **dhcp relay server-ip** *ip-address* command to specify the DHCP server IP address on a interface.

Step 4 (Optional) Run:

```
ip binding vpn-instance vpn-instance-name
```

The VLANIF interface is bound to a VPN instance.

If a user connected to the switch interface is on a private network, bind the interface to a VPN instance. The bound VPN instance must be the same as the VPN instance bound to the DHCP server group. For details on how to bind a VPN instance to a DHCP server group, see **4.6.2 Configuring a Destination DHCP Server Group**.

----End

4.6.4 (Optional) Configuring the DHCP Relay Agent to Send DHCP Release Messages

Context

If a user is forcibly disconnected, you can manually release the IP address assigned to the user on the DHCP server. You can configure the DHCP relay agent to actively send DHCP Release messages to the DHCP server. The DHCP server then releases the specified IP addresses.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 (Optional) Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Layer 3 GE sub-interfaces, XGE sub-interfaces, 40GE sub-interfaces, Layer 3 Ethernet sub-interfaces, Layer 3 Eth-Trunk sub-interfaces, VLANIF interfaces on the switch support the DHCP relay function.

Step 3 Run:

```
dhcp relay release client-ip-address mac-address [ vpn-instance vpn-instance-name ]
[ server-ip-address ]
```

The DHCP relay agent is configured to send DHCP Release messages to the DHCP server.

NOTE

vpn-instance *vpn-instance-name* is not available in VLANIF interface view.

- When you use the **dhcp relay release** command in the system view:
  - If no DHCP server is specified, the DHCP relay agent will send DHCP Release messages to the servers in all DHCP server groups bound to the DHCP relay interfaces.
  - If a DHCP server is specified, the DHCP relay agent sends DHCP Release messages to only the specified DHCP server.
- When you use the **dhcp relay release** command in the VLANIF interface view:
  - If no DHCP server is specified, the DHCP relay agent will send DHCP Release messages to all the servers in the DHCP server group bound to this VLANIF interface.
  - If a DHCP server is specified, the DHCP relay agent sends DHCP Release messages to only the specified DHCP server.

----End

4.6.5 (Optional) Configuring Strategies for Processing Option 82 Information on the DHCP Relay Agenet

Context

When DHCP Request messages carry Option 82 information, the DHCP server can locate user positions accurately and assign IP addresses to users using different policies. You can configure strategies that the DHCP relay agent uses to process Option 82 information.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

The device supports VLANIF interfaces to work in DHCP relay mode.

Step 3 Run:

```
dhcp relay information enable
```

The Option 82 function is enabled on the DHCP relay agent.

By default, the Option 82 function is disabled for the DHCP relay agent.

Step 4 Run:

```
dhcp relay information strategy { drop | keep | replace }
```

Strategies used by the DHCP relay agent to process Option 82 information are configured.

By default, the strategy used by the DHCP relay agent to process Option 82 information is **replace**.

----End

4.6.6 Checking the Configuration

Procedure

- Run the **display dhcp relay** { **all** | **interface** *interface-type interface-number* } command to view the DHCP server group or the DHCP servers on the DHCP relay interface.
- Run the **display dhcp relay statistics** command to view packet statistics on the DHCP relay agent.
- Run the **display dhcp server group** [*group-name*] command to view the DHCP server group configuration.

----End

4.7 Maintaining DHCP

After DHCP configurations are complete, you can clear DHCP statistics and monitor DHCP operation.

4.7.1 Clearing DHCP Statistics

Context

During routine maintenance, you can use the reset commands to clear DHCP statistics.

CAUTION

DHCP statistics cannot be restored after they are cleared. Exercise caution when running the reset commands.

Procedure

- Run the **reset dhcp server statistics** command in the user view to clear DHCP server statistics.
- Run the **reset dhcp statistics** command in the user view to clear the DHCP message statistics.
- Run the **reset dhcp relay statistics** [**server-group** *group-name*] command in the user view to clear DHCP relay agent statistics.

----End

4.7.2 Clearing the DHCP Address Pool

Procedure

- Run the **reset ip pool** { **interface** *pool-name* | **name** *ip-pool-name* } { *start-ip-address* [*end-ip-address*] | **all** | **conflict** | **expired** | **used** } command to reset the configured IP address pool on the device.

----End

4.7.3 Monitoring DHCP Operation

Context

DHCP packet statistics contain only the number of packets received and sent by the DHCP module.

Procedure

- Run the **display dhcp statistics** command to view DHCP message statistics.
- Run the **display dhcp relay statistics** command to view statistics on the DHCP Relay Agent.
- Run the **display dhcp server statistics** command to view statistics on the DHCP Server.
- Run the **display dhcp relay** { **all** | **interface** *interface-type interface-number* } command to view the DHCP server group or the DHCP server on a VLANIF interface.

----End

4.8 Configuration Examples

This section provides DHCP configuration examples including networking requirements and configuration roadmap.

4.8.1 Example for Configuring a DHCP Server Based on the Global Address Pool

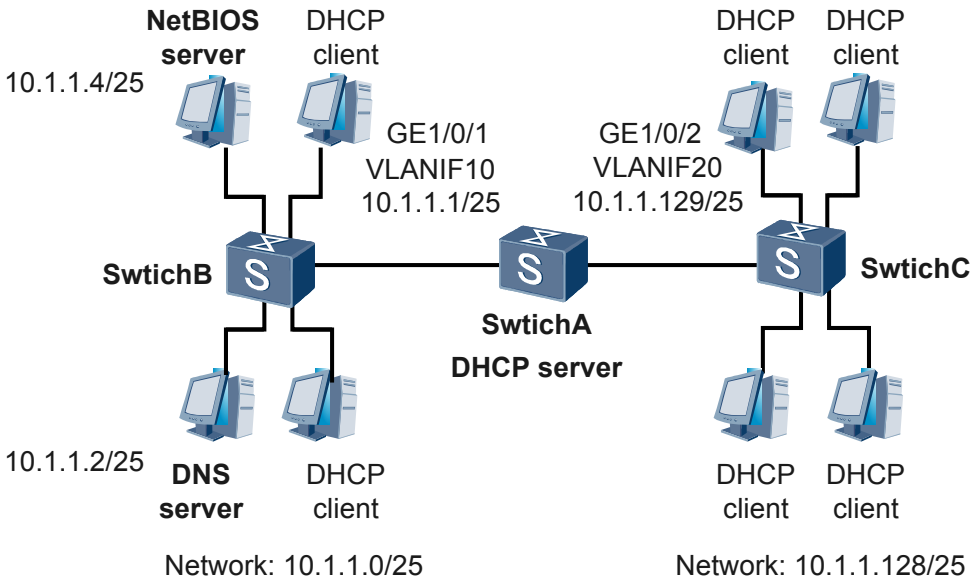
Networking Requirements

As shown in **Figure 4-4**, an enterprise has two offices on the same network segment. To reduce network construction cost, the enterprise uses one DHCP server to assign IP addresses for hosts in the two offices.

All the hosts in Office1 are on the network segment 10.1.1.0/25 and added to VLAN 10. Hosts in Office1 only use the DNS service with a lease of ten days. All the hosts in Office2 are on the network segment 10.1.1.128/25 and added to VLAN 20. Hosts in Office2 use the DNS service and NetBIOS service with a lease of two days.

You can configure a global address pool on SwitchA and enable the server to dynamically assign IP addresses to hosts in the two offices.

Figure 4-4 Networking diagram for configuring a DHCP server based on the global address pool



Configuration Roadmap

The configuration roadmap is as follows:

1. Create two global address pools on the SwitchA and set attributes of the pools. Assign IP addresses to Office1 and Office2 as required.
2. Configure VLANIF interfaces to use the global address pool to assign IP addresses to clients.

Procedure

Step 1 Enable DHCP

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] dhcp enable
```

Step 2 Create address pools and set the attributes of the address pools

Set the attributes of IP address pool 1, including the address pool range, DNS server address, gateway address, and address lease.

```
[SwitchA] ip pool 1
[SwitchA-ip-pool-1] network 10.1.1.0 mask 255.255.255.128
[SwitchA-ip-pool-1] dns-list 10.1.1.2
[SwitchA-ip-pool-1] gateway-list 10.1.1.1
[SwitchA-ip-pool-1] excluded-ip-address 10.1.1.2
[SwitchA-ip-pool-1] excluded-ip-address 10.1.1.4
[SwitchA-ip-pool-1] lease day 10
[SwitchA-ip-pool-1] quit
```

Set the attributes of IP address pool 2, including the address pool range, DNS server address, egress gateway address, NetBIOS server address, and address lease

```
[SwitchA] ip pool 2
[SwitchA-ip-pool-2] network 10.1.1.128 mask 255.255.255.128
[SwitchA-ip-pool-2] dns-list 10.1.1.2
[SwitchA-ip-pool-2] nbns-list 10.1.1.4
[SwitchA-ip-pool-2] gateway-list 10.1.1.129
[SwitchA-ip-pool-2] lease day 2
[SwitchA-ip-pool-2] quit
```

Step 3 Set the address assignment mode on the VLANIF interfaces

Add GigabitEthernet1/0/1 and GigabitEthernet1/0/2 to the corresponding VLANs.

```
[SwitchA] vlan batch 10 20
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 10
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 10
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface gigabitethernet 1/0/2
[SwitchA-GigabitEthernet1/0/2] port hybrid pvid vlan 20
[SwitchA-GigabitEthernet1/0/2] port hybrid untagged vlan 20
[SwitchA-GigabitEthernet1/0/2] quit
```

Configure clients on VLANIF 10 to obtain IP addresses from the global address pool.

```
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] ip address 10.1.1.1 255.255.255.128
[SwitchA-Vlanif10] dhcp select global
[SwitchA-Vlanif10] quit
```

Configure clients on VLANIF 20 to obtain IP addresses from the global address pool.

```
[SwitchA] interface vlanif 20
[SwitchA-Vlanif20] ip address 10.1.1.129 255.255.255.128
[SwitchA-Vlanif20] dhcp select global
[SwitchA-Vlanif20] quit
```

Step 4 Verify the configuration

Run the **display ip pool** command on the SwitchA to view the IP address pool configuration.

```
[SwitchA] display ip pool
  ------------------------------------------------------------------------
  Pool-name      : 1
  Pool-No        : 0
  Position       : Local           Status           : Unlocked
  Gateway-0      : 10.1.1.1
  Mask           : 255.255.255.128
  VPN instance   : --

  ------------------------------------------------------------------------
  Pool-name      : 2
  Pool-No        : 1
  Position       : Local           Status           : Unlocked
  Gateway-0      : 10.1.1.129
  Mask           : 255.255.255.128
  VPN instance   : --


   IP address Statistic
    Total       :250
    Used        :6          Idle        :242
    Expired     :0          Conflict    :0          Disable   :2
```

----End

Configuration Files

Configuration file of SwitchA

```
#
 sysname SwitchA
#
 vlan batch 10 20
#
dhcp enable
#
ip pool 1
 gateway-list 10.1.1.1
 network 10.1.1.0 mask 255.255.255.128
 excluded-ip-address 10.1.1.2
 excluded-ip-address 10.1.1.4
 lease day 10 hour 0 minute 0
 dns-list 10.1.1.2
#
ip pool 2
 gateway-list 10.1.1.129
 network 10.1.1.128 mask 255.255.255.128
 lease day 2 hour 0 minute 0
 dns-list 10.1.1.2
 nbns-list 10.1.1.4
#
interface Vlanif10
 ip address 10.1.1.1 255.255.255.128
 dhcp select global
#
interface Vlanif20
 ip address 10.1.1.129 255.255.255.128
 dhcp select global
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 10
 port hybrid untagged vlan 10
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 20
 port hybrid untagged vlan 20
#
return
```

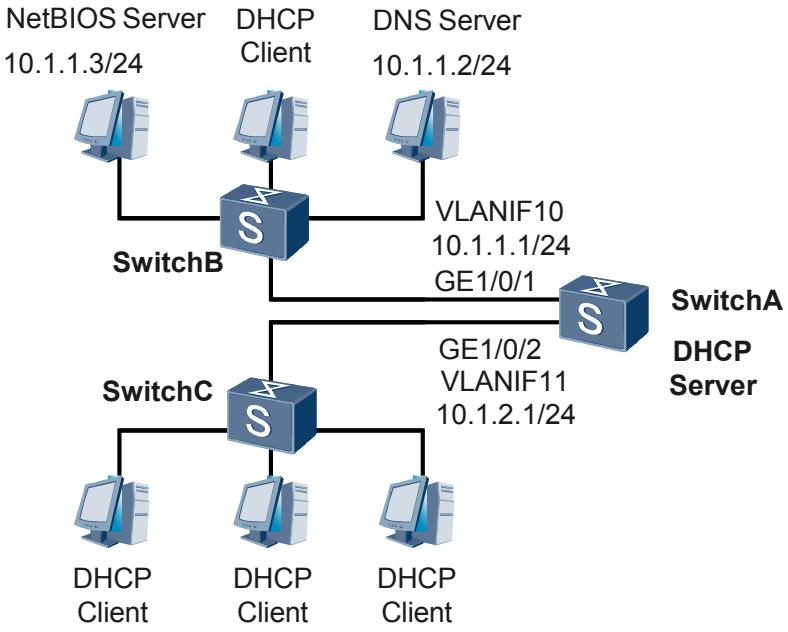
4.8.2 Example for Configuring a DHCP Server Based on the Interface Address Pool

Networking Requirements

As shown in **Figure 4-5**, an enterprise has two offices on the same network segment. To reduce network construction cost, the enterprise uses one DHCP server to assign IP addresses for hosts in the two offices.

All the hosts in Office1 are on the network segment 10.1.1.0/24 and added to VLAN 10. Hosts in Office1 use the DNS service and NetBIOS service with a lease of thirty days. All the hosts in Office2 are on the network segment 10.1.2.0/24 and added to VLAN 11. Hosts in Office2 do not use the DNS service or NetBIOS service. The lease of the IP address is tweenty days.

Figure 4-5 Networking diagram for configuring a DHCP server based on the VLANIF interface address pool



Configuration Roadmap

The configuration roadmap is as follows:

1. Create two interface address pools on the SwitchA and set attributes of the address pool. Configure the interface address pools to enable the DHCP server to assign IP addresses and configuration parameters to hosts from different interface address pools.
2. Configure VLANIF interfaces to assign IP addresses to hosts from the interface address pool.

Procedure

Step 1 Enable DHCP

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] dhcp enable
```

Step 2 Adds the interface to the VLAN

Add GE1/0/1 to VLAN 10.

```
[SwitchA] vlan batch 10 to 11
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 10
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 10
[SwitchA-GigabitEthernet1/0/1] quit
```

Add GE1/0/2 to VLAN 11.

```
[SwitchA] interface gigabitethernet 1/0/2
[SwitchA-GigabitEthernet1/0/2] port hybrid pvid vlan 11
```

```
[SwitchA-GigabitEthernet1/0/2] port hybrid untagged vlan 11
[SwitchA-GigabitEthernet1/0/2] quit
```

Step 3 Assign IP addresses to VLANIF interfaces

Assign an IP address to VLANIF 10.

```
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] ip address 10.1.1.1 24
[SwitchA-Vlanif10] quit
```

Allocate an IP address to VLANIF 11.

```
[SwitchA] interface vlanif 11
[SwitchA-Vlanif11] ip address 10.1.2.1 24
[SwitchA-Vlanif11] quit
```

Step 4 Enable the VLANIF interface address pool

Configure clients on VLANIF 10 to obtain IP addresses from the interface address pool.

```
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] dhcp select interface
[SwitchA-Vlanif10] quit
```

Configure clients on VLANIF 11 to obtain IP addresses from the interface address pool.

```
[SwitchA] interface vlanif 11
[SwitchA-Vlanif11] dhcp select interface
[SwitchA-Vlanif11] quit
```

Step 5 Configure the DNS service and NetBIOS service for the interface address pool

Configure the DNS service and NetBIOS service for the interface address pool on VLANIF 10.

```
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] dhcp server domain-name huawei.com
[SwitchA-Vlanif10] dhcp server dns-list 10.1.1.2
[SwitchA-Vlanif10] dhcp server nbns-list 10.1.1.3
[SwitchA-Vlanif10] dhcp server excluded-ip-address 10.1.1.2
[SwitchA-Vlanif10] dhcp server excluded-ip-address 10.1.1.3
[SwitchA-Vlanif10] dhcp server netbios-type b-node
[SwitchA-Vlanif10] quit
```

Step 6 Set IP address leases of IP address pools

Set the IP address lease of VLANIF 10 address pool to 30 days.

```
[SwitchA] interface vlanif 10
[SwitchA-Vlanif10] dhcp server lease day 30
[SwitchA-Vlanif10] quit
```

Set the IP address lease of VLANIF 11 address pool to 20 days.

```
[SwitchA] interface vlanif 11
[SwitchA-Vlanif11] dhcp server lease day 20
[SwitchA-Vlanif11] quit
```

Step 7 Verify the configuration

Run the **display ip pool interface** command on SwitchA to view interface address pool configuration.

```
[SwitchA] display ip pool interface vlanif 10
  Pool-name      : Vlanif10
  Pool-No        : 0
```

```
  Lease          : 30 Days 0 Hours 0 Minutes
  Domain-name    : huawei.com
  DNS-server0    : 10.1.1.2
  NBNS-server0   : 10.1.1.3
  Netbios-type   : b-node
  Position       : Interface       Status           : Unlocked
  Gateway-0      : 10.1.1.1
  Mask           : 255.255.255.0
  VPN instance   : --
 ------------------------------------------------------------------------------
         Start           End     Total  Used  Idle(Expired)  Conflict  Disable
 ------------------------------------------------------------------------------
       10.1.1.1       10.1.1.254  253    1       250(0)         0        2
 ------------------------------------------------------------------------------
[SwitchA] display ip pool interface vlanif 11
  Pool-name      : Vlanif11
  Pool-No        : 1
  Lease          : 20 Days 0 Hours 0 Minutes
  Domain-name    : -
  DNS-server0    : -
  NBNS-server0   : -
  Netbios-type   : -
  Position       : Interface       Status           : Unlocked
  Gateway-0      : 10.1.2.1
  Mask           : 255.255.255.0
  VPN instance   : --
 ------------------------------------------------------------------------------
         Start           End     Total  Used  Idle(Expired)  Conflict  Disable
 ------------------------------------------------------------------------------
       10.1.2.1      10.1.2.254   253     3       250(0)         0         0
 ------------------------------------------------------------------------------
```

----End

Configuration Files

Configuration file of SwitchA

```
#
 sysname Quidway
#
 vlan batch 10 to 11
#
dhcp enable
#
interface Vlanif10
 ip address 10.1.1.1 255.255.255.0
 dhcp select interface
 dhcp server excluded-ip-address 10.1.1.2 10.1.1.3
 dhcp server lease day 30 hour 0 minute 0
 dhcp server dns-list 10.1.1.2
 dhcp server netbios-type b-node
 dhcp server nbns-list 10.1.1.3
 dhcp server domain-name huawei.com
#
interface Vlanif11
 ip address 10.1.2.1 255.255.255.0
 dhcp select interface
 dhcp server lease day 20 hour 0 minute 0
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 10
 port hybrid untagged vlan 10
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 11
 port hybrid untagged vlan 11
```

```
#
return
```

4.8.3 Example for Configuring a DHCP Server and a DHCP Relay Agent

Networking Requirements

When the DHCP server and clients are on different network segments, a DHCP relay agent is required.

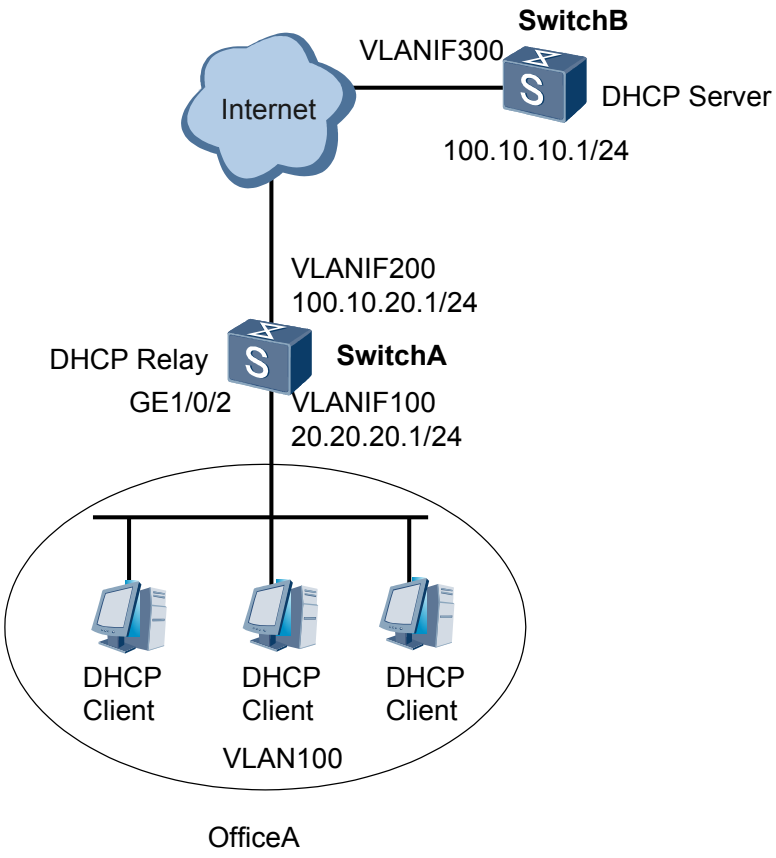
As shown in **Figure 4-6**, an enterprise has multiple offices, which are distributed in different office buildings. The offices in different buildings belong to different VLANs. The enterprise uses SwitchB, which functions as the DHCP server, to assign IP addresses to hosts in different offices.

Hosts in OfficeA are on 20.20.20.0/24 and the DHCP server is on 100.10.10.0/24. By using SwitchA enabled with DHCP relay, the DHCP clients can obtain IP addresses from the DHCP server.

On SwitchA, the public address of VLANIF200 is 100.10.20.1/24 and the interface address of SwitchA connected to the carrier device is 100.10.20.2/24.

On SwitchB, the public address of VLANIF300 is 100.10.10.1/24 and the interface address of SwitchB connected to the carrier device is 100.10.10.2/24.

Figure 4-6 DHCP relay agent



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure DHCP relay on SwitchA to enable SwitchA to forward DHCP messages from different network segments.
2. Configure a global address pool at 20.20.20.0/24 to enable the DHCP server to assign IP address to clients on different network segments.

Procedure

Step 1 Configure DHCP relay on SwitchA.

1. Create a DHCP server group and add DHCP servers to the group.

   # Create a DHCP server group.

   ```
   <Quidway> system-view
   [Quidway] sysname SwitchA
   [SwitchA] dhcp server group dhcpgroup1
   ```

   # Add a DHCP server to the DHCP server group.

   ```
   [SwitchA-dhcp-server-group-dhcpgroup1] dhcp-server 100.10.10.1
   [SwitchA-dhcp-server-group-dhcpgroup1] quit
   ```

2. Enable DHCP relay on the interface.

   # Create a VLAN and add GE1/0/2 to the VLAN.

   ```
   [SwitchA] vlan batch 100 200
   [SwitchA] interface gigabitethernet 1/0/2
   [SwitchA-GigabitEthernet1/0/2] port hybrid pvid vlan 100
   [SwitchA-GigabitEthernet1/0/2] port hybrid untagged vlan 100
   [SwitchA-GigabitEthernet1/0/2] quit
   ```

   # Enable DHCP globally and DHCP relay on the interface.

   ```
   [SwitchA] dhcp enable
   [SwitchA] interface vlanif 100
   [SwitchA-Vlanif100] dhcp select relay
   [SwitchA-Vlanif100] quit
   ```

3. Bind an interface to a DHCP server group.

   # Assign IP addresses to interfaces.

   ```
   [SwitchA] interface vlanif 100
   [SwitchA-Vlanif100] ip address 20.20.20.1 24
   ```

   Bind the interface to the DHCP server group.

   ```
   [SwitchA-Vlanif100] dhcp relay server-select dhcpgroup1
   [SwitchA-Vlanif100] quit
   ```

Step 2 Configure a default route on SwitchA.

```
[SwitchA] interface vlanif 200
[SwitchA-Vlanif200] ip address 100.10.20.1 24
[SwitchA-Vlanif200] quit
[SwitchA] ip route-static 0.0.0.0 0.0.0.0 100.10.20.2
```

Step 3 Configure the DHCP server based on the global address pool on SwitchB.

Enable DHCP.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] dhcp enable
```

Configure VLANIF300 to use the global address pool.

```
[SwitchB] vlan 300
[SwitchB-vlan300] quit
[SwitchB] interface vlanif 300
[SwitchB-Vlanif300] ip address 100.10.10.1 24
[SwitchB-Vlanif300] dhcp select global
[SwitchB-Vlanif300] quit
```

Create an address pool and set the attributes of the address pool.

```
[SwitchB] ip pool pool1
[SwitchB-ip-pool-pool1] network 20.20.20.0 mask 24
[SwitchB-ip-pool-pool1] gateway-list 20.20.20.1
[SwitchB-ip-pool-pool1] quit
```

Step 4 Configure a default route on SwitchB.

```
[SwitchB] ip route-static 0.0.0.0 0.0.0.0 100.10.10.2
```

Step 5 Verify the configuration.

Run the **display dhcp relay interface vlanif 100** command on SwitchA to view the DHCP relay configuration on the interface.

```
[SwitchA] display dhcp relay interface vlanif 100
 DHCP relay agent running information of interface Vlanif100 :
 Server group name : dhcpgroup1
 Gateway address in use : 20.20.20.1
```

Run the **display ip pool** command on SwitchB to view the IP address pool configuration.

```
[SwitchB] display ip pool
   ------------------------------------------------------------------------
  Pool-name      : pool1
  Pool-No        : 0
  Position       : Local           Status           : Unlocked
  Gateway-0      : 20.20.20.1
  Mask           : 255.255.255.0
  VPN instance   : --


 IP address Statistic
    Total       :253
    Used        :2          Idle        :251
    Expired     :0          Conflict    :0          Disable   :0
```

----End

Configuration Files

Configuration file of SwitchA

```
#
 sysname SwitchA
#
vlan batch 100 200
#
 dhcp enable
#
dhcp server group dhcpgroup1
```

```
 dhcp-server 100.10.10.1 0
#
interface Vlanif100
 ip address 20.20.20.1 255.255.255.0
 dhcp select relay
 dhcp relay server-select dhcpgroup1
#
interface Vlanif200
 ip address 100.10.20.1 255.255.255.0
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
 ip route-static 0.0.0.0 0.0.0.0 100.10.20.2
#
return
```

Configuration file of SwitchB

```
#
 sysname SwitchB
#
vlan batch 300
#
 dhcp enable
#
ip pool pool1
 gateway-list 20.20.20.1
 network 20.20.20.0 mask 255.255.255.0
#
interface Vlanif300
 ip address 100.10.10.1 255.255.255.0
 dhcp select global
#
 ip route-static 0.0.0.0 0.0.0.0 100.10.10.2
#
return
```

4.9 Common Configuration Errors

This section provides DHCP troubleshooting procedures.

4.9.1 DHCP Client Cannot Obtain IP Addresses When switch Functions as the DHCP Server

Fault Description

When the switch functions as the DHCP server, the DHCP client cannot obtain IP addresses.

Procedure

Step 1 Run the **display current-configuration | include dhcp enable** command to check whether DHCP is enabled. By default, DHCP is disabled.

- If no DHCP information is displayed, DHCP is disabled. Run the **dhcp enable** command to enable DHCP.
- If **dhcp enable** is displayed, DHCP is enabled. Go to step 2.

Step 2 In the switch interface view, run the **display this** command to check whether the DHCP address assignment mode is set.

Command Output	Description	Follow-up Operation
dhcp select global	The DHCP server has assigned IP addresses to clients from the global address pool.	Go to step 3.
dhcp select interface	The DHCP server has assigned IP addresses to clients from the interface address pool.	Go to step 4.
The preceding information is not displayed.	The DHCP address assignment mode is not set on the VLANIF interface.	Run the **dhcp select global** or **dhcp select interface** command to set the DHCP address assignment mode on the interface.

Step 3 Run the **display ip pool** command to check whether the global address pool has been created.

- If the global address pool has not been created, run the **ip pool** *ip-pool-name* and **network** *ip-address* [**mask** { *mask* | *mask-length* }] commands to create a global address pool and set the range of IP addresses that can be dynamically assigned.
- If the global address pool has been created, obtain the value of *ip-pool-name*. Then run the **display ip pool name** *ip-pool-name* command to check whether the IP addresses in the global address pool are on the same network segment with the IP address on the interface.
  - If the client and server are located on the same network segment and no relay agent is deployed:
    - If IP addresses in the global address pool and the VLANIF interface IP address are located on different network segments, run the **ip address** *ip-address* { *mask* | *mask-length* } [**sub**] command to change the VLANIF interface IP address to be on the same network segment as IP addresses in the global address pool.
    - If IP addresses in the global address pool and the switch interface IP address are located on the same network segment, go to step 4.
  - If the client and server are located on different network segments and a relay agent is deployed:
    - If IP addresses in the global address pool and the relay agent IP address are located on different network segments, run the **ip address** *ip-address* { *mask* | *mask-length* } [**sub**] command to change the IP address to be on the same network segment as IP addresses in the global address pool.
    - If IP addresses in the global address pool and the relay agent interface IP address are located on the same network segment, go to step 4.

Step 4 Run the **display ip pool** [{ **interface** *interface-pool-name* | **name** *ip-pool-name* } [*start-ip-address* [*end-ip-address*] | **all** | **conflict** | **expired** | **used**]] command to check the usage of IP addresses in the global or interface address pool. If the value of **Idle (Expired)** is 0, IP addresses in the address pool have been used up.

- If the server assigns IP addresses to clients from the global address pool on the interface, re-create a global address pool where the network segment can be connected to the previous network segment but cannot overlap with the previous network segment.
- If the DHCP server allocates IP addresses to clients from the interface address pool, you can reduce the mask length of IP address so that more IP addresses can be allocated.

----End

4.9.2 DHCP Client Cannot Obtain IP Addresses When switch Functions as the DHCP Relay Agent

Fault Description

When the switch functions as the DHCP relay agent, the DHCP client cannot obtain IP addresses.

Procedure

Step 1 Run the **display current-configuration | include dhcp enable** command to check whether DHCP is enabled. By default, DHCP is disabled.

- If no DHCP information is displayed, DHCP is disabled. Run the **dhcp enable** command to enable DHCP.
- If **dhcp enable** is displayed, DHCP is enabled.

Step 2 In the switch interface view, run the **display this** command to check whether the DHCP relay function is enabled.

- If **dhcp select relay** is displayed, the DHCP relay function is enabled. Go to step 3.
- If no information is displayed, the DHCP relay function is disabled. Then run the **dhcp select relay** command to enable the DHCP relay function.

Step 3 In the switch interface view, run the **display this** command to check whether the DHCP server is configured on the DHCP relay agent.

- If **dhcp relay server-ip** *ip-address* is displayed, the DHCP server IP address is configured on the DHCP relay agent.
- If **dhcp relay server-select** *group-name* is displayed, the interface on the DHCP relay agent is bound to a DHCP server group. Go to step 4.
- If no information is displayed, the DHCP server IP address is not configured on the DHCP relay agent. Configure the DHCP server using either of the following methods:
  - Run the **dhcp relay server-ip** *ip-address* command to configure the DHCP server IP address on the DHCP relay agent.
  - Run the **dhcp-server** command to add DHCP servers to the DHCP server group and run the **dhcp relay server-select** *group-name* command to bind the VLANIF interface to a DHCP server group.

Step 4 Run the **display dhcp server group** *group-name* command to check whether DHCP servers are configured in the DHCP server group.

- If the **Server-IP** field is displayed, DHCP servers are configured in the DHCP server group.

- If the **Server-IP** field is not displayed, DHCP servers are not configured in the DHCP server group. Then run the **dhcp-server** command to add DHCP servers to the DHCP server group.

----End

5 IP Session Configuration

About This Chapter

An IP session, also called DHCP proxy, is an application of the DHCP protocol. It manages DHCP users.

5.1 IP Session Overview
IP session, also called DHCP proxy, is an application of the DHCP protocol. It manages DHCP users.

5.2 Configuring an IP Session
Configuring an IP session includes binding a user authentication domain to an interface, configuring the DHCP user name and password, configuring the processing of option fields, configuring ARP probe parameters, setting the NAS interface type, and binding an interface to a VPN instance.

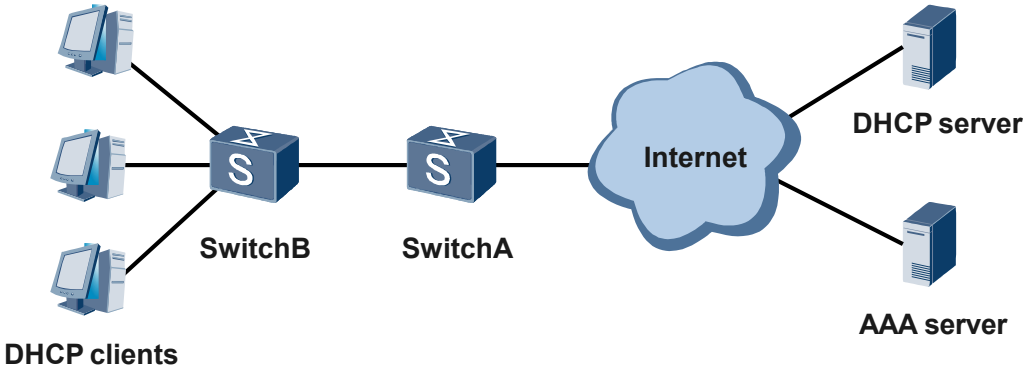
5.3 Configuration Examples
This section describes how to configure the IP session function.

5.1 IP Session Overview

IP session, also called DHCP proxy, is an application of the DHCP protocol. It manages DHCP users.

To flexibly manage DHCP users and meet user requirements on the network, you can use local authentication, remote authentication (RADIUS or HWTACACS authentication), and non-authentication to authenticate and authorize users during user access. When the user access status is stable, you can manage specified users, for example, charge the users and force the users to go offline.

Figure 5-1 Typical application of IP session



As shown in **Figure 5-1**, SwitchB is the access device of DHCP users, the IP session service runs on the aggregation SwitchA. SwitchA allocates IP addresses to users through the DHCP server, and the AAA server authenticates and authorizes users. Then the users can go online once they power on their computers. The DHCP server can be a remote server or a local server. If a local server is used, the S7700&S9700 functions as the DHCP server.

NOTE

IP session and L2VPN cannot be configured simultaneously on an sub-interface.

The S7700&S9700 supports the access of IP sessions to Layer 3 sub-interfaces, but does not support the access to main interfaces.

For the configurations of AAA and user management function on the S7700&S9700, see the *S7700&S9700 Smart&Core Routing Switch Configuration Guide - Security*.

5.2 Configuring an IP Session

Configuring an IP session includes binding a user authentication domain to an interface, configuring the DHCP user name and password, configuring the processing of option fields, configuring ARP probe parameters, setting the NAS interface type, and binding an interface to a VPN instance.

Pre-configuration Tasks

Before configuring an IP session, complete the following tasks:

- Setting physical parameters of a sub-interface.
- To enable users to go online successfully, you must configure a static route between the egress gateway and DHCP server on the device.

NOTE

Creating a VPN instance before binding an interface to the VPN instance

Configuration Flowchart

You must enable the IP session function before performing the following configurations: bind a user authentication domain to an interface, configure the DHCP user name and password, configure the processing of option fields, configure ARP probe parameters, set the NAS interface type, and bind an interface to a VPN instance. The preceding configurations can be performed in any sequence.

5.2.1 Enabling the IP Session Function

Context

The S7700&S9700 can terminate DHCP packets on sub-interfaces. You can configure a remote or local DHCP server to allocate IP addresses. After the IP session function is enabled on a sub-interface, you can configure related parameters and functions on the sub-interface to effectively manage DHCP users.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled globally.

Step 3 Run:

```
interface interface-type interface-number.subnumber
```

The sub-interface view is displayed.

Currently, the IP session function can only be enabled on the S7700&S9700 in the GE sub-interface view, Eth-Trunk sub-interface view, or Ethernet sub-interface view.

NOTE

If an Eth-Trunk interface includes member interfaces that do not support the IP session function, the IP session function does not take effect on sub-interfaces of the Eth-Trunk.

Step 4 Run:

```
ip-session enable
```

The IP session function is enabled.

By default, the IP session function is disabled.

----End

5.2.2 Binding a User Authentication Domain to an Interface

Context

If a user authentication domain is bound to a sub-interface, when a user goes online, the device selects the bound domain to authenticate, authorize the user and charge the user.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

aaa

The AAA view is displayed.

Step 3 Run:

domain *domain-name*

A user authentication domain is created.

Step 4 Run:

quit

The AAA view is displayed.

Step 5 Run:

quit

Return to the system view.

Step 6 Run:

interface *interface-type interface-number.subnumber*

The sub-interface view is displayed.

Step 7 Run:

authentication-domain *domain-name*

The user authentication domain is bound to the sub-interface.

By default, the global default user authentication domain is bound to a sub-interface.

----End

5.2.3 (Optional) Configuring the DHCP User Name and Password

Context

The DHCP user name and password must be configured during server authentication.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp user-name format-include { ip-address | mac-address | option82 | sysname } *
```

The DHCP user name format and the sequence of elements in the user name are specified.

By default, the DHCP user name is in the following format: system name + "-" + slot ID (two digits, prefixed 0 if it contains only one digit) + subcard ID (one digit, set to 0 if the subcard does not exist) + port number (two digits, prefixed 0 if it contains only one digit) + outer VLAN ID (four digits, prefixed 0 if it contains less than four digits) + inner VLAN ID (five digits, prefixed 0 if it contains less than five digits) + @ + access domain name, for example, Quidway-02024000000768@domain1.

If the user name is generated according to the Option 82 field and the user name contains non-ASCII characters, the non-ASCII characters are displayed as "..." for example, ...session1@domain1.

Step 3 Run:

```
dhcp user-password cipher password
```

The DHCP user password is set.

By default, the DHCP user password is vlan.

----End

5.2.4 (Optional) Configuring the S7700&S9700 to Process Option Fields

Context

When DHCP snooping is configured on the S7700&S9700, the S7700&S9700 can add the Option 82 field in DHCP Discover and DHCP Request messages sent from users. The Option 82 field records physical information about a user, such as the interface number and VLAN ID. The information is used by an upper-layer authentication server to authenticate the user. The S7700&S9700 can select service policies based on Option 60 information. The user access IP address and information required for authentication and authorization are obtained from the Option 60 information.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number.subnumber
```

The sub-interface view is displayed.

Step 3 Perform either of the following operations as required:

- Run:

  dhcp option82 insert enable

  The S7700&S9700 is enabled to add the Option 82 field in DHCP messages.

- Run:

  dhcp option82 rebuild enable

  The function that forcibly adds the Option 82 field to DHCP messages is enabled.

By default, the S7700&S9700 does not process the Option 82 field of DHCP messages.

NOTE

After the **dhcp option82 rebuild enable** command is used, the S7700&S9700 replaces the Option 82 field of DHCP messages sent from online IP session users. The Option 82 field is generated according to the configuration and is used to send DHCP messages to the remote DHCP server.

Step 4 Run:

dhcp service-policy option60

The service policy is configured.

By default, users connected to a sub-interface go online through the service scheme in the domain bound to the sub-interface.

----End

5.2.5 (Optional) Setting ARP Probe Parameters

Context

Using the DHCP protocol, a server leases IP addresses to clients. Clients need to apply for new IP addresses when the leases expire. In practice, a client who already has a leased IP address does not send a Release message to the DHCP server after going offline unexpectedly.

In this case, the S7700&S9700 needs to periodically send ARP probes to check whether users are online. When ARP probes expire, users are disconnected. In addition, DHCP Release messages are constructed and sent to the DHCP server to enable the DHCP server to release the IP addresses of the users.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

interface *interface-type interface-number.subnumber*

The sub-interface view is displayed.

Step 3 Run:

dhcp user-detect retransmit *times* **interval** *interval*

The interval for sending ARP probes and the maximum number of timeouts are set.

By default, the interval for sending ARP probes is 30 seconds and the maximum number of timeouts is 5.

----End

5.2.6 (Optional) Setting the NAS Interface Type

Context

During AAA authentication, you can set the NAS interface type on a sub-interface. When a user is being authenticated, the NAS interface type encapsulated in the RADIUS attribute is reported.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number.subnumber
```

The sub-interface view is displayed.

Step 3 Run:

```
dhcp nas-port-type { 802.11 | adsl-cap | adsl-dmt | async | cable | ethernet | g.3-fax | hdlc | idsl | isdn-async-v110 | isdn-async-v120 | isdn-sync | piafs | sdsl | sync | virtual | wireless-other | x.25 | x.75 | xdsl }
```

The NAS interface type is set.

By default, the NAS interface type is **ethernet**.

----End

5.2.7 (Optional) Binding VPN Instance to an Interface

Context

After a VPN instance is bound to a sub-interface, access users can go online only when the VPN instance of the IP address pool used by the users is the same as the VPN instance bound to the sub-interface.

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.

Step 3 Run:

route-distinguisher *route-distinguisher*

A route distinguisher (RD) for a VPN instanceaddress family is configured.

Step 4 Run:

quit

Return to the system view.

Step 5 Run:

interface *interface-type interface-number.subnumber*

The sub-interface view is displayed.

Step 6 Run:

vpn-instance *vpn-instance-name*

The VPN instance is bound to the sub-interface.

----End

5.2.8 Checking the Configuration

Procedure

- Run the **display session-interface** [*interface-type interface-number* [*.subnumber*]] command to view the status of the sub-interface on which the IP session service is enabled.

----End

5.3 Configuration Examples

This section describes how to configure the IP session function.

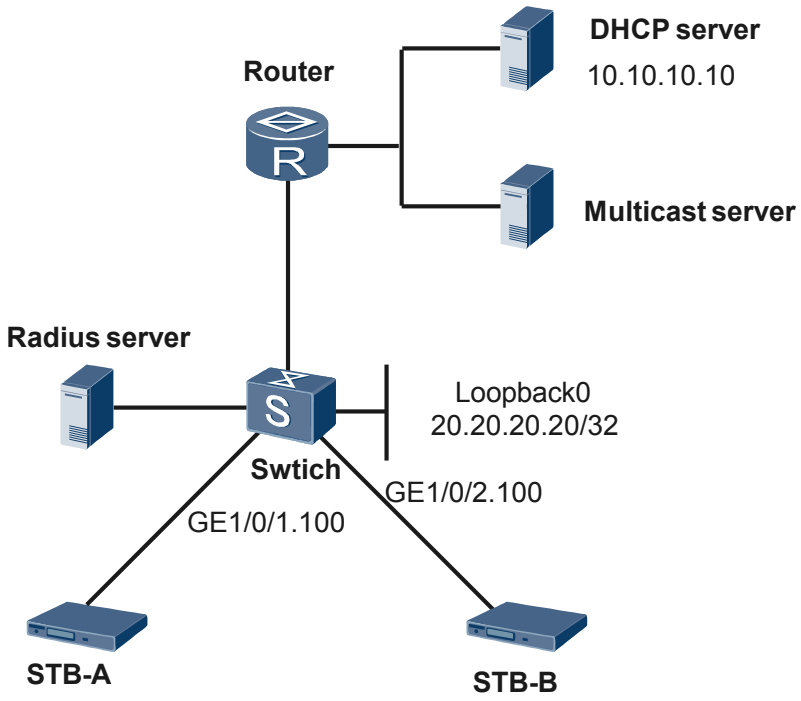
5.3.1 Example for Configuring the IP Session Function

Networking Requirements

As shown in **Figure 5-2**, STB-A (a set top box) connects to GE1/0/1.100 of the switch and STB-B (the other set top box) connects to GE1/0/2.100 of the switch.

STB-A and STB-B users need to go online immediately after they power on their computers.

Figure 5-2 Networking diagram of IPTV



Configuration Roadmap

On the switch, enable the IP session function and bind user authentication domains to sub-interfaces to enable STB-A and STB-B users online immediately after they power on their computers.

NOTE

Only IP session-related configurations are involved in this example. The AAA configuration, RADIUS configuration, multicast configuration, and router configuration are not mentioned in this example.

Procedure

Step 1 Enable DHCP globally.

```
<Quidway> system-view
[Quidway] dhcp enable
```

Step 2 Enable the IP session function on sub-interfaces.

Enable the IP session function on GE1/0/1.100.

```
[Quidway] interface gigabitethernet 1/0/1.100
[Quidway-GigabitEthernet1/0/1.100] ip-session enable
[Quidway-GigabitEthernet1/0/1.100] quit
```

Enable the IP session function on GE1/0/2.100.

```
[Quidway] interface gigabitethernet 1/0/2.100
[Quidway-GigabitEthernet1/0/2.100] ip-session enable
[Quidway-GigabitEthernet1/0/2.100] quit
```

Step 3 Bind user authentication domains to sub-interfaces.

Bind user authentication domain **stb-a** to GE1/0/1.100.

```
[Quidway] aaa
[Quidway-aaa] domain stb-a
[Quidway-aaa-domain-stb-a] quit
[Quidway-aaa] quit
[Quidway] interface gigabitethernet 1/0/1.100
[Quidway-GigabitEthernet1/0/1.100] authentication-domain stb-a
[Quidway-GigabitEthernet1/0/1.100] quit
```

Bind user authentication domain **stb-b** to GE1/0/2.100.

```
[Quidway] aaa
[Quidway-aaa] domain stb-b
[Quidway-aaa-domain-stb-b] quit
[Quidway-aaa] quit
[Quidway] interface gigabitethernet 1/0/2.100
[Quidway-GigabitEthernet1/0/2.100] authentication-domain stb-b
[Quidway-GigabitEthernet1/0/2.100] quit
```

Step 4 Set sub-interface-related parameters on the Switch.

Set the detection interval for sending ARP probes to 60s and the number of ARP probe timeouts to 8 for GE1/0/1.100.

```
[Quidway] interface gigabitethernet 1/0/1.100
[Quidway-GigabitEthernet1/0/1.100] control-vid 100 dot1q-termination
[Quidway-GigabitEthernet1/0/1.100] dot1q termination vid 100
[Quidway-GigabitEthernet1/0/1.100] dhcp user-detect retransmit 8 interval 60
[Quidway-GigabitEthernet1/0/1.100] quit
```

Set the detection interval for sending ARP probes to 60s and the number of ARP probe timeouts to 8 for GE1/0/2.100.

```
[Quidway] interface gigabitethernet 1/0/2.100
[Quidway-GigabitEthernet1/0/2.100] control-vid 100 dot1q-termination
[Quidway-GigabitEthernet1/0/2.100] dot1q termination vid 100
[Quidway-GigabitEthernet1/0/2.100] dhcp user-detect retransmit 8 interval 60
[Quidway-GigabitEthernet1/0/2.100] quit
```

Step 5 Set the format of the DHCP user name to **mac-address** and the password to **stb** in cipher text.

```
[Quidway] dhcp user-name format-include mac-address
[Quidway] dhcp user-password cipher stb
```

Step 6 Configure a DHCP server group.

```
[Quidway] dhcp server group dhcp-group
[Quidway-dhcp-server-group-dhcp-group] dhcp-server 10.10.10.10
[Quidway-dhcp-server-group-dhcp-group] gateway 20.20.20.20
[Quidway-dhcp-server-group-dhcp-group] quit
```

Step 7 Configure an egress gateway.

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 20.20.20.20 32
[Quidway-LoopBack0] quit
```

Step 8 Configure a static route.

```
[Quidway] ip route-static 10.10.10.0 255.255.255.0 NULL 0
```

Step 9 Verify the configuration.

Check the IP session configuration on GE1/0/1.100.

```
<Quidway> display session-interface gigabitethernet 1/0/1.100
```

```
  Access type                           : Enable
  IPSessIF state                        : Updated
  Authentication default domain         : stb-a
  Nas port type                         : ethernet (15)
  Vpn Instance                          :
  User detect interval                  : 60 (s)
  User detect retransmit times          : 8
  Option82 policy                       : none (0)
  Service policy                        : default (0)
```

Check the IP session configuration on GE1/0/2.100.

```
<Quidway> display session-interface gigabitethernet 1/0/2.100

  Access type                           : Enable
  IPSessIF state                        : Updated
  Authentication default domain         : stb-b
  Nas port type                         : ethernet (15)
  Vpn Instance                          :
  User detect interval                  : 60 (s)
  User detect retransmit times          : 8
  Option82 policy                       : none (0)
  Service policy                        : default (0)
```

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
 dhcp enable
#
 dhcp server group dhcp-group
#
 dhcp user-name format-include mac-address
 dhcp user-password cipher stb
#
dhcp server group dhcp-group
 dhcp-server 10.10.10.10 0
 gateway 20.20.20.20
#
aaa
 domain stb-a
 domain stb-b
#
interface GigabitEthernet1/0/1.100
 ip-session enable
 authentication-domain stb-a
 dhcp user-detect retransmin 8 interval 60
 control-vid 100 dot1q-termination
 dot1q termination vid 100
#
interface GigabitEthernet1/0/2.100
 ip-session enable
 authentication-domain stb-b
 dhcp user-detect retransmin 8 interval 60
 control-vid 100 dot1q-termination
 dot1q termination vid 100
#
interface NULL0
#
interface LoopBack0
 ip address 20.20.20.20 255.255.255.255
```

```
#
 ip route-static 10.10.10.0 255.255.255.0 NULL0
#
return
```

6 DHCPv6 Configuration

About This Chapter

This section describes how to configure the DHCPv6 function. Currently, the switch can function as the DHCPv6 server, DHCPv6 PD server, DHCPv6 relay on the IPv6 network.

6.1 DHCPv6 Overview

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) is a technology that dynamically manages and configures IPv6 addresses in a centralized manner. It is designed to assign IPv6 addresses and other network configuration parameters to hosts. DHCPv6 uses the client/server model. A client requests configurations such as the IPv6 address and DNS server address from the server, the server replies with requested configurations based on policies.

The IPv6 protocol provides huge address space formed by 128-bit IPv6 addresses that require proper and efficient policies for assignment and management. IPv6 stateless address autoconfiguration (for details, see RFC 2462) is widely used. Hosts configured with the stateless address autoconfiguration function automatically configure IPv6 addresses based on prefixes carried in the Route Advertisement (RA) message sent from a neighboring switch.

In stateless address autoconfiguration, routers do not record IPv6 addresses of the hosts. Therefore, the stateless address autoconfiguration has poor manageability. In addition, hosts configured with the stateless address autoconfiguration cannot obtain other configuration parameters such as the DNS server address. ISPs do not provide instructions for automatic allocation of IPv6 prefixes for routers. Therefore, users need to manually configure IPv6 addresses for devices during IPv6 network deployment.

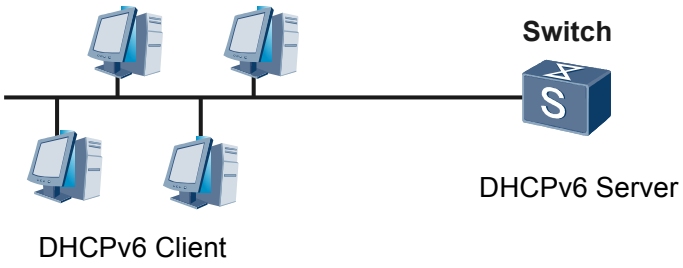
DHCPv6 solves this problem. DHCPv6 is a stateful protocol for configuring IPv6 addresses automatically. During stateful address configuration, the DHCPv6 server assigns a complete IPv6 address to a host and provides other configuration parameters such as the DNS server address and domain name. The relay agent may be used to forward DHCPv6 packets. The DHCPv6 server binds the IPv6 address to a client. This improves network manageability.

6.2 DHCPv6 Features Supported by the Device

The switch can be used as the DHCPv6 server, DHCPv6 PD server, DHCPv6 relay.

Using the switch as a DHCPv6 Server

Figure 6-1 Networking of the DHCPv6 server

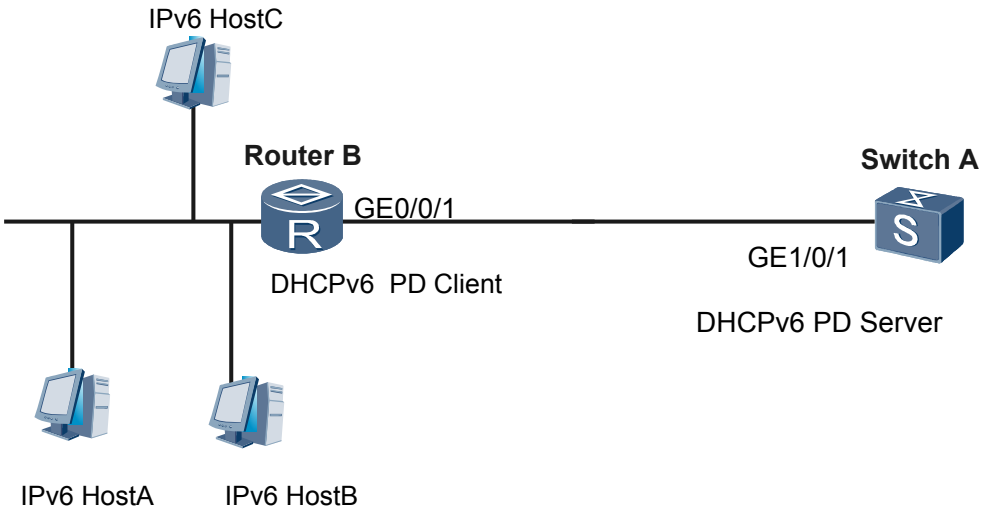


The switch in the figure functions as the DHCPv6 server to assign IPv6 addresses to clients. The DHCPv6 client applies to the DHCPv6 server for configurations including an IPv6 address and DNS server address. The DHCPv6 server replies with related configurations according to policies.

The DHCPv6 server assigns a complete IPv6 address to a host and provides other configuration parameters such as the DNS server address. The DHCPv6 server also provides stateless DHCPv6 services. That is, the DHCPv6 server does not assign IPv6 addresses but provides hosts with configuration parameters such as the DNS server address and domain name. Hosts automatically configure IPv6 addresses based on RA messages. This overcomes the limitations of IPv6 stateless address autoconfiguration.

Using the switch as a DHCPv6 PD Server

Figure 6-2 Networking of the DHCPv6 PD server



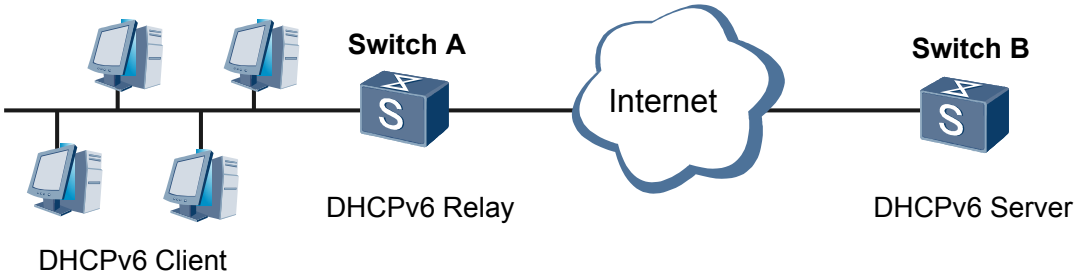
The switch in the figure functions as the DHCPv6 PD server to assign IPv6 address prefixes to DHCPv6 PD clients.

The DHCPv6 PD mechanism allows RouterB to function as a DHCPv6 PD client to request IPv6 prefixes from SwitchA and allows SwitchA to function as a DHCPv6 PD server to assign prefixes to RouterB. In this way, RouterB does not need to assign IPv6 prefixes for user-side links.

RouterB divides the obtained prefix (the length of the obtained prefix is smaller than 64 bits) into 64-bit prefix of subnet segments and sends an RA message on the link that hosts directly connect to. The RA message contains 64-bit prefix of subnet segments. This enables hosts to automatically configure addresses.

Using the switch as a DHCPv6 Relay Agent

Figure 6-3 Networking of the DHCPv6 relay agent



The switch in the figure supports the DHCPv6 relay function. When the switch functions as a DHCPv6 relay agent, the client can communicate with a DHCPv6 server on another network segment through the switch, and obtain an IPv6 address and other configuration parameters from the global address pool on the DHCP server. In this manner, DHCPv6 clients on multiple network segments can share one DHCPv6 server. This reduces costs and facilitates centralized management.

6.3 Default Configuration

This section provides default DHCPv6 configurations.

Table 1 DHCPv6 default configuration

Parameter	Default Value
DHCPv6 Function	disabled
DHCPv6 DUID	based on the link-layer (LL) address
the time for updating IPv6 address pool configurations	86400s (24 hours).

6.4 Configuring a DHCPv6 Server

If a DHCPv6 server is configured, users going online through any interface can obtain IPv6 addresses from the IPv6 address pool of the DHCPv6 server.

Pre-configuration Tasks

Before configuring the DHCPv6 server, complete the following tasks:

- Ensuring that the link between the DHCPv6 client and the switch works properly and the DHCPv6 client can communicate with the switch
- Configuring the route between the switch and DHCPv6 relay agent or client

6.4.1 Configuring the DHCPv6 DUID

Context

The DUID identifies a DHCPv6 device. Each DHCPv6 server or client has a unique DUID. DHCPv6 servers use DUIDs to identify DHCPv6 clients and DHCPv6 clients use DUIDs to identify DHCPv6 servers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 duid { ll | llt }
```

A DUID is configured for the device.

By default, the device generates a DUID based on the link-layer (LL) address.

----End

6.4.2 Configuring an IPv6 Address Pool

Context

To implement the DHCPv6 function, you need to create an IPv6 address pool and configure its attributes including the IPv6 address range, IPv6 configuration update time, IPv6 addresses not to be automatically allocated, and IP addresses to be statically bound to clients. IPv6 addresses can be dynamically assigned or statically bound to clients based on client requirements.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 pool pool-name
```

An IPv6 address pool is created and the address pool view is displayed.

By default, no IPv6 address pool is created on the device.

Step 3 Run:

```
address prefix ipv6-prefix/ipv6-prefix-length [ life-time { valid-lifetime | infinite } { preferred-lifetime | infinite } ]
```

An IPv6 address prefix is bound to the address pool.

By default, no IPv6 address prefix is bound to the address pool.

Step 4 (Optional) Run:

```
static-bind address ipv6-address duid client-duid [ iaid iaid-value ] [ life-time { valid-lifetime | infinite } { preferred-lifetime | infinite } ]
```

The IPv6 address is statically bound to the client DUID.

By default, no IPv6 address is bound to client DUID in the address pool view.

Step 5 (Optional) Run:

```
excluded-address start-ipv6-address [ to end-ipv6-address ]
```

The range of the IPv6 addresses that cannot be automatically assigned is specified in the IPv6 address pool. If only one IPv6 address is not automatically assigned, you can specify only the value of *start-ipv6-address*.

By default, all IPv6 addresses in the address pool can be automatically assigned to clients.

Step 6 (Optional) Run:

```
information-refresh time
```

The time is configured for updating configuration parameters assigned to clients through stateless DHCPv6 address autoconfiguration.

By default, the time for updating IPv6 address pool configuration is 86400s (24 hours).

----End

6.4.3 Enabling the DHCPv6 Server Function on an Interface

Context

The DHCPv6 server function is enabled on an interface after an IPv6 address pool is bound to the interface. When the DHCPv6 server receives a DHCPv6 request from a DHCPv6 client, it selects an idle IPv6 address from the bound IPv6 address pool and allocates the IPv6 address to the client.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled.

Step 3 Run:

```
ipv6
```

IPv6 is enabled in the system view.

Step 4 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces on the switch can work in DHCPv6 server mode.

Step 5 Run:

```
ipv6 enable
```

IPv6 is enabled on the interface.

Step 6 Run:

```
ipv6 address { ipv6-address prefix-length |ipv6-address/prefix-length }
```

A global unicast IPv6 address is configured for the interface.

Step 7 Run:

```
dhcpv6 server pool-name [ allow-hint | preference preference-value | rapid-commit | unicast ] *
```

The DHCPv6 server function is enabled on the interface.

----End

6.4.4 (Optional) Configuring Network Server Addresses for the IPv6 Address Pool

Context

To successfully connect DHCPv6 clients to the Internet, the DHCPv6 server needs to specify network service configurations such as the DNS server address and SIP server address when assigning IPv6 addresses to the clients. The DHCPv6 server dynamically allocates carrier-assigned configurations such as the DNS server address and SIP server address to DHCPv6 clients.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 pool pool-name
```

An IPv6 address pool is created and the address pool view is displayed.

By default, no IPv6 address pool is created on the device.

Step 3 In the IPv6 address pool view, you can run one or multiple following commands to configure network server addresses.

1. Run:

   ```
   dns-server ipv6-address
   ```

   The DNS server address is configured for the DHCPv6 address pool.

2. Run:

   ```
   dns-domain-name dns-domain-name
   ```

   The DNS domain name suffix allocated by the DHCPv6 server to the client is configured.

3. Run:
   sip-server *ipv6-address*

   The SIP server IPv6 address is configured for the DHCPv6 address pool.
4. Run:
   sip-domain-name *sip-domain-name*

   The SIP domain name suffix allocated by the DHCPv6 server to the client is configured.
5. Run:
   nis-server *ipv6-address*

   The NIS server IPv6 address is configured for the DHCPv6 address pool.
6. Run:
   nis-domain-name *nis-domain-name*

   The NIS domain name suffix allocated by the DHCPv6 server to the client is configured.
7. Run:
   nisp-server *ipv6-address*

   The NISP server IPv6 address is configured for the DHCPv6 address pool.
8. Run:
   nisp-domain-name *nisp-domain-name*

   The NISP domain name suffix allocated by the DHCPv6 server to the client is configured.
9. Run:
   sntp-server *ipv6-address*

   The SNTP server IPv6 address is configured for the DHCPv6 address pool.

   NOTE
   By default, DNS, SIP, NIS, NISP, and SNTP server addresses are not configured for the IPv6 address pool. A maximum of two addresses and four domain names can be configured for each server in the IPv6 address pool.

----End

6.4.5 (Optional) Configuring the Options of an IPv6 Address Pool

Context

DHCPv6 provides various options. To use these options, add them to the attribute list of the DHCPv6 server manually. If the DHCPv6 server is configured with the vendor-defined Option field, the client can obtain the configuration information in the Option field of the DHCPv6 reply packet from the server when a DHCPv6 client applies for an IPv6 address.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:

dhcpv6 pool *pool-name*

An IPv6 address pool is created and the address pool view is displayed.

By default, no IPv6 address pool is created on the device.

Step 3 Run:

vendor-specific *vendor-id*

Vendor-defined options are configured for the IPv6 address pool and the vendor-defined mode view is displayed.

By default, no vendor-defined option is configured. A maximum of eight vendor-defined options can be configured for one IPv6 address pool.

vendor-id indicates the vendor identifier ID, which is assigned by the IANA. The identifier ID of Huawei is 2011.

Step 4 Run:

suboption *suboption-code* { **address** *ipv6-address &<1-4>* | **ascii** *ascii-string* | **hex** *hex-string* }

Vendor-defined DHCPv6 sub-options are configured in the vendor-defined mode view.

A maximum of 16 vendor-defined sub-options can be configured in the vendor-defined mode view.

----End

6.4.6 Checking the Configuration

Procedure

- Run the **display dhcpv6 duid** command to check the DUID of the DHCPv6 device on the network.
- Run the **display dhcpv6 pool** [*pool-name* [**allocated** { **address** | **prefix** } | **binding** [*duid*] | **conflict address** | *ipv6-address* | *ipv6-prefix/prefix-length*]] command to check IPv6 address pool configurations.
- Run the **display dhcpv6 server** [**database** | [**statistics**] **interface** *interface-type intface-number*] command to check information about the DHCPv6 server function.

----End

6.5 Configuring a DHCPv6 PD Server

If a DHCPv6 PD server is configured, users going online through any interface can obtain IPv6 addresses from the address pool of the DHCPv6 PD server.

Pre-configuration Tasks

Before configuring the DHCPv6 PD server, complete the following tasks:

- Ensuring that the link between the DHCPv6 client and the switch works properly and the DHCPv6 client can communicate with the switch

- Configuring the route between the switch and DHCPv6 relay agent or client

6.5.1 Configuring the DHCPv6 DUID

Context

The DUID identifies a DHCPv6 device. Each DHCPv6 server or client has a unique DUID. DHCPv6 servers use DUIDs to identify DHCPv6 clients and DHCPv6 clients use DUIDs to identify DHCPv6 servers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 duid { ll | llt }
```

A DUID is configured for the device.

By default, the device generates a DUID based on the link-layer (LL) address.

----End

6.5.2 Configuring an IPv6 PD Address Pool

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 pool pool-name
```

An IPv6 PD address pool is created and the address pool view is displayed.

By default, no IPv6 address PD pool is created on the device.

NOTE

Switch supports 512 DHCPv6 PD users.

Step 3 Run:

```
prefix-delegation ipv6-prefix/ipv6-prefix-length assign-prefix-length [ life-time { valid-lifetime | infinite } { preferred-lifetime | infinite }]
```

An IPv6 address prefix agent is bound to the address pool.

By default, no IPv6 address prefix agent is bound to the address pool.

Step 4 Run:

static-bind prefix *ipv6-prefix/ipv6-prefix-length* **duid** *client-duid* [**iaid** *iaid-value*] [**life-time** { *valid-lifetime* | **infinite** } { *preferred-lifetime* | **infinite** }]

An IPv6 address prefix agent is statically bound to the DHCPv6 PD client in the address pool view.

By default, no IPv6 address prefix agent is bound to the DHCPv6 PD client.

----End

6.5.3 (Optional) Configuring Network Server Addresses for the IPv6 Address Pool

Context

To successfully connect DHCPv6 clients to the Internet, the DHCPv6 server needs to specify network service configurations such as the DNS server address and SIP server address when assigning IPv6 addresses to the clients. The DHCPv6 server dynamically allocates carrier-assigned configurations such as the DNS server address and SIP server address to DHCPv6 clients.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

dhcpv6 pool *pool-name*

An IPv6 address pool is created and the address pool view is displayed.

By default, no IPv6 address pool is created on the device.

Step 3 In the IPv6 address pool view, you can run one or multiple following commands to configure network server addresses.

1. Run:

   dns-server *ipv6-address*

   The DNS server address is configured for the DHCPv6 address pool.

2. Run:

   dns-domain-name *dns-domain-name*

   The DNS domain name suffix allocated by the DHCPv6 server to the client is configured.

3. Run:

   sip-server *ipv6-address*

   The SIP server IPv6 address is configured for the DHCPv6 address pool.

4. Run:

   sip-domain-name *sip-domain-name*

   The SIP domain name suffix allocated by the DHCPv6 server to the client is configured.

5. Run:
   nis-server *ipv6-address*

   The NIS server IPv6 address is configured for the DHCPv6 address pool.
6. Run:
   nis-domain-name *nis-domain-name*

   The NIS domain name suffix allocated by the DHCPv6 server to the client is configured.
7. Run:
   nisp-server *ipv6-address*

   The NISP server IPv6 address is configured for the DHCPv6 address pool.
8. Run:
   nisp-domain-name *nisp-domain-name*

   The NISP domain name suffix allocated by the DHCPv6 server to the client is configured.
9. Run:
   sntp-server *ipv6-address*

   The SNTP server IPv6 address is configured for the DHCPv6 address pool.

   NOTE

   By default, DNS, SIP, NIS, NISP, and SNTP server addresses are not configured for the IPv6 address pool. A maximum of two addresses and four domain names can be configured for each server in the IPv6 address pool.

----End

6.5.4 (Optional) Configuring the Options of an IPv6 Address Pool

Context

DHCPv6 provides various options. To use these options, add them to the attribute list of the DHCPv6 server manually. If the DHCPv6 server is configured with the vendor-defined Option field, the client can obtain the configuration information in the Option field of the DHCPv6 reply packet from the server when a DHCPv6 client applies for an IPv6 address.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
dhcpv6 pool *pool-name*

An IPv6 address pool is created and the address pool view is displayed.

By default, no IPv6 address pool is created on the device.

Step 3 Run:
vendor-specific *vendor-id*

Vendor-defined options are configured for the IPv6 address pool and the vendor-defined mode view is displayed.

By default, no vendor-defined option is configured. A maximum of eight vendor-defined options can be configured for one IPv6 address pool.

vendor-id indicates the vendor identifier ID, which is assigned by the IANA. The identifier ID of Huawei is 2011.

Step 4 Run:

```
suboption suboption-code { address ipv6-address &<1-4> | ascii ascii-string | hex hex-string }
```

Vendor-defined DHCPv6 sub-options are configured in the vendor-defined mode view.

A maximum of 16 vendor-defined sub-options can be configured in the vendor-defined mode view.

----End

6.5.5 Enabling the DHCPv6 PD Server Function on an Interface

Context

Enabling the DHCPv6 PD server function on an interface is to bind an IPv6 PD address pool to the interface. When the DHCPv6 PD server receives a DHCPv6 request packet from a DHCPv6 PD client, it selects an appropriate address prefix and allocates the address prefix to the client.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled.

Step 3 Run:

```
ipv6
```

IPv6 is enabled in the system view.

Step 4 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces on the switch can work in DHCPv6 PD server mode.

Step 5 Run:

```
ipv6 enable
```

IPv6 is enabled on the interface.

Step 6 Run:

```
ipv6 address { ipv6-address prefix-length |ipv6-address/prefix-length }
```

A global unicast IPv6 address is configured for the interface.

Step 7 Run:

```
dhcpv6 server pool-name [ allow-hint | preference preference-value | rapid-commit | unicast ] *
```

The DHCPv6 PD server function is enabled on the interface.

----End

6.5.6 Checking the Configuration

Procedure

- Run the **display dhcpv6 duid** command to check the DUID of the DHCPv6 device on the network.
- Run the **display dhcpv6 pool** [*pool-name* [**allocated** { **address** | **prefix** } | **binding** [*duid*] | **conflict address** | *ipv6-address* | *ipv6-prefix/prefix-length*]] command to check IPv6 address pool configurations.
- Run the **display dhcpv6 server** [**database** | [**statistics**] **interface** *interface-type interface-number*] command to check information about the DHCPv6 server function.

----End

6.6 Configuring a DHCPv6 Relay Agent

A DHCPv6 relay agent enables the DHCPv6 client and server on different links to exchange DHCPv6 messages. The DHCPv6 relay agent transparently transmits DHCP messages to the destination DHCPv6 server on a different network segment. DHCPv6 clients on multiple networks can share one DHCPv6 server.

Pre-configuration Tasks

Before configuring the DHCPv6 relay agent, complete the following tasks:

- Configuring a DHCPv6 server
- Configuring a route from the switch to the DHCPv6 server

6.6.1 Configuring the DHCPv6 DUID

Context

The DUID identifies a DHCPv6 device. Each DHCPv6 server or client has a unique DUID. DHCPv6 servers use DUIDs to identify DHCPv6 clients and DHCPv6 clients use DUIDs to identify DHCPv6 servers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 duid { ll | llt }
```

A DUID is configured for the device.

By default, the device generates a DUID based on the link-layer (LL) address.

----End

6.6.2 Enabling the DHCPv6 Relay Function

Context

You can enable the DHCPv6 relay function on an interface of the switch, set the IPv6 address of the DHCPv6 server or the next hop relay agent, and specify the outbound interface of relay packets.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcp enable
```

DHCP is enabled.

Step 3 Run:

```
ipv6
```

The IPv6 packet forwarding function is enabled.

Step 4 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

VLANIF interfaces on the switch can work in DHCPv6 relay mode.

Step 5 Run:

```
ipv6 enable
```

The IPv6 packet forwarding function is enabled.

Step 6 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }
```

An IPv6 address is configured for the interface.

Step 7 Run:

```
dhcpv6 relay destination ipv6-address [ interface interface-type interface-number ]
```

The DHCPv6 relay function is enabled on the interface and the IPv6 address of the DHCPv6 server or next-hop relay agent is configured.

By default, the DHCPv6 relay function is disabled on an interface.

- The configured IPv6 address must be a global unicast IPv6 address or a link-local IPv6 address. The DHCPv6 relay sends relay packets to the configured IPv6 address by searching for a route.
- On the switch, up to 4094 interfaces can be enabled with the DHCPv6 relay function and each interface can be configured with up to 8 destination addresses.

----End

6.6.3 (Optional) Configuring the Remote ID

Context

The remote ID carries information about a client and identifies a client. The DHCPv6 server can determine address allocation, parameter setting, prefix agent according to the remote ID. The format of the remote ID is defined by the vendor. Usually, the remote ID carries the phone number of the caller in a dialup connection, user name, IP address of the peer in a point-to-point connection, and access interface. The maximum length of the remote ID is 247 bytes.

When functioning as the DHCPv6 relay, the switch processes the remote ID in the following way:

- When receiving a message from a DHCPv6 client, the switch adds the Remote-ID option in the Relay-forward message.
- If the Relay-Reply message received by the switch from the DHCPv6 server contains the remote ID, the switch removes the remote ID from the Relay-Reply message before forwarding it to DHCPv6 clients or other relay agents.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dhcpv6 remote-id format { default | user-defined text }
```

The format of the remote ID in DHCPv6 messages is set.

By default, the remote ID in DHCPv6 messages is in default format.

Step 3 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Step 4 Run:

dhcpv6 remote-id insert enable

The function of appending the remote ID to DHCPv6 relay packets is enabled.

Run:

dhcpv6 remote-id rebuild enable

The function of forcibly appending the remote ID to DHCPv6 relay messages is enabled.

----End

6.6.4 (Optional) Configuring Rate Limit of DHCPv6 Messages

Context

To prevent clients or relay agents from sending a large number of messages to attack the switch, the switch limits the rate of DHCPv6 messages to be forwarded.

After rate limit of DHCPv6 messages is enabled, DHCPv6 messages are discarded when the rate of DHCPv6 messages exceeds the limit. When the number of discarded DHCPv6 messages exceeds the threshold, the switch supports the alarm function.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

dhcp enable

DHCP is enabled.

Step 3 Run:

dhcpv6 packet-rate *packet-rate*

Rate limit of DHCPv6 packets is enabled and the rate threshold is configured.

By default, rate limit of DHCPv6 messages is disabled on the switch.

Step 4 Run:

dhcpv6 packet-rate drop-alarm enable

The function of generating logs is enabled on the device.

After the function of generating logs is enabled, if the number of DHCPv6 messages that pass through the switch every second exceeds the rate limit, they are discarded. By default, the device generates a log when the number of discarded DHCPv6 messages exceeds 100.

Step 5 Run:

dhcpv6 packet-rate drop-alarm threshold *threshold*

A log threshold for the number of discarded DHCPv6 messages when the DHCPv6 message rate exceeds the rate threshold is set.

----End

6.6.5 Checking the Configuration

Procedure

- Run the **display dhcpv6 relay** [**interface** *interface-type interface-number*] command to check the configuration of the DHCPv6 relay interface.
- Run the **display dhcpv6 relay statistics** [**interface** *interface-type interface-number*] command to view DHCPv6 message statistics on the DHCPv6 relay agent.

----End

6.7 Maintaining DHCPv6

This section describes how to clear DHCPv6 statistics and monitor DHCPv6 running status after DHCPv6 configurations are complete.

6.7.1 Clearing DHCPv6 Message Statistics on the DHCPv6 Relay Agent

Context

If the switch is enabled with the DHCPv6 relay function, the system collects the statistics about DHCPv6 messages passing through the DHCP relay agent. To clear the statistics about these DHCPv6 messages, use the **reset dhcpv6 relay statistics** command in the user or system view.

CAUTION

DHCPv6 message statistics cannot be restored after being cleared. Confirm your operation before clearing them.

Procedure

- Run the **reset dhcpv6 relay statistics** [**interface** *interface-type interface-number*] command to clear DHCPv6 message statistics on the DHCPv6 relay agent.

  If no interface is specified, all the DHCPv6 message statistics are cleared. If an interface is specified, DHCPv6 message statistics on the specified interface are cleared.

----End

6.7.2 Checking Message Statistics on the DHCPv6 Server

Procedure

- Run the **display dhcpv6 server** [**database** | [**statistics**] [**interface** *interface-type interface-number*]] command to check message statistics on the DHCPv6 server.

----End

6.7.3 Clearing DHCPv6 Message Statistics of the DHCPv6 Server

Procedure

- Run the **reset dhcpv6 server statistics** [**interface** *interface-type interface-number*] command to clear DHCPv6 message statistics of the DHCPv6 server.

----End

6.7.4 Resetting the Status of the IPv6 Address Pool

Procedure

- Run the **reset dhcpv6 pool** *pool-name* [**allocated** { **address** | **prefix** } | **binding** [*duid*] | **conflict address** | *ipv6-address* [**to** *ipv6-address*] | *ipv6-prefix/prefix-length*] command to clear IPv6 address pool configurations.

----End

6.7.5 Monitoring the Running Status of the DHCPv6 Relay Agent

Procedure

- Run the **display dhcpv6 relay** [**interface** *interface-type interface-number*] command to check the configuration of the interface where the DHCPv6 relay function is configured.
- Run the **display dhcpv6 relay statistics** [**interface** *interface-type interface-number*] command to check DHCPv6 message statistics on the DHCPv6 relay agent.

----End

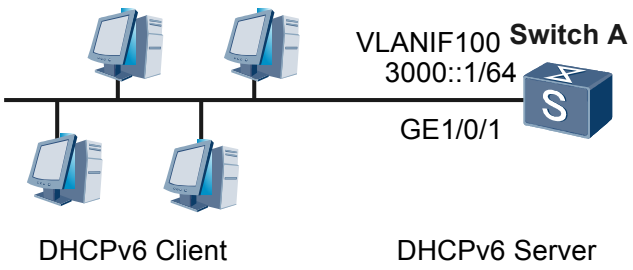
6.8 Configuration Examples

6.8.1 Example for Configuring a DHCPv6 Server

Networking Requirements

If a large number of IPv6 addresses need to be manually configured, the workload on configuration will be huge, and the manually configured addresses have poor manageability. The administrator requires that IPv6 addresses and network configuration parameters be obtained automatically to facilitate centralized management and hierarchical IPv6 network deployment.

Figure 6-4 Networking diagram for configuring the DHCPv6 server



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IPv6 functions on the interface so that devices can communicate using IPv6.
2. Enable the DHCPv6 PD Server function so that devices can obtain IPv6 address prefixes using DHCPv6.

Procedure

Step 1 Enable the DHCP service

```
<Quidway> system-view
[Quidway] sysname Switch A
[Switch A] dhcp enable
```

Step 2 Configure the ipv6 function on interfaces

```
[Switch A] ipv6
[Switch A] vlan 100
[Switch A-vlan100] quit
[Switch A] interface gigabitethernet 1/0/1
[Switch A-GigabitEthernet1/0/1] port link-type access
[Switch A-GigabitEthernet1/0/1] port default  vlan  100
[Switch A-GigabitEthernet1/0/1] quit
[Switch A] interface vlanif 100
[Switch A-Vlanif100] ipv6 enable
[Switch A-Vlanif100] ipv6 address 3000::1/64
[Switch A-Vlanif100] quit
```

Step 3 Configure a DHCPv6 server

```
[Switch A] dhcpv6 pool pool1
[Switch A-dhcpv6-pool-pool1] address prefix 3000::2/64
[Switch A-dhcpv6-pool-pool1] dns-server 4000::1
[Switch A-dhcpv6-pool-pool1] quit
```

Step 4 Enable the DHCPv6 server function on the interface

Enable the DHCPv6 server function on Vlanif100.

```
[Switch A] interface vlanif 100
[Switch A-Vlanif100] dhcpv6 server pool1
```

Step 5 Verify the configuration

Run the **display dhcpv6 pool** command on the switch to check information about the DHCPv6 address pool.

```
<Switch A> display dhcpv6  pool
DHCPv6 pool: pool1
  Address prefix: 3000::/64
    lifetime valid 172800 seconds, preferred 86400 seconds
    0 in use, 0 conflicts
  Information refresh time: 86400
  DNS server address: 4000::1
  Conflict-address expire-time: 172800
  Active normal clients: 0
```

Run the **display dhcpv6 server** command on the switch to check information about the DHCPv6 server.

```
<Switch A> display  dhcpv6  server
  Interface                                DHCPv6 pool
  Vlanif100                                pool1
```

----End

Configuration File

Configuration file of Switch A

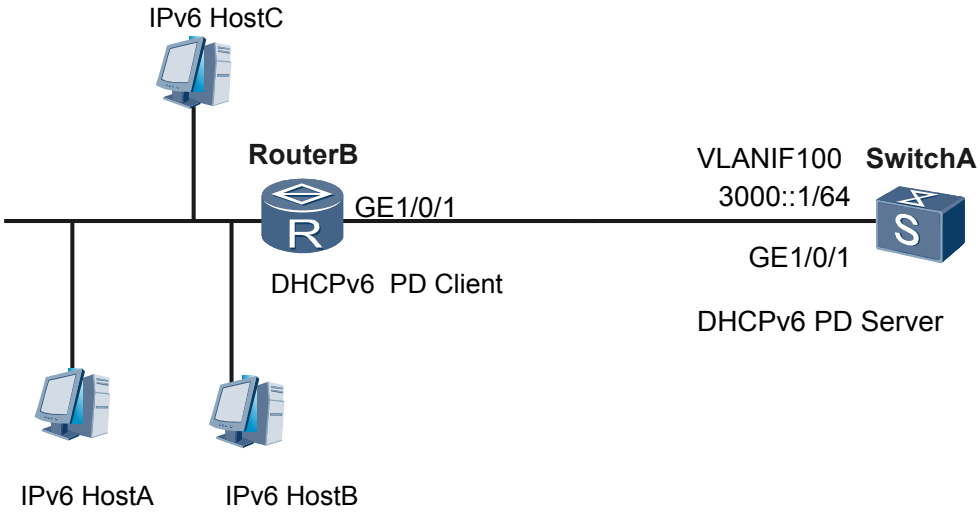
```
#
 sysname Switch A
#
 ipv6
#
 vlan batch 100
#
 dhcp enable
#
 dhcpv6 pool pool1
  address prefix 3000::2/64
  dns-server 4000::1
#
 interface GigabitEthernet1/0/1
  port link-type access
  port default vlan
100
#
 interface Vlanif100
  ipv6 enable
  ipv6 address 3000::1/64
  dhcpv6 server pool1
#
 return
```

6.8.2 Example for Configuring a DHCPv6 PD Server

Networking Requirements

As shown in **Figure 6-5**, RouterB and SwitchA are directly connected and on the same link. RouterB cannot communicate with other devices because it has no IPv6 address and other network configuration parameters. The Switch A needs to be configured as a DHCPv6 PD server to assign IPv6 addresses and other network configuration parameters to DHCPv6 clients. This facilitates centralized management and layered IPv6 network deployment.

Figure 6-5 Networking diagram of configuring the DHCPv6 PD server



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IPv6 on interfaces so that devices can communicate using IPv6.
2. Enable the DHCPv6 PD server function so that DHCPv6 PD server can assign IPv6 addresses using DHCPv6.

Procedure

Step 1 Enable the DHCP service

```
<Quidway> system-view
[Quidway] sysname Switch A
[Switch A] dhcp enable
```

Step 2 Configure IPv6 functions on interfaces

```
[Switch A] ipv6
[Switch A] vlan 100
[Switch A-vlan100] quit
[Switch A] interface gigabitethernet 1/0/1
[Switch A-GigabitEthernet1/0/1]  port link-type access
[Switch A-GigabitEthernet1/0/1]  port default  vlan  100
[Switch A-GigabitEthernet1/0/1] quit
[Switch A] interface vlanif 100
[Switch A-Vlanif100] ipv6 enable
[Switch A-Vlanif100] ipv6 address 3000::1/64
[Switch A-Vlanif100] quit
```

Step 3 Configure a DHCPv6 PD server

```
[Switch A] dhcpv6 pool pool1
[Switch A-dhcpv6-pool-pool1] prefix-delegation 3000::/60 64
[Switch A-dhcpv6-pool-pool1] dns-server 4000::1
[Switch A-dhcpv6-pool-pool1] quit
```

Step 4 Enable the DHCPv6 PD server function on an interface

Enable the DHCPv6 PD server function on VLANIF 100.

```
[Switch A] interface vlanif 100
[Switch A-Vlanif100] dhcpv6 server pool1
```

Step 5 Verify the configuration

Run the **display dhcpv6 pool** command on the switch to check information about the DHCPv6 address pool.

```
<Switch A> display dhcpv6  pool
 DHCPv6 pool: pool1
  Prefix delegation: 3000::/60 64
    lifetime valid 172800 seconds, preferred 86400 seconds
    0 in use
  Information refresh time: 86400
  DNS server address: 4000::1
  Conflict-address expire-time: 172800
  Active pd clients: 0
```

Run the **display dhcpv6 server** command on the switch to check information about the DHCPv6 PD server.

```
<Switch A> display  dhcpv6  server
  Interface                                DHCPv6 pool
 Vlanif100                                  pool1
```

----End

Configuration File

Configuration file of SwitchA

```
#
 sysname Switch A
#
 ipv6
#
 vlan batch 100
#
 dhcp enable
#
 dhcpv6 pool pool1
  prefix-delegation 3000::/60 64
  dns-server 4000::1
#
 interface GigabitEthernet1/0/1
  port link-type access
  port default vlan
100
#
interface Vlanif100
  ipv6 enable
  ipv6 address 3000::1/64
  dhcpv6 server pool1
#
return
```

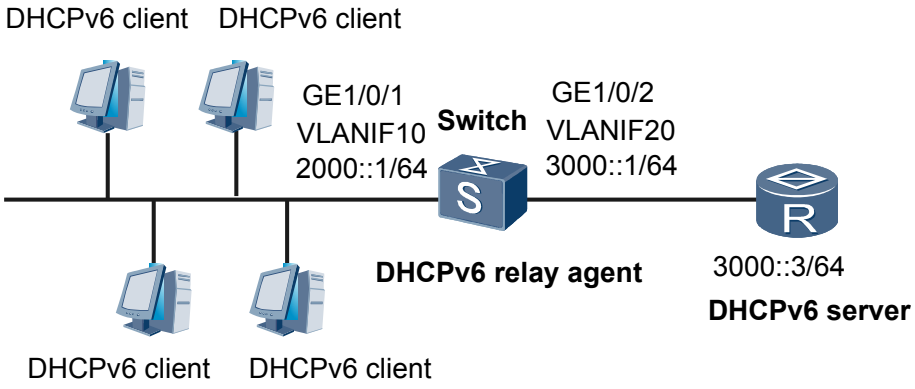
6.8.3 Example for Configuring a DHCPv6 Relay Agent

Networking Requirements

As shown in **Figure 6-6**, the DHCPv6 client address is 2000::/64 and the DHCPv6 server address is 3000::3/64. The DHCPv6 client and server are on different links; therefore, a DHCPv6 relay agent is required to forward DHCPv6 packets.

The Switch needs to function as the DHCPv6 relay agent to forward DHCPv6 packets between the DHCPv6 client and server. In addition, the Switch functions as the gateway device of the network at 2000::/64. The M flag bit and O flag bit in RA messages allow hosts on the network to obtain IPv6 addresses and other network configuration parameters through DHCPv6.

Figure 6-6 Networking diagram of configuring a DHCPv6 relay agent



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IPv6 on interfaces so that devices can communicate using IPv6.
2. Enable the DHCPv6 relay function so that the DHCPv6 server and client on different links can transmit packets.

Procedure

Step 1 Enable the DHCPv6 service

```
<Quidway> system-view
[Quidway] dhcp enable
```

Step 2 Adding interfaces to VLANs

Add GigabitEthernet1/0/1 to VLAN 10.

```
[Quidway] vlan batch 10 20
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port hybrid pvid vlan 10
[Quidway-GigabitEthernet1/0/1] port hybrid untagged vlan 10
[Quidway-GigabitEthernet1/0/1] quit
```

Add GigabitEthernet1/0/2 to VLAN 20.

```
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port hybrid pvid vlan 20
[Quidway-GigabitEthernet1/0/2] port hybrid untagged vlan 20
[Quidway-GigabitEthernet1/0/2] quit
```

Step 3 Assign IPv6 addresses to VLANIF interfaces

Enable the IPv6 packet forwarding function.

```
[Quidway] ipv6
```

Assign an IPv6 address to VLANIF 10.

```
[Quidway] interface vlanif 10
[Quidway-Vlanif10] ipv6 enable
[Quidway-Vlanif10] ipv6 address 2000::1 64
[Quidway-Vlanif10] quit
```

Assign an IPv6 address to VLANIF 20.

```
[Quidway] interface vlanif 20
[Quidway-Vlanif20] ipv6 enable
[Quidway-Vlanif20] ipv6 address 3000::1 64
[Quidway-Vlanif20] quit
```

Step 4 Enable the DHCPv6 relay function

Enable the DHCPv6 relay function on VLANIF 10 and specify the IPv6 address of the DHCPv6 server.

```
[Quidway] interface vlanif 10
[Quidway-Vlanif10] dhcpv6 relay destination 3000::3
```

Step 5 Configure the Switch as the gateway

Configure the Switch to send RA messages and configure M and O flag bits.

```
[Quidway-Vlanif10] undo ipv6 nd ra halt
[Quidway-Vlanif10] ipv6 nd autoconfig managed-address-flag
[Quidway-Vlanif10] ipv6 nd autoconfig other-flag
[Quidway-Vlanif10] quit
```

Step 6 Verify the configuration

Run the **display dhcpv6 relay** command on the Switch, and you can view the DHCPv6 relay configuration.

```
[Quidway] display dhcpv6 relay
 Interface    Mode    Destination
------------------------------------------------------------------
Vlanif10     Relay   3000::3
------------------------------------------------------------------
```

Run the **display dhcpv6 relay statistics** command on the Switch, and you can view statistics about DHCPv6 packets passing through the DHCPv6 relay agent.

```
[Quidway] display dhcpv6 relay statistics
MessageType              Receive          Send             Error
 Solicit                  0                0                0
 Advertise                0                0                0
 Request                  0                0                0
 Confirm                  0                0                0
 Renew                    0                0                0
 Rebind                   0                0                0
 Reply                    0                0                0
 Release                  0                0                0
 Decline                  0                0                0
 Reconfigure              0                0                0
 Information-request      0                0                0
 Relay-forward            0                0                0
 Relay-reply              0                0                0
 UnknownType              0                0                0
```

----End

Configuration File

Configuration file of the Switch

```
#
 sysname Quidway
#
 vlan batch 10 20
#
 ipv6
#
dhcp enable
#
interface Vlanif10
 ipv6 enable
 ipv6 address 2000::1/64
 undo ipv6 nd ra halt
 ipv6 nd autoconfig managed-address-flag
 ipv6 nd autoconfig other-flag
 dhcpv6 relay destination 3000::3
#
interface Vlanif20
 ipv6 enable
 ipv6 address 3000::1/64
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 10
 port hybrid untagged vlan 10
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 20
 port hybrid untagged vlan 20
#
return
```

7 IP Performance Configuration

About This Chapter

You can optimize IP performance by adjusting parameters on the network.

7.1 IP Performance Overview
Parameters on certain networks need to be modified to optimize network performance.

7.2 Default Configuration
This section provides the default IP performance configuration.

7.3 Optimizing IP Performance
This section describes how to optimize IP performance. You can set IP performance parameters to achieve best network performance.

7.4 Maintaining IP Performance
This section describes how to clear IP performance statistics to maintain IP performance.

7.5 Configuration Examples
This section provides IP performance configurations including the networking requirements, networking diagram, configuration roadmap, and configuration procedures.

7.1 IP Performance Overview

Parameters on certain networks need to be modified to optimize network performance.

A large number of packets need to be forwarded on the network, which may cause network congestion and degrade network performance.IP performance optimization can solve the problem. You can adjust parameters or forwarding modes for IP packets to achieve optimal network performance.

7.2 Default Configuration

This section provides the default IP performance configuration.

Table 7-1 describes the default configuration of IP performance.

Table 7-1 Default IP performance configuration

Parameter	Default Configuration
Source IP address verification	Disabled
IP packet fragmentation on outbound interface	Disabled
Fast ICMP reply function	Disabled
Discarding ICMP packets whose TTL values are 1 on an LPU	Disabled
Discarding ICMP packets that carry options on an LPU	Disabled
Discarding ICMP destination unreachable packets	Disabled
Sending ICMP port unreachable packets	Enabled
Sending ICMP host unreachable packets	Enabled
Sending ICMP redirection packets	Enabled
Not sending ICMP unreachable packets	Disabled
TCP SYN-Wait timer	75s
TCP FIN-Wait timer	675s
TCP window size	8k bytes

7.3 Optimizing IP Performance

This section describes how to optimize IP performance. You can set IP performance parameters to achieve best network performance.

Prerequisite

Before optimizing IP performance, complete the following task:

- Configuring IP addresses for interfaces

7.3.1 Configuring Source IP Addresses Verification

Context

Configuring source IP address verification enables an interface to check validity of source IP addresses of received packets. Packets with invalid addresses are discarded. The interface only check validity of source IP addresses of the packets that are forwarded to the CPU and does not check validity of source IP addresses of the packets that will be directly forwarded according to the FIB table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed. The interface can be a VLANIF, Loopback, POS, GE sub-interface, 40GE sub-interface or tunnel interface.

NOTE

If the interface is a VLANIF interface, a VLAN must be created.

Step 3 Run:

```
ip verify source-address
```

Source IP address verification is configured.

The device only checks validity of source IP addresses of packets forwarded from an interface to the CPU.

----End

7.3.2 Configuring an Outbound Interface to Fragment IP Packets

Context

If the size of IP packets exceeds the MTU, oversized packets will be discarded. After IP packet fragmentation is enabled, the system sets the DF field of an IP packet to 0 and fragments the IP packet to ensure that all packets are forwarded.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface interface-type interface-number
```

The interface view is displayed. The interface can be a VLANIF, Loopback, or tunnel interface.

NOTE

If the interface is a VLANIF interface, a VLAN must be created.

Step 3 Run:

```
clear ip df
```

The function that clears the DF field is configured to enable IP packet fragmentation on an outbound interface.

By default, an outbound interface does not fragment IP packets.

NOTE

This command takes effect only for the packets that are sent to the CPU for software forwarding but not for the packets that are forwarded by the chip.

----End

7.3.3 Configuring a Load Balancing Mode for IP Packet Forwarding

Context

If flow-based load balancing is used, the hash algorithm is used to calculate a value based on the protocol type, source IP address and mask, destination IP address and mask, source port number, and destination port number. A link is selected based on the value. Packets then are forwarded over this link. For details, see (Optional) Configuring Load Balancing Modes.

By default, flow-based load balancing is used.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
load-balance { flow | packet } [ all | slot slot-id ]
```

A load balancing mode is configured for IP packet forwarding.

----End

7.3.4 Controlling IP packets with Route-alert Options

Context

By controlling IP packets with route-alert options, the device can reduces pressure on CPU and improves network performance and security.

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:

```
discard ra
```

The interface is configured to discard IP packets with route-alert options.

By default, packets with route-alert options are processed by the CPU.

----End

7.3.5 Controlling IP packets with Record-route Options

Context

By controlling IP packets with record-route options, the device can reduces pressure on CPU and improves network performance and security.

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:

discard rr

The interface is configured to discard IP packets with record-route options.

By default, packets with record-route options are processed by the CPU.

----End

7.3.6 Controlling IP packets with Time-stamp Options

Context

By controlling IP packets with time-stamp options, the device can reduces pressure on CPU and improves network performance and security.

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:

discard ts

The interface is configured to discard IP packets with time-stamp options.

By default, packets with time-stamp options are processed by the CPU.

----End

7.3.7 Configuring ICMP properties

Context

CAUTION

- If the function of sending ICMP redirection packets is disabled, the device does not send ICMP redirection packets in any situations.
- If the function of sending ICMP host unreachable packets is disabled, the device does not send ICMP host unreachable packets

Optimizing ICMP properties can reduce ICMP packets on the network and reduce the burden on the peer device.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
icmp-reply fast
```

The fast ICMP reply function is enabled.

By default, the fast ICMP reply function is disabled on the device.

NOTE

After the fast ICMP reply function is enabled on switch, switch respond to ICMP Echo packets quickly in any of the following situations:

- switch do not have the ARP entry of the device that initiates the ping and cannot learn the ARP entry of the device.
- switch do not have route entries to the device that initiates the ping.
- switch receive ICMP Echo packets with incorrect checksum.

Step 3 Run:

```
icmp ttl-exceeded drop { slot slot-id | all }
```

The LPU is configured to discard the ICMP packets whose TTL values are 1.

By default, the function of discarding ICMP packets with TTL values 1 is disabled.

Step 4 Run:

```
icmp with-options drop { slot slot-id | all }
```

The LPU is configured to discard the ICMP packets that carry options.

By default, the function of discarding ICMP packets that carry options is disabled.

Step 5 Run:

```
icmp unreachable drop
```

The function of discarding ICMP destination unreachable packets is enabled.

By default, the function of discarding ICMP destination unreachable packets is disabled.

Step 6 Run:

```
undo icmp broadcast-address echo enable
```

The function of receiving ICMP Echo broadcast messages is disabled.

By default, the function of receiving ICMP Echo broadcast messages is enabled.

Step 7 Run:

```
icmp port-unreachable send
```

The function of sending ICMP port unreachable packets is enabled.

The function of sending ICMP port unreachable packets is enabled.

Step 8 Run:

```
icmp host-unreachable send
```

The function of sending ICMP host unreachable packets is enabled.

By default, the function of sending ICMP host unreachable packets is enabled.

NOTE

icmp host-unreachable send command is used in the system view and the **icmp host-unreachable send** command is also used in the interface view:

- When the function of sending ICMP host unreachable packets is disabled in the system view, all interfaces do not send ICMP host unreachable packets. Even if the function is enabled on an interface, the interface does not send ICMP host unreachable packets.
- When the function of sending ICMP host unreachable packets is enabled in the system view, all interfaces send ICMP host unreachable packets because the function is enabled on all interfaces by default. You can run the **undo icmp host-unreachable send** command in the interface view to disable the function of sending ICMP host unreachable packets on a specified interface.

Step 9 Run:

```
icmp protocol-unreachable send
```

The function of sending ICMP unreachable packets is enabled.

By default, the function of sending ICMP unreachable packets is enabled.

Step 10 Run:

```
interface vlanif vlan-id
```

The VLANIF interface view is displayed.

Step 11 Run:

```
icmp redirect send
```

The function of sending ICMP redirection packets is enabled.

By default, the function of sending ICMP redirection packets is enabled..

NOTE

When the default CPCAR value is used, the 128-byte ICMP packet sent by the switch every 6 ms is not discarded.

Step 12 Run:

```
icmp blackhole unreachable send
```

The BRAS is disabled from sending a Destination Unreachable ICMP packet to an initiator when a tracert packet matches an IPv4 blackhole route.

By default, the BRAS is disabled from sending a Destination Unreachable ICMP packet to an initiator when a tracert packet matches an IPv4 blackhole route.

----End

7.3.8 Configuring TCP Properties

Context

When a TCP connection is set up between switch and other devices, TCP properties such as TCP connection for BGP need to be configured.

The following TCP properties can be configured on switch:

- SYN-Wait timer: When SYN packets are sent, the SYN-Wait timer is started. If no response packet is received after the SYN-Wait timer expires, the TCP connection is closed.
- FIN-Wait timer: When the TCP connection status changes from FIN_WAIT_1 to FIN_WAIT_2, the FIN-Wait timer is started. If no response packet is received after the FIN-Wait timer expires, the TCP connection is closed.
- Receive/send buffer size of connection-oriented socket *window size*.

If you configure TCP properties in the system view for multiple times, only the last configuration takes effect.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
tcp timer syn-timeout interval
```

The SYN-Wait timer of TCP connections is configured.

The value of the TCP SYN-Wait timer is an integer that ranges from 2 to 600, in seconds. The default value is 75.

Step 3 Run:

```
tcp timer fin-timeout interval
```

The FIN-WAIT timer of TCP connections is configured.

The value of the TCP FIN-Wait timer is an integer that ranges from 76 to 3600, in seconds. The default value is 675.

Step 4 Run:

```
tcp window window-size
```

The socket receive/send buffer size is configured.

The value of *window-size* ranges from 1k bytes to 32k bytes. The default value is 8k bytes.

Step 5 Run:

```
tcp min-mss mss-value
```

The minimum MSS value is set for a TCP connection.

The default minimum MSS value for a TCP connection is 216 bytes.

----End

7.3.9 Checking the Configuration

Procedure

- Run the **display tcp status** [[**task-id** *task-id*] [**socket-id** *socket-id*] | [**local-ip** *ip-address*] [**local-port** *local-port-number*] [**remote-ip** *ip-address*] [**remote-port** *remote-port-number*]] command to check the TCP connection status.

- Run the **display tcp statistics** command to view the TCP traffic statistics.
- Run the **display udp statistics** command to view the UDP traffic statistics.
- Run the **display ip statistics** command to view the IP traffic statistics.
- Run the **display ip socket** [**monitor**] [**task-id** *task-id* **socket-id** *socket-id* | **sock-type** *socket-type*] command to view information about the created IPv4 socket.
- Run the **display icmp statistics** command to view the ICMP traffic statistics.

----End

7.4 Maintaining IP Performance

This section describes how to clear IP performance statistics to maintain IP performance.

7.4.1 Clearing IP Performance Statistics

Context

CAUTION

The IP/TCP/UDP traffic statistics cannot be restored after being cleared. Therefore, confirm your operation before clearing the IP performance statistics.

Procedure

- Run the **reset ip statistics** [**interface** *interface-type interface-number*] command in the user view to clear IP statistics.
- Run the **reset ip socket monitor** [**task-id** *task-id* **socket-id** *socket-id*] command in the user view to clear information in a socket monitor.
- Run the **reset tcp statistics** command in the user view to clear TCP statistics.
- Run the **reset udp statistics** command in the user view to clear UDP statistics.

----End

7.5 Configuration Examples

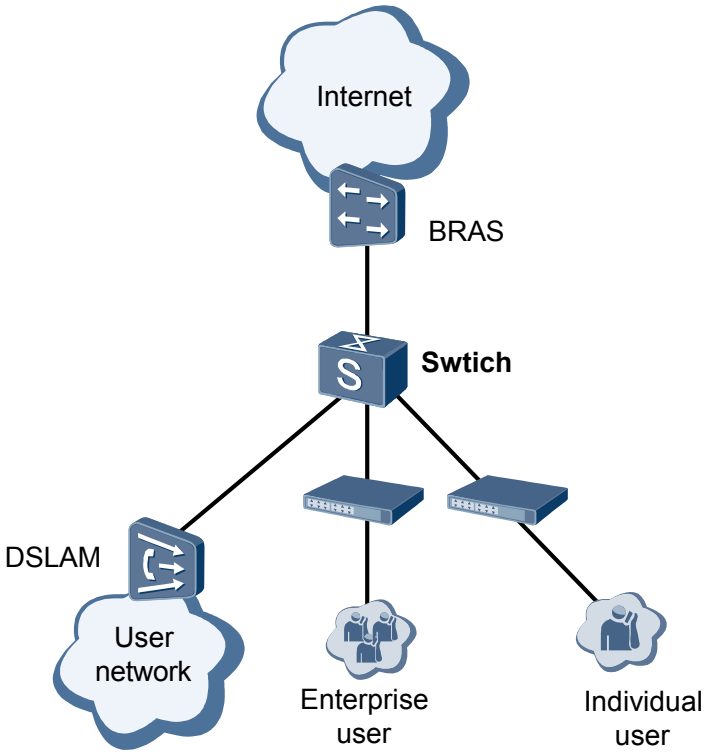
This section provides IP performance configurations including the networking requirements, networking diagram, configuration roadmap, and configuration procedures.

7.5.1 Example for Optimizing System Performance by Discarding Certain ICMP Packets

Networking Requirements

The switch in **Figure 7-1** functions as the aggregation device. Enterprise users, individual users, and DSLAMs are attached to the switch and the switch is connected to the Internet through a BRAS. When a large amount of information is exchanged on the network or the network is attacked, lots of ICMP packets are forwarded and the network performance is degraded. In this case, some ICMP packets are required to be discarded to reduce the burden on the switch.

Figure 7-1 Networking diagram for configuring ICMP security function



Configuration Roadmap

The configuration roadmap is as follows:

Configure the function of discarding ICMP packets whose TTL value is 1, ICMP packets that carry options, and ICMP destination unreachable packets to reduce the burden of the device in processing a large number of ICMP packets.

Procedure

Step 1 Configure the device to discard certain ICMP packets.

Configure the device to discard ICMP packets whose TTL value is 1.

```
<Quidway> system-view
[Quidway] icmp ttl-exceeded drop all
```

Configure the device to discard ICMP packets that carry options.

```
[Quidway] icmp with-options drop all
```

Configure the device to discard ICMP packets whose destination addresses are unreachable.

```
[Quidway] icmp unreachable drop
```

Step 2 Verify the configuration.

Run the **display this** command in the system view to view the ICMP security configurations.

```
[Quidway] display this
#
 icmp unreachable drop
 icmp ttl-exceeded drop slot 1
 icmp with-options drop slot 1
 icmp ttl-exceeded drop slot 2
 icmp with-options drop slot 2
 icmp ttl-exceeded drop slot 3
 icmp with-options drop slot 3
```

----End

Configuration Files

Configuration file of the switch

```
#
 sysname Quidway
#
 icmp unreachable drop
 icmp ttl-exceeded drop slot 1
 icmp with-options drop slot 1
 icmp ttl-exceeded drop slot 2
 icmp with-options drop slot 2
 icmp ttl-exceeded drop slot 3
 icmp with-options drop slot 3
#
return
```

8 DNS Configuration

About This Chapter

This chapter describes the principles, basic functions and configuration procedures of DNS on the switch, and provides configuration examples.

8.1 DNS Overview

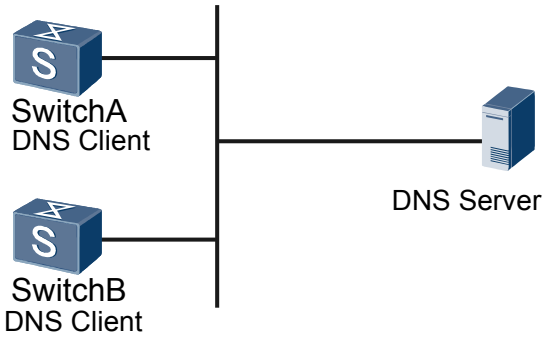
Domain Name System (DNS) is a distributed database used in TCP and IP applications and completes resolution between IP addresses and domain names.

Each host on the network is identified by an IP address. To access a host, a user must obtain the host IP address first. It is difficult for users to remember IP addresses of hosts. Therefore, host names in the format of strings are designed. Each host name maps an IP address. In this way, users can use the simple and meaningful domain names instead of the complicated IP addresses to access hosts.

8.2 DNS Features Supported by the device

The device can function as a DNS client.

Figure 8-1 Functioning as a DNS client



As shown in **Figure 8-1**, the device functions as a DNS client and supports static and dynamic domain name resolution.

- Static domain name resolution: Mappings between domain names and IP addresses are configured manually. To obtain the IP address by resolving a domain name, the client searches the static domain name resolution table for the specified domain name.
- Dynamic domain name resolution: Dynamic DNS resolution is implemented by a DNS server. The DNS server receives domain name resolution requests from DNS clients. The DNS server searches for the corresponding IP address of the domain name in its DNS database. If no matching entry is found, it sends a query message to a higher-level DNS server. This process continues until the DNS server finds the corresponding IP address or detects that the IP address mapping the domain name does not exist. Then the DNS server returns a result to the DNS client.

8.3 Configuring the DNS Client

This section describes how to configure the switch as a DNS client to allow users to use domain names to access other devices.

Pre-configuration Tasks

Before configuring a DNS client, complete the following tasks:

- Configuring link layer protocol parameters for interfaces to ensure that the link layer protocol status on the interfaces is Up
- Configuring a route between the switch and the DNS server

8.3.1 Configuring the Static DNS

Context

A static domain name resolution table is manually set up, describing the mappings between domain names and IP addresses. Some common domain names are added to the table. Static domain name resolution can be performed based on the static domain name resolution table. To obtain the IP address by resolving a domain name, the client searches the static domain name resolution table for the specified domain name. In this manner, the efficiency of domain name resolution is improved.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip host host-name ip-address
```

Static DNS entries are configured.

By default, no static DNS entries are configured.

----End

Follow-up Procedure

Each host name can be mapped to only one IP address. When multiple IP addresses are mapped to a host name, only the latest configuration takes effect. If multiple host names need to be resolved, repeat step 2.

You can configure a maximum of 50 static DNS entries.

8.3.2 Configuring the Dynamic DNS

Context

To implement dynamic DNS, you need to enable dynamic DNS resolution, configure a DNS server, and configure a source IP address for the local device and a domain name suffix. If the local device uses an IP address allocated by the DHCP server and the information delivered by the DHCP server to the local device contains the DNS server IP address and the domain name suffix list, you only need to enable dynamic DNS resolution.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dns resolve
```

Dynamic domain name resolution is enabled.

By default, dynamic DNS resolution is disabled.

Step 3 Run:

```
dns server ip-address
```

The IP address of the DNS server is configured.

By default, no IP address of the DNS server is configured. A maximum of six DNS server IP addresses can be configured on the device.

Step 4 (Optional) Run:

```
dns server source-ip ip-address
```

The source IP address is configured for the local device to function as the DNS client to send and receive DNS packets.

The local device uses the specified IP address to communicate with the DNS server. This ensures communication security.

Step 5 (Optional) Run:

```
dns domain domain-name
```

A domain name suffix is configured.

By default, no domain name suffix is configured on a DNS client.

----End

Follow-up Procedure

The system supports a maximum of six DNS servers, one specified source address, and ten domain name suffixes. If multiple DNS servers are required, repeat step 3. If multiple domain name suffixes are required, repeat step 5.

8.3.3 Checking the Configuration

Procedure

- Run the **display ip host** command to check static DNS entries.
- Run the **display dns server** command to check the DNS server configuration.
- Run the **display dns domain** command to check the domain name suffix configuration.

----End

8.4 Maintaining DNS

Maintaining DNS includes clearing dynamic DNS entries and monitoring DNS running status.

8.4.1 Deleting Dynamic DNS Entries

Context

CAUTION

Dynamic DNS entries cannot be restored after being deleted. Exercise caution when you run the command.

Procedure

- Run the **reset dns dynamic-host** command to delete dynamic DNS entries.

----End

8.4.2 Monitoring the Running Status of DNS

Context

In routine maintenance, you can run the following command in any view to check the running status of DNS.

Procedure

- Run the **display dns dynamic-host** command to display dynamic DNS entries.

----End

8.5 Configuration Examples

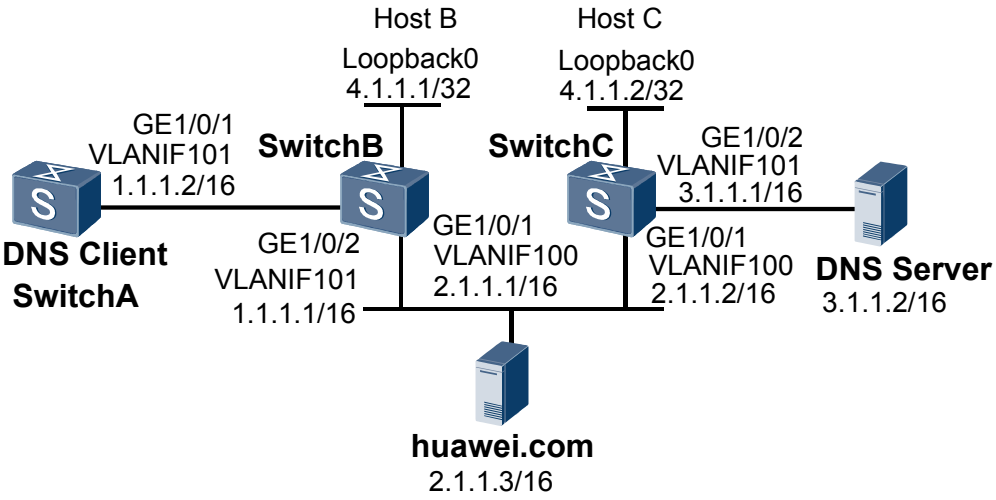
This section provides DNS configuration examples, including networking requirements, configuration roadmap, and configuration procedure.

8.5.1 Example for Configuring the DNS Client

Networking Requirements

Compared with an IP address, the URL is easy to remember. Users want to access network servers using domain names. It is required that the DNS server can resolve a domain name after a user enters some fields of the domain name. For example, when a user attempts to access the host **huawei.com**, the user only needs to enter **huawei**. It is required that the DNS server can fast resolve common domain names.

Figure 8-2 Networking diagram for configuring the DNS client



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure static DNS entries on Switch A to access HostB and HostC.
2. Configure the dynamic DNS resolution on SwitchA to access the network server.
3. Configure the domain name suffix on SwitchA to support a domain name suffix list.
4. Configure OSPF on switches to ensure routes among all devices are reachable.

Procedure

Step 1 Configure SwitchA.

Configure an IP address for VLANIF101.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] vlan 101
[SwitchA-vlan101] quit
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port link-type access
[SwitchA-GigabitEthernet1/0/1] port default vlan 101
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 101
[SwitchA-Vlanif101] ip address 1.1.1.2 255.255.0.0
[SwitchA-Vlanif101] quit
```

Configure OSPF.

```
[SwitchA] ospf
[SwitchA-ospf-1] area 0
[SwitchA-ospf-1-area-0.0.0.0] network 1.1.0.0 0.0.255.255
[SwitchA-ospf-1-area-0.0.0.0] quit
[SwitchA-ospf-1] quit
```

Configure static DNS entries.

```
[SwitchA] ip host hostB 4.1.1.1
[SwitchA] ip host hostC 4.1.1.2
```

Enable DNS resolution.

```
[SwitchA] dns resolve
```

Configure an IP address for the DNS server.

```
[SwitchA] dns server 3.1.1.2
```

Set the domain name suffix to ".net".

```
[SwitchA] dns domain net
```

Set the domain name suffix to ".com".

```
[SwitchA] dns domain com
```

NOTE

You need to configure OSPF on SwitchB and SwitchC to ensure reachable routes between them. For details about OSPF configurations on SwitchB and SwitchC, see the configuration files.

Step 2 Verify the configuration.

Run the **ping hostB** command on SwitchA. You can see that the ping operation succeeds and the destination IP address is 4.1.1.1.

```
<SwitchA>  ping hostB
 PING hostB (4.1.1.1): 56  data bytes, press CTRL_C to break
    Reply from 4.1.1.1: bytes=56 Sequence=1 ttl=126 time=4 ms
    Reply from 4.1.1.1: bytes=56 Sequence=2 ttl=126 time=1 ms
    Reply from 4.1.1.1: bytes=56 Sequence=3 ttl=126 time=1 ms
    Reply from 4.1.1.1: bytes=56 Sequence=4 ttl=126 time=1 ms
    Reply from 4.1.1.1: bytes=56 Sequence=5 ttl=126 time=1 ms

  --- hostB ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/4 ms
```

Run the **ping huawei.com** command on SwitchA. You can see that the ping operation succeeds and the destination IP address is 2.1.1.3.

```
<SwitchA> ping huawei.com
  PING huawei.com (2.1.1.3): 56  data bytes, press CTRL_C to break
    Reply from 2.1.1.3: bytes=56 Sequence=1 ttl=126 time=6 ms
    Reply from 2.1.1.3: bytes=56 Sequence=2 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=3 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=4 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=5 ttl=126 time=4 ms

  --- huawei.com ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 4/4/6 ms
```

Run the **ping huawei** command on SwitchA. You can see that the ping operation succeeds, the domain name changes to huawei.com, and the destination IP address is 2.1.1.3.

```
<SwitchA> ping huawei
  PING huawei.com (2.1.1.3): 56  data bytes, press CTRL_C to break
    Reply from 2.1.1.3: bytes=56 Sequence=1 ttl=126 time=6 ms
```

```
    Reply from 2.1.1.3: bytes=56 Sequence=2 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=3 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=4 ttl=126 time=4 ms
    Reply from 2.1.1.3: bytes=56 Sequence=5 ttl=126 time=4 ms

  --- huawei.com ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 4/4/6 ms
```

Run the **display ip host** command on SwitchA. You can view mappings between host names and IP addresses in static DNS entries.

```
<SwitchA> display ip host
Host                 Age        Flags  Address
hostB                0          static 4.1.1.1
hostC                0          static 4.1.1.2
```

Run the **display dns dynamic-host** command on SwitchA. You can view information about dynamic DNS entries saved in the cache.

```
<SwitchA> display dns dynamic-host
No  Domain-name          IpAddress      TTL    Alias
1   huawei.com           2.1.1.3        114
```

----End

Configuration File

Configuration file of SwitchA

```
#
 sysname SwitchA
#
 vlan batch 101
#
 ip host hostB 4.1.1.1
 ip host hostC 4.1.1.2
#
 dns resolve
 dns server 3.1.1.2
 dns domain net
 dns domain com
#
interface Vlanif101
 ip address 1.1.1.2 255.255.0.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 101
#
ospf 1
 area 0.0.0.0
  network 1.1.0.0 0.0.255.255
#
return
```

Configuration file of SwitchB

```
#
 sysname SwitchB
#
 vlan batch 100 101
#
interface LoopBack0
```

```
 ip address 4.1.1.1 255.255.255.255
#
interface Vlanif101
 ip address 1.1.1.1 255.255.0.0
#
interface Vlanif100
 ip address 2.1.1.1 255.255.0.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 100
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 101
#
ospf 1
 area 0.0.0.0
  network 1.1.0.0 0.0.255.255
  network 2.1.0.0 0.0.255.255
  network 4.1.1.1 0.0.0.0
#
return
```

Configuration file of SwitchC

```
#
 sysname SwitchC
#
 vlan batch 100 101
#
interface LoopBack0
 ip address 4.1.1.2 255.255.255.255
#
interface Vlanif101
 ip address 3.1.1.1 255.255.0.0
#
interface Vlanif100
 ip address 2.1.1.2 255.255.0.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 100
#
interface GigabitEthernet1/0/2
 port link-type access
 port default vlan 101
#
ospf 1
 area 0.0.0.0
  network 2.1.0.0 0.0.255.255
  network 3.1.0.0 0.0.255.255
  network 4.1.1.2 0.0.0.0
#
return
```

9 Basic IPv6 Configurations

About This Chapter

The IPv6 protocol stack supports routing protocols and application protocols on an IPv6 network.

9.1 IPv6 Overview
Internet Protocol Version 6 (IPv6), also called IP Next Generation (IPng), is the second-generation Internet Protocol. It is a set of specifications defined by the Internet Engineering Task Force (IETF).

9.2 IPv6 Features Supported by the Device
Basic IPv6 functions include IPv6 address configuration, rate limiting for sending ICMPv6 error packets, IPv6 neighbor discovery (ND), path maximum transmission unit (PMTU) discovery, and Transmission Control Protocol for IPv6 (TCP6).

9.3 Configuration Notes
This section describes notes about configuring IPv6.

9.4 Default Configuration
This section describes the default IPv6 configuration.

9.5 Configuring IPv6 Addresses for Interfaces
To enable network devices to communicate at the network layer, configure interface IPv6 addresses on the network devices.

9.6 Setting Rate Limit for Sending ICMPv6 Error Packets
Rate limiting for sending ICMPv6 error packets reduces network traffic and prevents malicious attacks.

9.7 Configuring IPv6 Neighbor Discovery
The Neighbor Discovery Protocol (NDP) is a basic IPv6 protocol. It replaces the Address Resolution Protocol (ARP) and ICMP Router Discovery on an IPv4 network. Additionally, IPv6 ND provides redirection and neighbor unreachability detection.

9.8 Configuring PMTU
The PMTU is the minimum MTU of the path from the source to the destination. Packets that are sent according to this MTU do not need to be fragmented during transmission. This reduces loads on routing devices and optimizes network resource efficiency to obtain the maximum throughput.

9.1 IPv6 Overview

Internet Protocol Version 6 (IPv6), also called IP Next Generation (IPng), is the second-generation Internet Protocol. It is a set of specifications defined by the Internet Engineering Task Force (IETF).

IPv4 is the widely used Internet protocol. During initial development of the Internet, IPv4 rapidly developed because of its simplicity, ease of implementation, and good interoperability. However, as the Internet rapidly develops, deficiency in IPv4 design becomes obvious:

- IP address shortage: IPv4 addresses are being exhausted. There are several solutions to IPv4 address exhaustion. Classless Inter-domain Routing (CIDR) and Network Address Translator (NAT) are two such solutions. CIDR and NAT, however, have their disadvantages and unsolvable problems.
- Large routing table for the backbone router: Many non-contiguous IPv4 addresses are allocated in the early IPv4 technology development, so routes cannot be summarized effectively due to incorrect IPv4 address allocation and planning. The increasingly large routing table consumes a lot of memory and affects forwarding efficiency. Device manufacturers have to keep upgrading routers to improve route addressing and forwarding performance, increasing the router cost.
- Difficulty in address auto-configuration and readdressing: The IPv4 address space is insufficient and IP addresses are allocated unevenly. IP addresses often need to be reallocated during network expansion or replanning. Address autoconfiguration and readdressing are required to simplify address maintenance.
- Lack of end-to-end security: Security is not fully considered in the design of IPv4. Therefore, the original IPv4 framework does not support end-to-end security.
- No QoS guarantee: IPv4 has no native mechanism to support QoS.

To overcome the deficiency in IPv4, IPv6 emerges. IPv6 has the following advantages over IPv4:

- 128-bit address: A 128-bit address space allows for 2^{128} (4.3 billion x 4.3 billion x 4.3 billion x 4.3 billion) possible addresses. The biggest advantage of IPv6 is its almost infinite address space.
- Hierarchical network structure: A huge address space allows for the hierarchical network design in IPv6. The hierarchical network design facilitates route summarization and improves forwarding efficiency.
- Stateless autoconfiguration: IPv6 addresses can be automatically configured, realizing plug and play of network devices.
- Flexible and simple IPv6 packet header: IPv6 uses fixed and extension headers, which greatly improves packet processing efficiency.
- End-to-end security: IPv6 supports IP Security (IPSec) authentication and encryption at the network layer, so it provides end-to-end security.
- QoS: IPv6 has the Flow Label field, which guarantees QoS for voice, data, and video services.
- Mobility: The mobile node and peer node can communicate using the routing header and destination options header.

9.2 IPv6 Features Supported by the Device

Basic IPv6 functions include IPv6 address configuration, rate limiting for sending ICMPv6 error packets, IPv6 neighbor discovery (ND), path maximum transmission unit (PMTU) discovery, and Transmission Control Protocol for IPv6 (TCP6).

IPv6 Address

IPv6 addresses are classified into unicast, anycast, and multicast addresses. Compared to IPv4, IPv6 has no broadcast address, uses multicast addresses as broadcast addresses, and introduces a new address type anycast address.

An IPv6 address is 128 bits long. It has two parts: network prefix and interface identifier. An IPv6 network prefix corresponds to the network ID of an IPv4 address, and an interface identifier corresponds to the host ID of an IPv4 address.

Table 9-1 IPv6 address types and applications

Address Type		IPv6 Prefix	Application
Unicast address	Unspecified address	::/128	An IPv6 unspecified address is used by an interface that has not been assigned an IPv6 address as a temporary source address to communicate with other devices. Packets that carry this address are not forwarded.
	Loopback address	::1/128	Similar to IPv4 loopback address 127.0.0.1, an IPv6 loopback address is used when a node needs to send IPv6 packets to itself.
	Link-local address	FE80::/10	Link-local addresses are only used in communication between nodes on the same local link. Devices do not forward packets with link-local address as source or destination address. A device can automatically generate a link-local address that can be used in neighbor discovery and stateless address autoconfiguration.
	Unique local address (ULA)	FC00::/7	A unique local address is similar to an IPv4 private address but has a globally unique prefix. Any organization that does not obtain an IPv6 global unicast address can use a unique local address to construct a private network. Devices do not forward packets that carry unique local addresses on the public network.
	Global unicast address	Other addresses	An IPv6 global unicast address is similar to an IPv4 public address. IPv6 global unicast addresses support route prefix summarization, helping limit the number of global routing entries.

Address Type	IPv6 Prefix	Application
Multicast address	FF00::/8	An IPv6 multicast address is similar to an IPv4 multicast address, identifying a group of interfaces. Packets sent to an IPv6 multicast address are delivered to all the interfaces identified by the multicast address.

Rate Limiting for Sending ICMPv6 Error Packets

The Internet Control Message Protocol for IPv6 (ICMPv6) is a basic IPv6 protocol that allows network devices to report IPv6 packet forwarding information and errors to the source node.

If an error occurs on a device during IPv6 packet forwarding (such as unreachable destination or incorrect parameters), the device sends an ICMPv6 error packet to the source node of the packets. When network traffic load is heavy or the network is attacked by forged ICMPv6 packets, the device frequently sends ICMPv6 error packets. This burdens the network and causes high CPU usage. In this situation, you can limit the rate at which ICMPv6 error packets are sent.

NOTE

When the default CPCAR value and default timeout interval are used, ICMPv6 packets will not time out if a device sends 256-byte ICMPv6 packets every 6 ms.

IPv6 Neighbor Discovery

The Neighbor Discovery Protocol is a basic IPv6 protocol. It replaces the Address Resolution Protocol (ARP) and ICMP Router Discovery on an IPv4 network.

NDP provides the following functions:

- Address resolution: An IPv6 node parses an IPv6 unicast or anycast address into a link layer address according to neighbor solicitation (NS) and neighbor advertisement (NA) packets. IPv6 NDP cannot parse multicast addresses.
- Router or prefix discovery and address autoconfiguration: An IPv6 node detects a switch, determines whether other nodes on the local link are reachable, and obtains the network prefix of the local network segment based on the NA packets. The node uses the obtained network prefix and other parameters to automatically configure an IPv6 address.
- Redirection: A switch is required when a source node forwards IPv6 packets to a non-directly connected destination address. If the switch is not the optimal next hop, it sends a Redirect packet to the source node, notifying the source node of a better next-hop switch.
- Duplicate address detection (DAD): An IPv6 node uses NS and NA packets to check whether an IPv6 address is used by other nodes. This function is similar to IPv4 gratuitous ARP.
- Neighbor unreachability detection: After obtaining the link layer address of a neighbor, an IPv6 node detects whether the neighbor is reachable based on NS and NA packets.

NOTE

When the default CPCAR value is used, a switch can learn a maximum of 420 ND entries.

IPv6 PMTU Discovery Protocol

When packets are forwarded from the source node to the destination node, they may pass through links of different MTUs. The MTU of the whole transmission path is determined based on the minimum MTU of each link. The path MTU (PMTU) discovery protocol discovers the minimum MTU on the network path.

To reduce burdens on a transit node, the source node fragments IPv6 packets. When an interface of the transit node receives a packet whose size exceeds the MTU, the transit node discards the packet and sends an ICMPv6 Packet Too Big message to the source host. The ICMPv6 Packet Too Big message contains the MTU value of the outbound interface. The source host then fragments the packet based on the MTU and sends the packet again. This increases traffic volume. The path MTU discovery protocol dynamically discovers the MTU value of each link on the transmission path, reducing excessive traffic costs.

TCP6

On an IPv6 network, you can adjust the TCP6 timer and window size to improve network performance.

Precautions for Configuring ND-CPCAR

The switch has a default CIR value for each type of protocol packet. You can adjust CIR values for specified types of protocol packets based on services and network environment.

NOTE

The CIR values listed in the following tables are for reference only. Adjust CIR values based on services and network environment to prevent high CPU usage.

When switches are connected to a large number of IPv6 hosts, the default CIR value of ND packets cannot meet requirements for protocol packet exchange. As a result, many ND packets may be lost and ND neighbors may be not learned. To solve the problems, adjust an appropriate CIR value to prevent CPU overload. For details, see **Table 9-2**.

Table 9-2 Recommended CIR values for ND packets

Number of NDs in the System	Number of Hosts on Each VLANIF Interface	Recommended CIR (kbit/s)	Maximum CPU Usage
200	25	64	10%
280	35	64	10%
440	55	64	10%
800	100	128	13%
1600	200	192	14%
2400	300	256	17%
4000	500	320	26%
6400	800	832	42%

Number of NDs in the System	Number of Hosts on Each VLANIF Interface	Recommended CIR (kbit/s)	Maximum CPU Usage
9600	1200	1152	53%
12800	1600	1280	62%
16384	2048	1408	83%

9.3 Configuration Notes

This section describes notes about configuring IPv6.

When you configure IPv6 on the switch, note the following:

- The IPv6 function of the switch is controlled by the license. By default, the IPv6 function is disabled on the switch. To use the IPv6 function of the switch, buy the license from the Huawei local office.

9.4 Default Configuration

This section describes the default IPv6 configuration.

Default Configuration

Parameter	Default Configuration
IPv6 packet forwarding	Disabled
Interval for sending RA packets	Maximum interval: 600s; minimum interval: 200s
Neighbor reachable time	30000 ms

9.5 Configuring IPv6 Addresses for Interfaces

To enable network devices to communicate at the network layer, configure interface IPv6 addresses on the network devices.

Pre-configuration Tasks

Before configuring IPv6 addresses for interfaces, complete the following task:

- Configuring link layer protocol parameters for interfaces to ensure that the link layer protocol status on the interfaces is Up

9.5.1 Configuring Global Unicast Addresses for Interfaces

Context

A global unicast address is similar to an IPv4 public address and provided for the Internet Service Provider (ISP). A global unicast address can be generated using either of the following methods:

- Generated in the EUI-64 format: An IPv6 global unicast address in the EUI-64 format contains a manually configured prefix and an automatically generated interface identifier.
- Configured manually: An IPv6 global unicast address can be manually configured.

NOTE

- An interface can be configured with multiple global unicast addresses with different network prefixes.
- Manually configured global unicast addresses have higher priority than automatically generated ones. Manually configured addresses can overwrite automatically generated ones with the same prefix. The overwritten automatically generated addresses do not take effect even if manually configured addresses are deleted. A device needs to generate a new global unicast address based on the IP prefix carried in the received RA packet.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6
```

IPv6 packet forwarding is enabled.

By default, IPv6 packet forwarding is disabled.

Step 3 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Step 4 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

By default, the IPv6 function is disabled on an interface.

Step 5 You can run either of the following commands to configure an IPv6 global unicast address for an interface:

- Run:

  ```
  ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }
  ```

  An IPv6 global unicast address is manually configured.

- Run:

  ```
  ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length } eui-64
  ```

  An IPv6 global unicast address is generated in the EUI-64 format.

A maximum of 10 global unicast addresses can be configured on an interface.

----End

Checking the Configuration

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command in any view to check IPv6 information on an interface.
- Run the **display this ipv6 interface** command in the interface view to check IPv6 information on the interface.

9.5.2 Configuring Link-local Addresses for Interfaces

Context

Link-local addresses are used in neighbor discovery or stateless autoconfiguration. An IPv6 link-local address can be obtained using either of the following methods:

- Automatically generated: A device automatically generates a link-local address for an interface based on the link-local prefix (FE80::/10) and link layer address of the interface.
- Manually configured: You can manually configure an IPv6 link-local address for an interface.

NOTE

- Each interface can be configured with only one link-local address. To prevent link-local address conflict, automatically generated link-local addresses are recommended. After an interface is configured with an IPv6 global unicast address, it automatically generates a link-local address.
- Manually configured link-local addresses have higher priority than automatically generated ones. Manually configured addresses can overwrite automatically generated ones, but automatically generated addresses cannot overwrite manually configured ones. If manually configured addresses are deleted, the overwritten automatically generated ones take effect.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6
```

IPv6 packet forwarding is enabled.

By default, IPv6 packet forwarding is disabled.

Step 3 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Step 4 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

By default, the IPv6 function is disabled on an interface.

Step 5 You can run either of the following commands to configure a link-local address for an interface:

- Run:

  ipv6 address *ipv6-address* **link-local**

  A link-local address is configured for an interface.

- Run:

  ipv6 address auto link-local

  A link-local address is automatically generated.

----End

Checking the Configuration

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command in any view to check IPv6 information on an interface.
- Run the **display this ipv6 interface** command in the interface view to check IPv6 information on the interface.

9.5.3 Configuring Anycast Addresses for Interfaces

Context

IPv6 anycast addresses are allocated from the unicast address space. An anycast address identifies a group of interfaces, which usually belong to different nodes. When using anycast addresses, pay attention to the following points:

- Anycast addresses can only be used as destination addresses.
- Packets addressed to an anycast address are delivered to the nearest interface that is identified by the anycast address, depending on the routing protocols.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

ipv6

IPv6 packet forwarding is enabled on the switch.

By default, IPv6 packet forwarding is disabled.

Step 3 Run:

interface *interface-type interface-number*

The interface view is displayed.

Step 4 Run:

ipv6 enable

The IPv6 function is enabled on the interface.

By default, the IPv6 function is disabled on an interface.

Step 5 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length } anycast
```

An IPv6 anycast address is configured for the interface.

----End

Checking the Configuration

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command in any view to check IPv6 information on an interface.
- Run the **display this ipv6 interface** command in the interface view to check IPv6 information on the interface.

9.6 Setting Rate Limit for Sending ICMPv6 Error Packets

Rate limiting for sending ICMPv6 error packets reduces network traffic and prevents malicious attacks.

Context

If a large number of ICMPv6 error packets are sent on the network in a short period, network congestion may occur. To prevent network congestion, you can limit the maximum number of ICMPv6 error packets sent in a specified period using the token bucket algorithm.

You can set the bucket size and interval for placing tokens into the bucket. The bucket size indicates the maximum number of tokens that a bucket can hold. One token represents an ICMPv6 error packet. When an ICMPv6 error packet is sent, one token is taken out of the token bucket. When there is no token, ICMPv6 error packets cannot be sent until new tokens are placed into the token bucket after the interval.

If transmission of too many ICMPv6 error packets causes network congestion or the network is attacked by forged ICMPv6 error packets, you can disable the system from receiving ICMPv6 error packets, Host Unreachable packets, and Port Unreachable packets.

Pre-configuration Tasks

Before setting rate limit for sending ICMPv6 error packets, complete the following task:

- **9.5 Configuring IPv6 Addresses for Interfaces**

Procedure

- Control ICMPv6 error messages in the system view.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ipv6
     ```
     IPv6 packet forwarding is enabled.

By default, a device is disabled from forwarding IPv6 unicast packets.

3. Run:

   ipv6 icmp-error { **bucket** *bucket-size* | **ratelimit** *interval* } *

   Rate limit for sending ICMPv6 error packets is set.

   By default, a token bucket can hold a maximum of 10 tokens and the interval for placing tokens into the bucket is 100 ms.

   NOTE

   If transmission of too many ICMPv6 error packets causes network congestion or the network is attacked by forged ICMPv6 error packets, you can also run the **undo ipv6 icmp** { *icmpv6-type icmpv6-code* | *icmpv6-name* | **all** } **receive** command to disable the system from receiving ICMPv6 error packets, Host Unreachable packets, and Port Unreachable packets.

4. Run:

   ipv6 icmp too-big-rate-limit

   The device is enabled to reject jumbo ICMPv6 error messages.

   By default, the device is disabled from rejecting jumbo ICMPv6 error messages.

5. Run:

   undo ipv6 icmp { *icmpv6-type icmpv6-code* | *icmpv6-name* | **all** } **receive**

   The system is disabled from receiving ICMPv6 messages.

   By default, the system is enabled to receive ICMPv6 messages.

6. Run:

   undo ipv6 icmp { *icmpv6-type icmpv6-code* | *icmpv6-name* | **all** } **send**

   The system is disabled from sending ICMPv6 messages.

   By default, the system is enabled to send ICMPv6 messages.

7. Run:

   undo ipv6 icmp redirect send

   The system is disabled from sending ICMPv6 redirect messages.

   By default, the system is enabled to send ICMPv6 redirect messages.

8. Run:

   ipv6 icmp blackhole unreachable send

   The BRAS is enabled to send the BRAS a Destination Unreachable ICMP packet to an initiator when a tracert packet matches an IPv6 blackhole route.

   By default, the BRAS is disabled from sending a Destination Unreachable ICMP packet to an initiator when a tracert packet matches an IPv6 blackhole route.

- Control ICMPv6 messages in the interface view.

  1. Run:

     system-view

     The system view is displayed.

  2. Run:

     interface *interface-type interface-number*

The interface view is displayed.

3. Run:
   ipv6 enable

   The IPv6 function is enabled on the interface.

   By default, the IPv6 function is disabled on an interface.
4. Run:
   undo ipv6 icmp host-unreachable send

   The interface is disabled from sending ICMPv6 host-unreachable packets.

   By default, the interface is enabled to send ICMPv6 host-unreachable packets.
5. Run:
   undo ipv6 icmp port-unreachable send

   The interface is disabled from sending ICMPv6 port Unreachable messages.

   By default, the transmission of ICMPv6 Port Unreachable messages configured globally also takes effect on an interface.
6. Run:
   undo ipv6 icmp hop-limit-exceeded send

   The interface is disabled from sending ICMPv6 hop-limit-exceeded messages.

   By default, the transmission of ICMPv6 Hop Limit Exceeded messages configured globally also takes effect on an interface.

----End

Checking the Configuration

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command to check IPv6 information on an interface.
- Run the **display icmpv6 statistics** command to check ICMPv6 traffic statistics.

9.7 Configuring IPv6 Neighbor Discovery

The Neighbor Discovery Protocol (NDP) is a basic IPv6 protocol. It replaces the Address Resolution Protocol (ARP) and ICMP Router Discovery on an IPv4 network. Additionally, IPv6 ND provides redirection and neighbor unreachability detection.

Pre-configuration Tasks

Before configuring IPv6 ND, complete the following task:

- **9.5 Configuring IPv6 Addresses for Interfaces**

9.7.1 Configuring Static Neighbors

Context

To communicate with a destination host, a host needs to obtain the link-layer address of the destination host. The link-layer address of a neighbor node can be obtained using the neighbor

discovery mechanism or by manually configuring static neighbor entries. A device identifies a static neighbor entry based on the IPv6 address of this neighbor and number of the Layer 3 interface connected to this neighbor. To filter invalid packets, you can create static neighbor entries, binding the destination IPv6 addresses of these packets to nonexistent MAC addresses.

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:

```
ipv6 enable
```

The IPv6 function is enabled.

Step 4 Run either of the following commands based on the interface type:

- Run:

  ```
  ipv6 neighbor ipv6-address mac-address
  ```

  Static neighbor entries are configured on common Layer 3 interfaces.

- Run:

  ```
  ipv6 neighbor ipv6-address mac-address vid vlan-id interface-type interface-number
  ```

  Static neighbor entries are configured on VLANIF interfaces.

A maximum of 300 static neighbor entries can be configured on an interface.

----End

9.7.2 Configuring Neighbor Discovery

Context

IPv6 NDP provides the following functions: address resolution, neighbor unreachability detection, DAD, router/prefix discovery, address autoconfiguration, and redirection.

NOTE

After the IPv6 function is enabled on the switch, the switch automatically implements address resolution, DAD, and redirection. Neighbor unreachability detection, router/prefix discovery, and address autoconfiguration need to be manually configured. You can also configure the switch to send RA packets to enable router/prefix discovery and address autoconfiguration, and enable the automatic detection of ND entries to check whether neighbors are reachable.

After the automatic detection of ND entries is enabled on the switch, the switch can send NS packets to check whether neighbors are reachable before aging ND entries. If neighbors are reachable, the switch updates ND entries; otherwise, the switch ages ND entries.

You can enable the switch to send RA packets. After receiving the RA packets, network nodes perform address autoconfiguration and router/prefix discovery based on the prefix and other configuration information in the RA packets.

After the preceding configurations are complete, NDP functions work properly. You can also adjust ND parameters based on service requirements.

Procedure

Step 1 You can run the following commands to enable NDP functions to work properly.

1. Run:
   system-view
   The system view is displayed.
2. Run:
   interface *interface-type interface-number*
   The interface view is displayed.
3. Run:
   ipv6 enable
   The IPv6 function is enabled.
   By default, the IPv6 function is disabled on an interface.
4. Run:
   undo ipv6 nd ra halt
   The device is enabled to send RA packets.
   By default, the device is disabled from sending RA packets.

Step 2 (Optional) After completing the preceding configurations, adjust ND parameters to meet service requirements.

- In the system view, run:
  ipv6 nd hop-limit *limit*
  Hop limit is set.
  By default, the number of hops is limited to 64.
- In the system view, run:
  ipv6 nd stale-timeout *seconds*
  The aging time of ND entries in STALE state is set.
  By default, the aging time of ND entries in STALE state is 1200 seconds.
- In the system view, run:
  ipv6 nd learning strict
  IPv6 neighbor discovery (ND) strict learning is enabled.
  By default, IPv6 ND strict learning is disabled.

Perform the following operations on interfaces.

Run:

interface *interface-type interface-number*

The interface view is displayed.

- Run:

  ipv6 nd stale-timeout *seconds*

  The aging time of ND entries in STALE state is set.

  By default, the aging time of ND entries in STALE state is 1200 seconds.

- Run:

  ipv6 nd learning strict { **force-disable** | **force-enable** | **trust** }

  IPv6 neighbor discovery (ND) strict learning is enabled.

  By default, IPv6 ND strict learning is disabled.

- Run:

  ipv6 nd ra hop-limit *limit*

  Hop limit is set.

  By default, the number of hops is limited to 64.

  NOTE

  - If the **ipv6 nd ra hop-limit** command is executed on an interface, the hop limit for RA packets is determined by the interface configuration.
  - If the **ipv6 nd ra hop-limit** command is not executed on an interface, the hop limit for RA packets is determined by the hop limit configured using the **ipv6 nd hop-limit** command.

- Run:

  ipv6 nd ns retrans-timer *interval*

  The interval for sending NS packets is set.

  By default, the interval for sending NS packets is 1000 ms.

- Run:

  ipv6 nd ra { **max-interval** *maximum-interval* | **min-interval** *minimum-interval* }

  The interval for sending RA packets is set.

  By default, the maximum interval for sending RA packets is 600s and the minimum interval is 200s.

- Run:

  ipv6 nd ra prefix { *ipv6-address prefix-length* | *ipv6-address/prefix-length* } *valid-lifetime preferred-lifetime* [**no-autoconfig**] [**off-link**]

  Prefix information in RA packets is configured.

  By default, an RA packet carries only the address prefix configured using the **ipv6 address** command.

- Run:

  ipv6 nd autoconfig managed-address-flag

  The managed address configuration flag (M flag) for stateful autoconfiguration in RA packets is set.

  By default, the M flag in an RA packet is not set.

- Run:

  ipv6 nd autoconfig other-flag

  The other configuration flag (O flag) for stateful autoconfiguration in RA packets is set.

  By default, the O flag in an RA packet is not set.

NOTE

If the M flag in an RA packet is set to 1, the O flag must be set to 1.

- Run:

  ipv6 nd nud reachable-time *value*

  The neighbor reachable time is set.

  By default, the neighbor reachable time is 30000 ms.

- Run:

  ipv6 nd ra router-lifetime *ra-lifetime*

  The time to live (TTL) is set for RA packets.

  By default, the TTL of an RA packet is 1800s.

- Run:

  ipv6 nd dad attempts *value*

  The number of times NS packets are sent when the system performs Duplicate Address Detection is set.

  By default, the number of times NS packets are sent when the system performs DAD is 1.

- Run:

  ipv6 nd neighbor-limit *limit-number*

  The maximum number of dynamic neighbor entries that can be learned by a specified interface is set.

  By default, a specified interface can learn a maximum of 1024 dynamic neighbor entries.

----End

9.7.3 Checking the Configuration

Procedure

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command to check IPv6 information on an interface.
- Run the **display ipv6 neighbors** [*ipv6-address* | [**vid** *vid*] *interface-type interface-number* | **vpn-instance** *vpn-instance-name*] command to check information about neighbor entries.

9.8 Configuring PMTU

The PMTU is the minimum MTU of the path from the source to the destination. Packets that are sent according to this MTU do not need to be fragmented during transmission. This reduces loads on routing devices and optimizes network resource efficiency to obtain the maximum throughput.

Pre-configuration Tasks

Before configuring PMTU, complete the following tasks:

- Configuring IPv6 addresses for interfaces
- Configuring the IPv6 MTU value of the interfaces

NOTE

The switch supports the MTU setting on a VLANIF interface. Then packets sent by the protocol stack are fragmented based on the configured MTU. However, the hardware chip does not support the MTU setting, and the default MTU is 12K.

9.8.1 Configuring Static PMTU

Context

You can manually configure the PMTU based on the minimum MTU of the paths that packets pass through.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 pathmtu ipv6-address [ [ vpn-instance vpn-instance-name ] path-mtu ]
```

The PMTU is set for a specified IPv6 address.

By default, the PMTU of a specified destination IPv6 address is 1500 bytes.

----End

9.8.2 Setting the Aging Time of Dynamic PMTU

Context

After a source node obtains a PMTU to reach a destination node using the PMTU mechanism, the source node sends packets to the destination device based on the PMTU. After the PMTU aging time expires, the PMTU is deleted. The source node then uses the PMTU mechanism to obtain a new MTU.

NOTE

When both static PMTU and dynamic PMTU are configured, only static PMTU takes effect. Static PMTU entries never age.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 pathmtu age age-time
```

The aging time is set for dynamic PMTU entries.

By default, the aging time of dynamic PMTU entries is 10 minutes.

----End

9.8.3 Checking the Configuration

Procedure

- Run the **display ipv6 pathmtu** [**vpn-instance** *vpn-instance-name*]{ *ipv6-address* | **all** | **dynamic** | **static** } command to check all PMTU entries.
- Run the **display ipv6 interface** [*interface-type interface-number*] command to check the current MTU on the interface.

9.9 Configuring TCP6

You can configure TCP6 attributes to improve network performance.

Pre-configuration Tasks

Before configuring TCP6, complete the following task:

- Configuring link layer protocol parameters for interfaces to ensure that the link layer protocol status on the interfaces is Up

9.9.1 Setting TCP6 Timers

Context

You need to set the following TCP6 timers:

- SYN-Wait timer: When SYN packets are sent, the SYN-Wait timer is started. If no response packet is received after the SYN-Wait timer expires, the TCP6 connection is terminated.
- FIN-Wait timer: When the TCP connection status changes from FIN_WAIT_1 to FIN_WAIT_2, the FIN-Wait timer is started. If no response packet is received after the FIN-Wait timer expires, the TCP6 connection is terminated.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
tcp ipv6 timer syn-timeout interval
```

The SYN-Wait timer is set for TCP6 connections.

By default, the value of the SYN-Wait timer is set to 75s.

Step 3 Run:

```
tcp ipv6 timer fin-timeout interval
```

The FIN-Wait timer is set for TCP6 connections.

By default, the value of the FIN-Wait timer is set to 600s.

----End

9.9.2 Setting the TCP6 Sliding Window Size

Context

You can set the TCP6 sliding window size to improve network performance. The sliding window size indicates the receive or send buffer size of a TCP6 socket.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
tcp ipv6 window window-size
```

The receive and send buffer sizes of a TCP6 socket are set.

The receive or send buffer size of a TCP6 socket ranges from 1 KB to 32 KB. By default, the receive or send buffer size of a TCP6 socket is 8 KB.

----End

9.9.3 Setting the Minimum MSS Value for a TCP6 Connection

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
tcp ipv6 min-mss mss-value
```

The minimum MSS value is set for a TCP6 connection.

The default minimum MSS value for a TCP6 connection is 216 bytes.

----End

9.9.4 Checking the Configuration

Procedure

- Run the **display tcp ipv6 status** [[**task-id** *task-id* [**socket-id** *socket-id*]] | [**local-ip** *ipv6-address*] [**local-port** *local-port-number*] [**remote-ip** *ipv6-address*] [**remote-port** *remote-port-number*]] command to check TCP6 connection status.

- Run the **display tcp ipv6 statistics** command to check TCP6 traffic statistics.
- Run the **display udp ipv6 statistics** command to check UDP6 statistics.
- Run the **display ipv6 socket** [**socktype** *socket-type* | **task-id** *task-id* **socket-id** *socket-id*] command to check information about a specified socket.

----End

9.10 Maintaining IPv6

Maintaining IPv6 includes clearing IPv6 statistics and monitoring IPv6 running status.

9.10.1 Clearing IPv6 Statistics

Context

CAUTION

IPv6 statistics cannot be restored after being cleared. Therefore, exercise caution before clearing IPv6 statistics.

Procedure

- Run the **reset ipv6 attack-source overlapping-fragment** command in the user view to clear statistics on overlapping fragment attack packets.
- Run the **reset ipv6 socket pktsort task-id** *task-id* **socket-id** *socket-id* command in the user view to clear statistics on the dual receive buffer of an IPv6 socket.
- Run the **reset rawip ipv6 statistics** command in the user view to clear all Raw IPv6 packet statistics.
- Run the **reset tcp ipv6 authentication-statistics src-ip** *src-ip* **src-port** *src-port* **dest-ip** *dest-ip* **dest-port** *dest-port* command in the user view to clear authentication statistics of a specified TCP6 connection.
- Run the **reset ipv6 statistics** command in the user view to clear IPv6 traffic statistics.
- Run the **reset tcp ipv6 statistics** command in the user view to clear TCP6 statistics.
- Run the **reset udp ipv6 statistics** command in the user view to clear UDP6 statistics.
- Run the **reset ipv6 pathmtu** [**vpn-instance** *vpn-instance-name*] { **all** | **dynamic** | **static** } command in the user view to clear PMTU entries.
- Run the **reset ipv6 neighbors** { **all** | **dynamic** | **static** | **vid** *vlan-id* [*interface-type interface-number*] | *interface-type interface-number* [**dynamic** | **static**] } command in the user view to clear IPv6 neighbor entries.

----End

9.10.2 Monitoring IPv6 Running Status

Context

In routine maintenance, you can run the following commands in any view to view the IPv6 running status.

Procedure

- Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command to check IPv6 information on an interface.
- Run the **display ipv6 statistics** command to check IPv6 packet statistics.
- Run the **display icmpv6 statistics** command to check ICMPv6 packet statistics.
- Run the **display tcp ipv6 status** command in any view to check the TCP6 connection status.
- Run the **display tcp ipv6 statistics** command to check TCP6 statistics.
- Run the **display udp ipv6 statistics** command in any view to check UDP6 statistics.
- Run the **display ipv6 neighbors** [*ipv6-address* | [**vid** *vid*] *interface-type interface-number* | **vpn-instance** *vpn-instance-name*] command to check neighbor entries.
- Run the **display ipv6 socket** [**socktype** *socket-type* | **task-id** *task-id* **socket-id** *socket-id*] command in any view to check information about a specified socket.
- Run the **display ipv6 pathmtu** [**vpn-instance** *vpn-instance-name*] { *ipv6-address* | **all** | **dynamic** | **static** } command to check all PMTU entries.
- Run the **display default-parameter tcp6** command to check the default values of all configurable parameters on the TCP6 module.
- Run the **display ipv6 attack-source overlapping-fragment** command to check source information about overlapping fragment attacks.
- Run the **display rawip ipv6 statistics** command to check Raw IPv6 packet statistics.
- Run the **display tcp ipv6 authentication-statistics** command to check authentication statistics of a specified TCP6 connection.
- Run the **display this ipv6 interface** command to check IPv6 information on the current interface.

----End

9.11 Configuration Examples

This section provides IPv6 configuration examples, including networking requirements and configuration roadmap.

9.11.1 Example for Configuring IPv6 Addresses for Interfaces

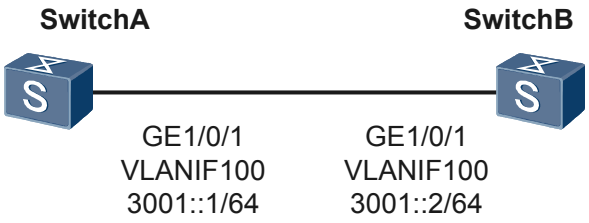
Networking Requirements

As shown in **Figure 9-1**, GE1/0/1 of SwitchA connects to GE1/0/1 of SwitchB. The two interfaces correspond to their VLANIF interfaces (VLANIF 100). You need to configure IPv6

global unicast addresses for the VLANIF interfaces and check the Layer 3 interconnection between the interfaces.

IPv6 global unicast addresses for the VLANIF interfaces are 3001::1/64 and 3001::2/64.

Figure 9-1 Networking diagram for configuring IPv6 addresses for interfaces



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable the IPv6 forwarding function on SwitchA and SwitchB.
2. Configure IPv6 global unicast addresses for the interfaces.

Procedure

Step 1 Enable the IPv6 forwarding function on switches.

Configure SwitchA.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] ipv6
```

Configure SwitchB.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] ipv6
```

Step 2 Configure global unicast addresses for interfaces.

Configure SwitchA.

```
[SwitchA] vlan 100
[SwitchA-vlan100] quit
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 100
[SwitchA-Vlanif100] ipv6 enable
[SwitchA-Vlanif100] ipv6 address 3001::1/64
[SwitchA-Vlanif100] quit
```

Configure SwitchB.

```
[SwitchB] vlan 100
[SwitchB-vlan100] quit
[SwitchB] interface gigabitethernet 1/0/1
```

```
[SwitchB-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchB-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchB-GigabitEthernet1/0/1] quit
[SwitchB] interface vlanif 100
[SwitchB-Vlanif100] ipv6 enable
[SwitchB-Vlanif100] ipv6 address 3001::2/64
[SwitchB-Vlanif100] quit
```

Step 3 Verify the configuration.

If the preceding configurations are successful, you can view the configured global unicast addresses. The interface status and the IPv6 protocol are Up.

Check interface information on SwitchA.

```
[SwitchA] display ipv6 interface vlanif 100
Vlanif100 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::218:20FF:FE00:83
  Global unicast address(es):
    3001::1, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF00:1
    FF02::1:FF00:83
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
```

Check interface information on SwitchB.

```
[SwitchB] display ipv6 interface vlanif 100
Vlanif100 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::2E0:FCFF:FE33:11
  Global unicast address(es):
    3001::2, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF00:2
    FF02::1:FF33:11
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
```

Ping the link-local address of SwitchB from SwitchA. You need to use the parameter **-i** to specify the interface of the link-local address.

```
[SwitchA] ping ipv6 FE80::2E0:FCFF:FE33:11 -i vlanif 100
  PING FE80::2E0:FCFF:FE33:11 : 56  data bytes, press CTRL_C to break
    Reply from FE80::2E0:FCFF:FE33:11
    bytes=56 Sequence=1 hop limit=64  time = 7 ms
    Reply from FE80::2E0:FCFF:FE33:11
    bytes=56 Sequence=2 hop limit=64  time = 3 ms
    Reply from FE80::2E0:FCFF:FE33:11
    bytes=56 Sequence=3 hop limit=64  time = 3 ms
    Reply from FE80::2E0:FCFF:FE33:11
    bytes=56 Sequence=4 hop limit=64  time = 3 ms
    Reply from FE80::2E0:FCFF:FE33:11
    bytes=56 Sequence=5 hop limit=64  time = 3 ms
```

```
  --- FE80::2E0:FCFF:FE33:11 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/3/7 ms
```

Ping the IPv6 global unicast address of SwitchB from SwitchA.

```
[SwitchA] ping ipv6 3001::2
  PING 3001::2 : 56  data bytes, press CTRL_C to break
    Reply from 3001::2
    bytes=56 Sequence=1 hop limit=64  time = 12 ms
    Reply from 3001::2
    bytes=56 Sequence=2 hop limit=64  time = 3 ms
    Reply from 3001::2
    bytes=56 Sequence=3 hop limit=64  time = 3 ms
    Reply from 3001::2
    bytes=56 Sequence=4 hop limit=64  time = 3 ms
    Reply from 3001::2
    bytes=56 Sequence=5 hop limit=64  time = 3 ms

  --- 3001::2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/4/12 ms
```

----End

Configuration File

- Configuration file of SwitchA

```
#
 sysname SwitchA
#
ipv6
#
 vlan batch 100
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3001::1/64
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
return
```

- Configuration file of SwitchB

```
#
 sysname SwitchB
#
ipv6
#
 vlan batch 100
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3001::2/64
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
return
```

10 IPv6 DNS configuration

About This Chapter

This section describes how to configure IPv6 DNS so that devices can use domain names to communicate.

10.1 IPv6 DNS Overview
IPv6 DNS is a distributed database used in TCP and IP applications and completes resolution between IPv6 addresses and domain names.

10.2 IPv6 DNS Features Supported by the Device
The switch can function as an IPv6 DNS client.

10.3 Configuring the IPv6 DNS Client
This section describes how to configure the IPv6 DNS client and the mapping between a domain name and IPv6 address on a device, so that the device can communicate with other devices using the domain name.

10.4 Maintaining IPv6 DNS
IPv6 DNS maintenance includes monitoring IPv6 DNS running status.

10.5 Configuration Examples
This section describes configuration examples of IPv6 DNS.

10.1 IPv6 DNS Overview

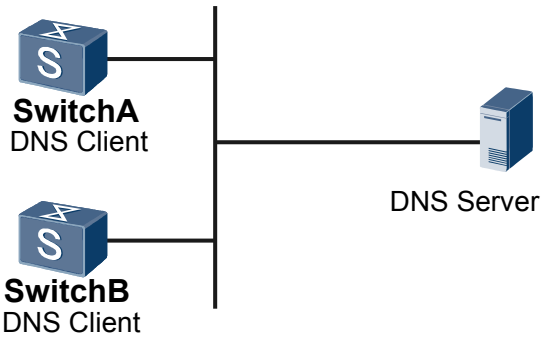
IPv6 DNS is a distributed database used in TCP and IP applications and completes resolution between IPv6 addresses and domain names.

Each host on the IPv6 network is identified by an IPv6 address. To access a host, a user must obtain the host IPv6 address first. It is difficult for users to remember IPv6 addresses of hosts. Therefore, host names in the format of strings are designed. Each host name maps an IPv6 address. In this way, users can use the simple and meaningful domain names instead of the complicated IPv6 addresses to access hosts by resolution of the IPv6 DNS server on the network.

10.2 IPv6 DNS Features Supported by the Device

The switch can function as an IPv6 DNS client.

Figure 10-1 Functioning as an IPv6 DNS client



As shown in **Figure 10-1**, the switch functions as an IPv6 DNS client and supports static and dynamic domain name resolution.

- Static domain name resolution. Mappings between domain names and IPv6 addresses are configured manually. To obtain the IPv6 address by resolving a domain name, the client searches the static domain name resolution table for the specified domain name.
- Dynamic DNS resolution. Dynamic DNS resolution is implemented by a DNS server. The DNS server receives domain name resolution requests from DNS clients. The DNS server searches for the corresponding IPv6 address of the domain name in its DNS database. If no matching entry is found, it sends a query message to a higher-level DNS server. This process continues until the DNS server finds the corresponding IPv6 address or detects that the corresponding IPv6 address of the domain name does not exist. Then the DNS server returns a result to the DNS client.

  The switch IPv6 domain name resolution system must support the following DNS query modes:

  - AAAA query
  - A6 query
  - IPv6 PTR query

10.3 Configuring the IPv6 DNS Client

This section describes how to configure the IPv6 DNS client and the mapping between a domain name and IPv6 address on a device, so that the device can communicate with other devices using the domain name.

10.3.1 Configuring Static IPv6 DNS Entries

Context

A static domain name resolution table is manually set up, describing the mappings between domain names and IPv6 addresses. Some common domain names are added to the table. Static domain name resolution can be performed based on the static domain name resolution table. To obtain the IPv6 address by resolving a domain name, the client searches the static domain name resolution table for the specified domain name. In this manner, the efficiency of domain name resolution is improved.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6
```

The IPv6 function is enabled.

Step 3 Run:

```
ipv6 host host-name ipv6-address
```

The domain name and mapping IPv6 address are configured.

If multiple IPv6 addresses (a maximum of eight IPv6 addresses) are configured to map a domain name, the DNS client preferentially resolves the domain name to the first IPv6 address.

----End

10.3.2 Configuring the Dynamic IPv6 DNS Service

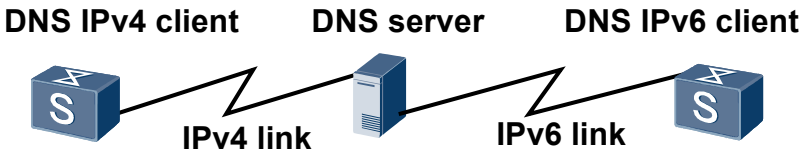
Context

Dynamic domain name resolution requires a special DNS server. This server provides mappings between domain names and IPv6 addresses, and processes DNS client's request for domain name resolution.

To implement dynamic DNS, you need to enable dynamic DNS resolution, configure a DNS server, and configure a source IPv6 address for the local device and a domain name suffix. If the local device uses an IPv6 address allocated by the DHCPv6 server and the information

delivered by the DHCPv6 server to the local device contains the DNS server IPv6 address and the domain name suffix list, you only need to enable dynamic DNS resolution.

Figure 10-2 DNS server connecting the IPv4 network and the IPv6 network



CAUTION

If multiple DNS servers are configured, query messages are sent to the DNS servers according to the order in which they are configured till correct reply packets are received. If both IPv4 and IPv6 servers are configured on the DNS client, the system sends a Class-A query packet to the IPv4 server and sends an AAAA query message to the IPv6 server.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
dns resolve
```

Dynamic domain name resolution is enabled.

Step 3 Run:

```
dns server ipv6 ipv6-address [ interface-type interface-number ]
```

The IPv6 DNS server is configured.

A maximum of six DNS server IPv6 addresses can be configured on the device.

Step 4 Run:

```
dns server ipv6 source-ip ipv6-address
```

The IPv6 address of the local switch is specified.

After the IPv6 address of the local switch is specified, the switch uses the specified IPv6 address to communicate with the DNS server to ensure the security check.

Step 5 Run:

```
dns domain domain-name
```

A suffix of a domain name is added.

----End

10.3.3 Checking the Configuration

Procedure

- Run the **display ipv6 host** command to view the static IPv6 DNS table.
- Run the **display dns server** command to check the DNS server configuration.
- Run the **display dns domain** command to check the domain name suffix configuration.
- Run the **display dns ipv6 dynamic-host** command to check IPv6 dynamic entries saved in the cache.

----End

10.4 Maintaining IPv6 DNS

IPv6 DNS maintenance includes monitoring IPv6 DNS running status.

10.4.1 Monitoring the Running Status of IPv6 DNS

Context

In routine maintenance, you can run the following commands in any view to check the running status of IPv6 DNS.

Procedure

- Run the **display dns domain** command to check the domain name suffix configuration.
- Run the **display dns server** command to check the DNS server configuration.
- Run the **display dns ipv6 dynamic-host** command to check IPv6 dynamic DNS entries saved in the cache.
- Run the **display ipv6 host** command to check the static DNS table.

----End

10.5 Configuration Examples

This section describes configuration examples of IPv6 DNS.

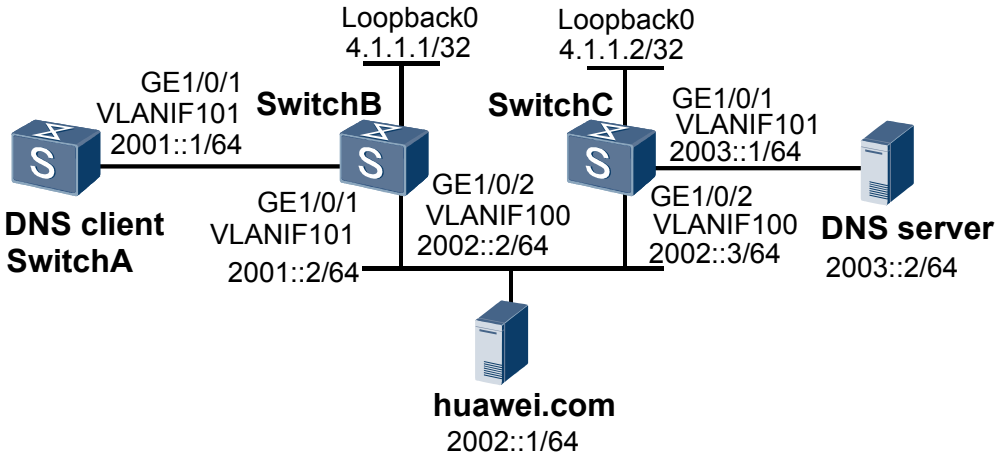
10.5.1 Example for Configuring IPv6 DNS Client

Networking Requirements

As shown in **Figure 10-3**, SwitchA, functioning as the IPv6 DNS client and working jointly with IPv6 DNS server, can access the host with the IPv6 address as 2002::1/64 based on the domain name huawei.com.

On SwitchA, the static IPv6 DNS entries of SwitchB and SwitchC are configured. This ensures that SwitchA can manage both the devices based on the domain names SwitchB and SwitchC.

Figure 10-3 Networking diagram of IPv6 DNS configurations



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure static DNS entries on SwitchA to access SwitchB and SwitchC using the domain name.
2. Configure dynamic DNS resolution on SwithcA to enable SwitchA to access the web server by querying dynamic DNS entries.
3. Configure domain name suffixes on SwitchA so that SwitchA can filter domain names using the domain name suffix list.
4. Configure OSPF on the switches to ensure reachable routes between them.

Procedure

Step 1 Configure SwitchA.

Configure IPv6 function.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] ipv6
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 101
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 101
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 101
[SwitchA-Vlanif101] ipv6 enable
[SwitchA-Vlanif101] ipv6 address 2001::1/64
[SwitchA-Vlanif101] quit
```

Configure static IPv6 DNS entries.

```
[SwitchA] ipv6 host SwitchB 2001::2
[SwitchA] ipv6 host SwitchC 2002::3
```

Enable the DNS resolution function.

```
[SwitchA] dns resolve
```

Configure the IPv6 address of the IPv6 DNS server.

```
[SwitchA] dns server ipv6 2003::2
```

Set the domain name suffix to ".net".

```
[SwitchA] dns domain net
```

Set the domain name suffix to ".com".

```
[SwitchA] dns domain com
[SwitchA] quit
```

NOTE

To resolve the domain name, you also need to configure the route from Switch A to the IPv6 DNS server. For details of how to configure the route, see Configuration example of IP static route in the *S7700&S9700 Smart&Core Routing Switch Configuration Guide: IP Routing*.

Step 2 Verify the configuration.

Run the **ping ipv6 huawei.com** command on Switch A. You can find that the Ping operation succeeds, and the destination IPv6 address is 2002::1.

```
<SwitchA> ping ipv6 huawei.com
  Resolved Host ( huawei.com -> 2002::1)
  PING huawei.com : 56  data bytes, press CTRL_C to break
    Reply from 2002::1: bytes=56 Sequence=1 ttl=126 time=6 ms
    Reply from 2002::1: bytes=56 Sequence=2 ttl=126 time=4 ms
    Reply from 2002::1: bytes=56 Sequence=3 ttl=126 time=4 ms
    Reply from 2002::1: bytes=56 Sequence=4 ttl=126 time=4 ms
    Reply from 2002::1: bytes=56 Sequence=5 ttl=126 time=4 ms

  --- huawei.com ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 4/4/6 ms
```

Run the **display ipv6 host** command on SwitchA. You can view the mapping relationships between the host names and the IPv6 addresses in IPv6 static DNS entries.

```
<SwitchA> display ipv6 host
Host              Age        Flags   IPv6Address (es)
SwitchB           0          static  2001::2
SwitchC           0          static  2002::3
```

Run the **display dns ipv6 dynamic-host** command on SwitchA. You can view information about IPv6 dynamic DNS entries in the dynamic cache.

```
<SwitchA> display dns ipv6 dynamic-host
No  Domain-name           Ipv6address          TTL
1   huawei.com            2002::1              3579
```

NOTE

TTL in the command output indicates the life time of the entry, in seconds.

----End

Configuration Files

- Configuration file of SwitchA
- ```
  #
   sysname SwitchA
  #
  ```
  ```

```
vlan batch 101
#
 ipv6
#
 ipv6 host SwitchB 2001::2
 ipv6 host SwitchC 2002::3
#
 dns resolve
 dns server ipv6 2003::2
 dns domain net
 dns domain com
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 101
 port hybrid untagged vlan 101
#
interface Vlanif101
 ipv6 enable
 ipv6 address 2001::1/64
#
return
```

- Configuration file of SwitchB

```
#
 sysname SwitchB
#
vlan batch 100 to 101
#
 ipv6
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 101
 port hybrid untagged vlan 101
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface Vlanif100
 ipv6 enable
 ipv6 address 2002::2/64
#
interface Vlanif101
 ipv6 enable
 ipv6 address 2001::2/64
#
return
```

- Configuration file of SwitchC

```
#
 sysname SwitchC
#
vlan batch 100 to 101
#
 ipv6
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 101
 port hybrid untagged vlan 101
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface Vlanif100
 ipv6 enable
 ipv6 address 2002::3/64
```

```
#
interface Vlanif101
ipv6 enable
 ipv6 address 2003::1/64
#
return
```

11 IPv6 over IPv4 Tunnel Configuration

About This Chapter

IPv6 over IPv4 tunnel technology enables transition from the IPv4 network to the IPv6 network.

11.1 IPv6 over IPv4 Tunnel Overview
An IPv6 over IPv4 tunnel connects isolated IPv6 sites through an IPv4 network.

11.2 IPv6 over IPv4 Tunnel Features Supported by the Device
The device supports the IPv4/IPv6 dual stack, 6PE, and IPv6 over IPv4 tunnels.

11.3 Configuring the IPv4/IPv6 Dual Stack
To establish IPv6 over IPv4 tunnels, you need to enable the IPv4/IPv6 dual stack on devices at the edge of the IPv6 network and the IPv4 network.

11.4 Configuring an IPv6 over IPv4 Tunnel
An IPv6 over IPv4 tunnel connects IPv6 networks through an IPv4 network.

11.5 Configuring 6PE
You can configure IPv6 Provider Edge (6PE) to connect IPv6 networks through an existing MPLS network.

11.6 Maintaining the IPv6 over IPv4 Tunnel
IPv6 over IPv4 tunnel maintenance includes monitoring the running status of the IPv6 over IPv4 tunnel.

11.7 Configuration Examples
Configuration examples include networking requirements, configuration roadmap, and configuration procedure.

11.1 IPv6 over IPv4 Tunnel Overview

An IPv6 over IPv4 tunnel connects isolated IPv6 sites through an IPv4 network.

Exhaustion of IPv4 addresses urgently requires IPv4 to IPv6 transition. IPv6 is incompatible with IPv4, so original IPv4 devices need to be replaced. This solution is infeasible because the replacement requires huge capital expenditures, and will interrupt services on the live network. In this situation, IPv4 needs to transit to IPv6 gradually. During early transition, IPv4 networks are widely deployed and IPv6 networks are isolated sites. An IPv6 over IPv4 tunnel allows IPv6 packets to be transmitted on an IPv4 network and connects all IPv6 sites.

11.2 IPv6 over IPv4 Tunnel Features Supported by the Device

The device supports the IPv4/IPv6 dual stack, 6PE, and IPv6 over IPv4 tunnels.

IPv4/IPv6 Dual Stack

The IPv4/IPv6 dual stack helps implement IPv4 to IPv6 transition. Network nodes must support IPv4 and IPv6 protocol stacks. A source node chooses a protocol stack according to the destination node, and a network device chooses a protocol stack based on the protocol type of packets. The IPv4/IPv6 dual stack can be implemented on a single device or on a dual-stack backbone network. On a dual-stack backbone network, all devices must support the IPv4/IPv6 dual stack, and interfaces connected to the dual-stack network must be configured with both IPv4 addresses and IPv6 addresses.

Dual stack technology is the prerequisite for IPv4 to IPv6 transition. You must configure the IPv4/IPv6 dual stack on the border devices before configuring an IPv6 over IPv4 tunnel.

IPv6 over IPv4 Tunnel

Exhaustion of IPv4 addresses urgently requires IPv4 to IPv6 transition. IPv6 is incompatible with IPv4, so original IPv4 devices need to be replaced. This solution is infeasible because the replacement requires huge capital expenditures, and will interrupt services on the live network. In this situation, IPv4 needs to transit to IPv6 gradually. During early transition, IPv4 networks are widely deployed and IPv6 networks are isolated sites. An IPv6 over IPv4 tunnel allows IPv6 packets to be transmitted on an IPv4 network and connects all IPv6 sites.

The following table describes key configurations and usage scenarios of the IPv6 over IPv4 tunnel.

Category	Subcategory	Tunnel Source/ Destination IP Address	Tunnel Interface IP Address	Usage Scenario
Manual tunnel	Manual IPv6 over IPv4 tunnel	Source and destination IP addresses use manually configured IPv4 addresses.	IPv6 address	Applies to simple IPv6 networks or point-to-point connections. Only IPv6 packets can be transmitted over the manual IPv6 over IPv4 tunnel.
	IPv6 over IPv4 GRE tunnel	Source and destination IP addresses use manually configured IPv4 addresses.	IPv6 address	Applies to simple IPv6 networks or point-to-point connections. The IPv6 over IPv4 GRE tunnel supports multiple upper-layer protocols including IPv6.
Other tunnel	6to4 tunnel	The source IP address uses a manually configured IPv4 address, and the destination address is automatically generated.	6to4 address in the format of 2002:IPv4-source-address::/48	Applies to point-to-multipoint connections on IPv6 networks.
	ISATAP tunnel	The source IP address uses a manually configured IPv4 address, and the destination address is automatically generated.	ISATAP address in the format of Prefix:0	Applies to connections of IPv6 nodes on an IPv4 network.

6PE

IPv6 provider edge (6PE) technology allows ISPs to provide access services for scattered IPv6 networks over the existing IPv4 backbone network. The 6PE device converts IPv6 routes to labeled IPv6 routes and floods the routes on the ISP's IPv4 backbone network using Internal Border Gateway Protocol (IBGP) sessions. The 6PE device tags the IPv6 packets before

forwarding them to tunnels on the IPv4 backbone network. The tunnels can be GRE tunnels or MPLS LSPs. MPLS 6PE technology allows ISPs to connect existing IPv4/MPLS networks to IPv6 networks by simply upgrading PEs. Therefore, using 6PE as an IPv6 transition mechanism is a cost-effective solution for ISPs.

11.3 Configuring the IPv4/IPv6 Dual Stack

To establish IPv6 over IPv4 tunnels, you need to enable the IPv4/IPv6 dual stack on devices at the edge of the IPv6 network and the IPv4 network.

Pre-configuration Tasks

Before configuring an IPv4/IPv6 dual stack, complete the following tasks:

- Connecting interfaces and setting physical parameters for the interfaces to ensure that the physical status of interfaces is Up
- Configuring link layer protocol parameters for interfaces

11.3.1 Enabling IPv6 Packet Forwarding

Context

To enable an interface to forward IPv6 packets, enable IPv6 packet forwarding in the system view and in the interface view.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6
```

IPv6 packet forwarding is enabled.

By default, IPv6 packet forwarding is disabled on the device.

To enable a device to forward IPv6 packets, enable IPv6 packet forwarding in the system view; otherwise, the device fails to forward IPv6 packets even if an IPv6 address is configured for an interface on the device.

Step 3 Run:

```
interface interface-type interface-number
```

The view of the interface to be enabled with IPv6 is displayed.

Step 4 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

Before performing IPv6 configurations in the interface view, enable the IPv6 function in the interface view.

By default, the IPv6 function is disabled on an interface.

----End

11.3.2 Configuring an IPv4 Address and an IPv6 Address for Interfaces Respectively

Context

The device to be enabled with the dual stack must be configured with an IPv4 address on the IPv4 network-side interface and an IPv6 address on the IPv6 network-side interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vlanif vlan-id
```

The IPv4 network-side interface view is displayed.

Step 3 Run:

```
ip address ip-address { mask | mask-length }
```

An IPv4 address is configured for the interface.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface vlanif vlan-id
```

The IPv6 network-side interface view is displayed.

Step 6 Run the following commands as required.

- Run:

  ```
  ipv6 address auto link-local
  ```

  The interface is configured to automatically generate a link-local address.

- Run:

  ```
  ipv6 address ipv6-address link-local
  ```

  A link-local ipv6 address is manually configured for the interface.

- Run:

  ```
  ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length  }
  ```

  A global unicast ipv6 address is configured for the interface.

- Run:

  ```
  ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length  } eui-64
  ```

An IPv6 address in EUI-64 format is configured for the interface.

----End

11.3.3 Checking the Configuration

Prerequisites

All configurations of the IPv4/IPv6 dual stack are complete.

Procedure

Step 1 Run the **display ipv6 interface** [*interface-type interface-number* | **brief**] command to check IPv6 attributes of an interface.

----End

11.4 Configuring an IPv6 over IPv4 Tunnel

An IPv6 over IPv4 tunnel connects IPv6 networks through an IPv4 network.

Prerequisites

Source and destination devices of an IPv6 over IPv4 tunnel have forwarding routes.

Pre-configuration Tasks

Before configuring an IPv6 over IPv4 tunnel, complete the following task:

- Configuring the IPv4/IPv6 dual stack

11.4.1 Enabling the Service Loopback Function on an Eth-Trunk

Context

Before enabling the service loopback function on an Eth-Trunk, pay attention to the following points:

- Ensure that an Eth-Trunk has been created, and member interfaces are added to the Eth-Trunk and in Up state.
- Only one interface enabled with the service loopback function is needed on a device.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

interface eth-trunk *trunk-id*

The Eth-Trunk interface view is displayed.

Step 3 Run:

```
service type tunnel
```

The Eth-Trunk is enabled with the service loopback function.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Step 6 Run:

```
eth-trunk trunk-id
```

The interface is added to the Eth-Trunk.

----End

11.4.2 Configuring a Manual IPv6 over IPv4 Tunnel

Context

When configuring a manual IPv6 over IPv4 tunnel, pay attention to the following points:

- You must create a tunnel interface before setting tunnel parameters.
- When the specified tunnel source interface is a physical interface, you are advised to set the tunnel number to be the same as the tunnel source interface number.
- The following configurations must be performed on devices at both ends of the tunnel.
- To support a dynamic routing protocol, configure a network address for the tunnel interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface tunnel interface-number
```

A tunnel interface is created.

Step 3 Run:

```
tunnel-protocol ipv6-ipv4
```

The tunnel mode is set to manual.

Step 4 Run:

```
eth-trunk trunk-id
```

The interface is added to the Eth-Trunk.

Step 5 Run:

```
source { ip-address | interface-type interface-number }
```

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

Step 6 Run:

```
destination dest-ip-address
```

A destination address is specified for the tunnel.

NOTE

The destination address of a tunnel can be the IP address of a physical interface or loopback interface.

Step 7 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

Step 8 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }
```

An IPv6 address is configured for the tunnel interface.

----End

11.4.3 Configuring a 6to4 Tunnel

Context

When configuring a 6to4 tunnel, pay attention to the following points:

- You must create a tunnel interface before setting tunnel parameters.
- You are advised to set the tunnel number to be the same as the number of the source physical interface of the tunnel.
- You only need to specify the source address of the tunnel when a 6to4 tunnel is configured. The destination address of the tunnel is obtained from the destination address of the original IPv6 packet. In addition, the source address of a 6to4 tunnel must be unique.
- You need to configure a 6to4 address for the interface that connects a border device to the 6to4 network, and an IPv4 address for the interface that connects a border device to the IPv4 network. You also need to configure a network address for the tunnel interface to support a dynamic routing protocol.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface tunnel interface-number
```

A tunnel interface is created.

Step 3 Run:

```
tunnel-protocol ipv6-ipv4 6to4
```

The tunnel mode is set to 6to4.

Step 4 Run:

```
eth-trunk trunk-id
```

The interface is added to the Eth-Trunk.

Step 5 Run:

```
source { source-ip-address | interface-type interface-number }
```

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

Step 6 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

Step 7 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }
```

An IPv6 address is configured for the tunnel interface.

NOTE

The IPv6 address prefix of the specified tunnel interface must be the same as the address prefix of the 6to4 network that the device belongs to.

----End

11.4.4 Configuring an ISATAP Tunnel

Context

When configuring an ISATAP tunnel, pay attention to the following points:

- You must create a tunnel interface before setting tunnel parameters.
- You are advised to set the tunnel number to be the same as the number of the source physical interface of the tunnel.
- The source interface of a tunnel is the physical interface that forwards tunnel packets. You can specify an IP address or interface name for the source interface.
- You need to configure an ISATAP address with a 64-bit prefix-length as the IPv6 address of a tunnel interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface tunnel interface-number
```

A tunnel interface is created.

Step 3 Run:

```
tunnel-protocol ipv6-ipv4 isatap
```

The tunnel mode is set to ISATAP.

Step 4 Run:

```
eth-trunk trunk-id
```

The interface is added to the Eth-Trunk.

Step 5 Run:

```
source { source-ip-address | interface-type interface-number }
```

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

Step 6 Run:

```
ipv6 enable
```

The IPv6 function is enabled on the interface.

Step 7 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length } eui-64
```

An IPv6 address is configured for the tunnel interface.

Step 8 Run:

```
undo ipv6 nd ra halt
```

The device is enabled to send router advertisement (RA) packets.

----End

11.4.5 Checking the Configuration

Prerequisites

All configurations of the IPv6 over IPv4 tunnel are complete.

Procedure

Step 1 Run the **display ipv6 interface tunnel** *interface-number* command to check IPv6 attributes of a tunnel interface.

----End

11.5 Configuring 6PE

You can configure IPv6 Provider Edge (6PE) to connect IPv6 networks through an existing MPLS network.

Pre-configuration Tasks

Before configuring 6PE, complete the following tasks:

- Enabling the dual stack on 6PE switches
- Configuring a 6PE-to-CE route
- Configuring a reachable route on the backbone network

11.5.1 Configuring the MPLS Function

Context

Configuring basic MPLS functions includes setting up an LSP and enabling LDP.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
mpls lsr-id *ip-address*

An LSR ID is specified.

Step 3 Run:
mpls

MPLS is enabled and the MPLS view is displayed.

Step 4 Run:
quit

Return to the system view.

Step 5 Run:
mpls ldp

LDP is enabled.

Step 6 Run:
quit

Return to the system view.

Step 7 Run:
interface *interface-type interface-number*

The IPv4 network-side interface view is displayed.

Step 8 Run:
mpls

MPLS is enabled on the interface.

Step 9 Run:
mpls ldp

LDP is enabled on the interface.

----End

11.5.2 Configuring a 6PE Peer

Context

A PE device configured with a 6PE peer can exchange routes with an IPv6 device in the IPv6 address family view.

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
```

The peer IP address and the ID of the AS where the peer resides are specified.

Step 4 Run:

```
peer ipv4-address connect-interface interface-type interface-number
```

The interface connected to the peer PE is specified.

Step 5 Run:

```
ipv6-family
```

The BGP-IPv6 unicast address family view is displayed.

Step 6 Run:

```
peer ipv4-address enable
```

The 6PE peer is enabled.

Step 7 Run:

```
peer ipv4-address label-route-capability
```

The 6PE device is enabled to exchange labeled routes.

----End

11.5.3 Checking the Configuration

Procedure

Step 1 Run the **display mpls lsp** command to check LSP information.

Step 2 Run the **display bgp ipv6 routing-table** command to check information about IPv6 BGP routes.

----End

11.6 Maintaining the IPv6 over IPv4 Tunnel

IPv6 over IPv4 tunnel maintenance includes monitoring the running status of the IPv6 over IPv4 tunnel.

11.6.1 Monitoring the Running Status of the IPv6 over IPv4 Tunnel

Context

In routine maintenance, you can run the following command in any view to monitor the running status of the IPv6 over IPv4 tunnel.

Procedure

Step 1 Run the **display ipv6 interface tunnel** *interface-number* command in any view to view the running status of the tunnel interface.

----End

11.7 Configuration Examples

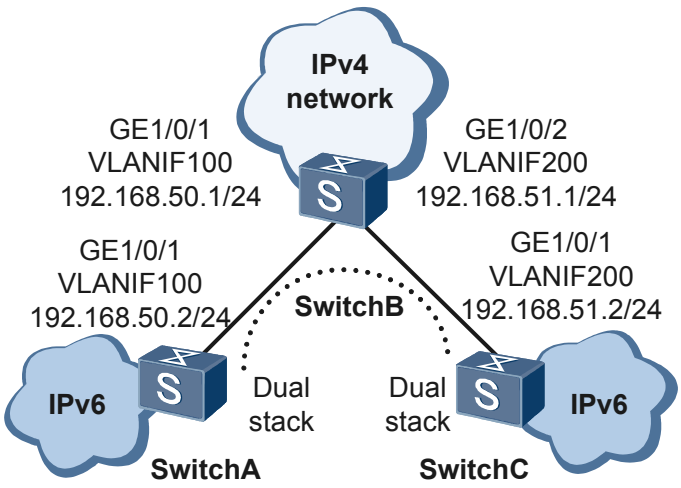
Configuration examples include networking requirements, configuration roadmap, and configuration procedure.

11.7.1 Example for Configuring a Manual IPv6 over IPv4 Tunnel

Networking Requirements

As shown in **Figure 11-1**, two IPv6 networks connect to SwitchB on an IPv4 backbone network respectively through SwitchA and SwitchC. A manual IPv6 over IPv4 tunnel needs to be set up between SwitchA and SwitchC so that hosts on the two IPv6 networks can communicate.

Figure 11-1 Networking diagram for configuring a manual IPv6 over IPv4 tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses for interfaces so that devices can communicate on the IPv4 backbone network.
2. Configure IPv6 addresses, source interfaces, and destination addresses for tunnel interfaces so that devices can communicate with hosts on the two IPv6 networks.
3. Set the tunnel protocol to IPv6-IPv4 so that hosts on the two IPv6 networks can communicate through the IPv4 backbone network.

Procedure

Step 1 Configure SwitchA.

Enable the service loopback function on an Eth-Trunk.

CAUTION

The interface must be idle. That is, the interface does not transmit services.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] interface eth-trunk 1
[SwitchA-Eth-Trunk1] service type tunnel
[SwitchA-Eth-Trunk1] quit
[SwitchA] interface gigabitethernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] eth-trunk 1
[SwitchA-GigabitEthernet1/0/3] quit
```

Configure an IP address for an interface.

```
[SwitchA] ipv6
[SwitchA] vlan 100
[SwitchA-vlan100] quit
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 100
[SwitchA-Vlanif100] ip address 192.168.50.2 255.255.255.0
[SwitchA-Vlanif100] quit
```

Set the tunnel protocol to IPv6-IPv4.

```
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] tunnel-protocol ipv6-ipv4
[SwitchA-Tunnel1] eth-trunk 1
```

Configure an IPv6 address and a destination address for the tunnel interface.

```
[SwitchA-Tunnel1] ipv6 enable
[SwitchA-Tunnel1] ipv6 address 3001::1 64
[SwitchA-Tunnel1] source vlanif 100
[SwitchA-Tunnel1] destination 192.168.51.2
[SwitchA-Tunnel1] quit
```

Configure a static route.

```
[SwitchA] ip route-static 192.168.51.2 255.255.255.0 192.168.50.1
```

Step 2 Configure SwitchB.

Configure IP addresses for interfaces.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] ipv6
[SwitchB] vlan 100
[SwitchB-vlan100] quit
[SwitchB] vlan 200
[SwitchB-vlan200] quit
[SwitchB] interface gigabitethernet 1/0/1
[SwitchB-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchB-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchB-GigabitEthernet1/0/1] quit
[SwitchB] interface gigabitethernet 1/0/2
[SwitchB-GigabitEthernet1/0/2] port hybrid pvid vlan 200
[SwitchB-GigabitEthernet1/0/2] port hybrid untagged vlan 200
[SwitchB-GigabitEthernet1/0/2] quit
[SwitchB] interface vlanif 100
[SwitchB-Vlanif100] ip address 192.168.50.1 255.255.255.0
[SwitchB-Vlanif100] quit
[SwitchB] interface vlanif 200
[SwitchB-Vlanif200] ip address 192.168.51.1 255.255.255.0
[SwitchB-Vlanif200] quit
```

Step 3 Configure SwitchC.

Enable the service loopback function on an Eth-Trunk.

CAUTION

The interface must be idle. That is, the interface does not transmit services.

```
<Quidway> system-view
[Quidway] sysname SwitchC
[SwitchC] interface eth-trunk 1
[SwitchC-Eth-Trunk1] service type tunnel
[SwitchC-Eth-Trunk1] quit
[SwitchC] interface gigabitethernet 1/0/3
[SwitchC-GigabitEthernet1/0/3] eth-trunk 1
[SwitchC-GigabitEthernet1/0/3] quit
```

Configure an IP address for an interface.

```
[SwitchC] ipv6
[SwitchC] vlan 200
[SwitchC-vlan200] quit
[SwitchC] interface gigabitethernet1/0/1
[SwitchC-GigabitEthernet1/0/1] port hybrid pvid vlan 200
[SwitchC-GigabitEthernet1/0/1] port hybrid untagged vlan 200
[SwitchC-GigabitEthernet1/0/1] quit
[SwitchC] interface vlanif 200
[SwitchC-Vlanif200] ip address 192.168.51.2 255.255.255.0
[SwitchC-Vlanif200] quit
```

Set the tunnel protocol to IPv6-IPv4.

```
[SwitchC] interface tunnel 1
[SwitchC-Tunnel1] tunnel-protocol ipv6-ipv4
[SwitchC-Tunnel1] eth-trunk 1
```

Configure an IPv6 address and a destination address for the tunnel interface.

```
[SwitchC-Tunnel1] ipv6 enable
[SwitchC-Tunnel1] ipv6 address 3001::2 64
[SwitchC-Tunnel1] source vlanif 200
[SwitchC-Tunnel1] destination 192.168.50.2
[SwitchC-Tunnel1] quit
```

Configure a static route.

```
[SwitchC] ip route-static 192.168.50.2 255.255.255.0 192.168.51.1
```

Step 4 Verify the configuration.

Ping the IPv4 address of VLANIF 100 on SwitchA from SwitchC. SwitchC can receive a Reply packet from SwitchA.

```
[SwitchC] ping 192.168.50.2
  PING 192.168.50.2: 56  data bytes, press CTRL_C to break
    Reply from 192.168.50.2: bytes=56 Sequence=1 ttl=255 time=84 ms
    Reply from 192.168.50.2: bytes=56 Sequence=2 ttl=255 time=27 ms
    Reply from 192.168.50.2: bytes=56 Sequence=3 ttl=255 time=25 ms
    Reply from 192.168.50.2: bytes=56 Sequence=4 ttl=255 time=3 ms
    Reply from 192.168.50.2: bytes=56 Sequence=5 ttl=255 time=24 ms

  --- 192.168.50.2 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 Tunnel1/0/1 on SwitchA from SwitchC. SwitchC can receive a Reply packet from SwitchA.

```
[SwitchC] ping ipv6 3001::1
  PING 3001::1 : 56  data bytes, press CTRL_C to break
    Reply from 3001::1
    bytes=56 Sequence=1 hop limit=64  time = 28 ms
    Reply from 3001::1
    bytes=56 Sequence=2 hop limit=64  time = 27 ms
    Reply from 3001::1
    bytes=56 Sequence=3 hop limit=64  time = 26 ms
    Reply from 3001::1
    bytes=56 Sequence=4 hop limit=64  time = 27 ms
    Reply from 3001::1
    bytes=56 Sequence=5 hop limit=64  time = 26 ms
  --- 3001::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 SwitchA

```
#
 sysname SwitchA
#
ipv6
#
```

```
vlan batch 100
#
interface Vlanif100
 ip address 192.168.50.2 255.255.255.0
#
interface Eth-Trunk1
 service type tunnel
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet1/0/3
 eth-trunk 1
#
interface Tunnel1
 ipv6 enable
 ipv6 address 3001::1/64
 tunnel-protocol ipv6-ipv4
 source Vlanif100
 destination 192.168.51.2
 eth-trunk 1
#
ip route-static 192.168.51.0 255.255.255.0 192.168.50.1
#
return
```

- Configuration file of SwitchB

```
#
 sysname SwitchB
#
ipv6
#
vlan batch 100 200
#
interface Vlanif100
 ip address 192.168.50.1 255.255.255.0
#
interface Vlanif200
 ip address 192.168.51.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
return
```

- Configuration file of SwitchC

```
#
 sysname SwitchC
#
ipv6
#
vlan batch 200
#
interface Vlanif200
 ip address 192.168.51.2 255.255.255.0
#
interface Eth-Trunk1
 service type tunnel
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 200
```

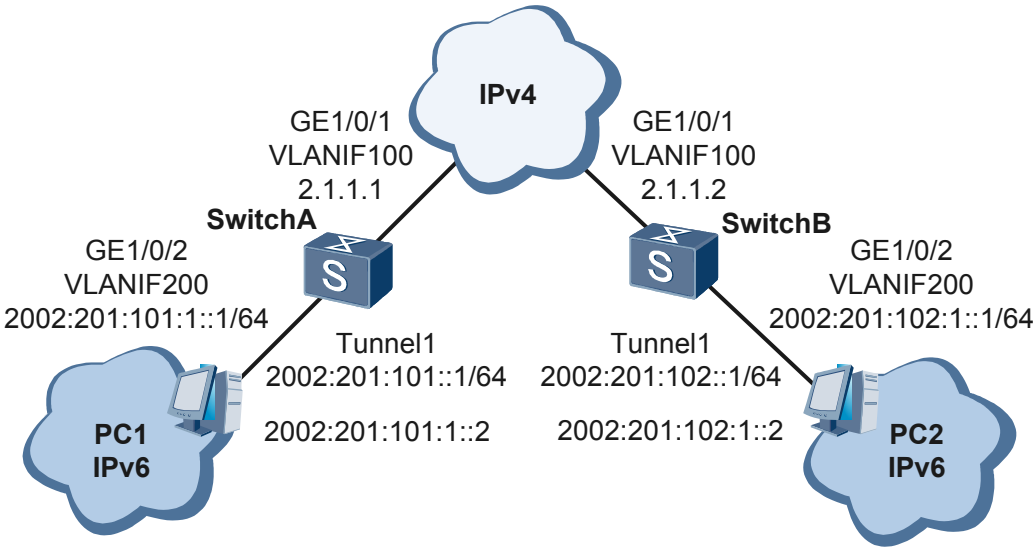
```
 port hybrid untagged vlan 200
#
interface GigabitEthernet1/0/3
 eth-trunk 1
#
interface Tunnel1
 ipv6 enable
 ipv6 address 3001::2/64
 tunnel-protocol ipv6-ipv4
 source Vlanif200
 destination 192.168.50.2
 eth-trunk 1
#
ip route-static 192.168.50.0 255.255.255.0 192.168.51.1
#
return
```

11.7.2 Example for Configuring a 6to4 Tunnel

Networking Requirements

As shown in **Figure 11-2**, the IPv6 network-side interface of 6to4 SwitchA connects to a 6to4 network. SwitchB is a 6to4 relay agent and connects to the IPv6 Internet (2002::/64). SwitchA and SwitchB are connected through an IPv4 backbone network. A 6to4 tunnel needs to be set up between SwitchA and SwitchB so that hosts on the 6to4 network and the IPv6 network can communicate.

Figure 11-2 Networking diagram for configuring a 6to4 tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IPv4/IPv6 dual stack on SwitchA and SwitchB so that they can access the IPv4 network and the IPv6 network.

2. Configure a 6to4 tunnel on SwitchA and SwitchB to connect IPv6 networks through the IPv4 backbone network.
3. Configure a static route between SwitchA and SwitchB so that they can be connected through the IPv4 backbone network.

Procedure

Step 1 Configure SwitchA.

Enable the service loopback function on an Eth-Trunk.

CAUTION

The interface must be idle. That is, the interface does not transmit services.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] interface eth-trunk 1
[SwitchA-Eth-Trunk1] service type tunnel
[SwitchA-Eth-Trunk1] quit
[SwitchA] interface gigabitethernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] eth-trunk 1
[SwitchA-GigabitEthernet1/0/3] quit
```

Configure an IPv4/IPv6 dual stack.

```
[SwitchA] ipv6
[SwitchA] vlan batch 100 200
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchA-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface vlanif 100
[SwitchA-Vlanif100] ip address 2.1.1.1 8
[SwitchA-Vlanif100] quit
[SwitchA] interface gigabitethernet 1/0/2
[SwitchA-GigabitEthernet1/0/2] port hybrid pvid vlan 200
[SwitchA-GigabitEthernet1/0/2] port hybrid untagged vlan 200
[SwitchA-GigabitEthernet1/0/2] quit
[SwitchA] interface vlanif 200
[SwitchA-Vlanif200] ipv6 enable
[SwitchA-Vlanif200] ipv6 address 2002:0201:0101:1::1/64
[SwitchA-Vlanif200] quit
```

Configure a 6to4 tunnel.

```
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] tunnel-protocol ipv6-ipv4 6to4
[SwitchA-Tunnel1] eth-trunk 1
[SwitchA-Tunnel1] ipv6 enable
[SwitchA-Tunnel1] ipv6 address 2002:0201:0101::1/64
[SwitchA-Tunnel1] source vlanif 100
[SwitchA-Tunnel1] quit
```

Configure a route to the other 6to4 network.

```
[SwitchA] ipv6 route-static 2002:: 16 tunnel 1
```

Step 2 Configure SwitchB.

Enable the service loopback function on an Eth-Trunk.

CAUTION

The interface must be idle. That is, the interface does not transmit services.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] interface eth-trunk 1
[SwitchB-Eth-Trunk1] service type tunnel
[SwitchB-Eth-Trunk1] quit
[SwitchB] interface gigabitethernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] eth-trunk 1
[SwitchB-GigabitEthernet1/0/3] quit
```

Configure an IPv4/IPv6 dual stack.

```
[SwitchB] ipv6
[SwitchB] vlan batch 100 200
[SwitchB] interface gigabitethernet1/0/1
[SwitchB-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[SwitchB-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[SwitchB-GigabitEthernet1/0/1] quit
[SwitchB] interface vlanif 100
[SwitchB-Vlanif100] ip address 2.1.1.2 8
[SwitchB-Vlanif100] quit
[SwitchB] interface gigabitethernet1/0/2
[SwitchB-GigabitEthernet1/0/2] port hybrid pvid vlan 200
[SwitchB-GigabitEthernet1/0/2] port hybrid untagged vlan 200
[SwitchB-GigabitEthernet1/0/2] quit
[SwitchB] interface vlanif 200
[SwitchB-Vlanif200] ipv6 enable
[SwitchB-Vlanif200] ipv6 address 2002:0201:0102:1::1/64
[SwitchB-Vlanif200] quit
```

Configure a 6to4 tunnel.

```
[SwitchB] interface tunnel 1
[SwitchB-Tunnel1] eth-trunk 1
[SwitchB-Tunnel1] tunnel-protocol ipv6-ipv4 6to4
[SwitchB-Tunnel1] ipv6 enable
[SwitchB-Tunnel1] ipv6 address 2002:0201:0102::1/64
[SwitchB-Tunnel1] source vlanif 100
[SwitchB-Tunnel1] quit
```

Configure a route to the other 6to4 network.

```
[SwitchB] ipv6 route-static 2002:: 16 tunnel 1
```

NOTE

There must be a reachable route between SwitchA and SwitchB. In this example, a routing protocol needs to be configured on VLANIF 100 of SwitchA and SwitchB. For details, see the *S7700&S9700 Smart&Core Routing Switch Configuration Guide - IP Routing*

Step 3 Verify the configuration.

Check the IPv6 status of Tunnel1 on SwitchA. You can see that the tunnel status is Up.

```
[SwitchA] display ipv6 interface tunnel 1
Tunnel1 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::201:101
  Global unicast address(es):
    2002:201:101::1, subnet is 2002:201:101::/64
  Joined group address(es):
    FF02::1:FF01:101
```

```
    FF02::1:FF00:1
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
```

Ping the 6to4 address of VLANIF200 on SwitchB from SwitchA. The 6to4 address can be pinged successfully.

```
[SwitchA] ping ipv6 2002:0201:0102:1::1
  PING 2002:0201:0102:1::1 : 56  data bytes, press CTRL_C to break
    Reply from 2002:201:102:1::1
    bytes=56 Sequence=1 hop limit=64  time = 8 ms
    Reply from 2002:201:102:1::1
    bytes=56 Sequence=2 hop limit=64  time = 25 ms
    Reply from 2002:201:102:1::1
    bytes=56 Sequence=3 hop limit=64  time = 4 ms
    Reply from 2002:201:102:1::1
    bytes=56 Sequence=4 hop limit=64  time = 5 ms
    Reply from 2002:201:102:1::1
    bytes=56 Sequence=5 hop limit=64  time = 5 ms

  --- 2002:0201:0102:1::1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 4/9/25 ms
```

----End

Configuration Files

- Configuration file of SwitchA

```
#
 sysname SwitchA
#
ipv6
#
vlan batch 100 200
#
interface Vlanif100
 ip address 2.1.1.1 255.0.0.0
#
interface Vlanif200
 ipv6 enable
 ipv6 address 2002:201:101:1::1/64
#
interface Eth-Trunk1
 service type tunnel
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface GigabitEthernet1/0/3
 eth-trunk 1
#
interface Tunnel1
 ipv6 enable
 ipv6 address 2002:201:101::1/64
```

```
 tunnel-protocol ipv6-ipv4 6to4
 source vlanif100
 eth-trunk 1
#
ipv6 route-static 2002:: 16 Tunnel1
#
return
```

- Configuration file of SwitchB

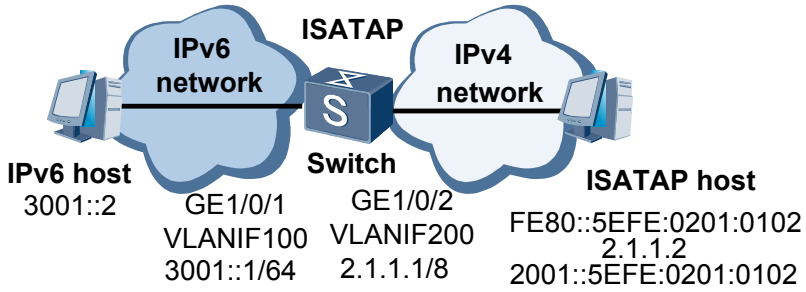
```
#
 sysname SwitchB
#
ipv6
#
vlan batch 100 200
#
interface Vlanif100
 ip address 2.1.1.2 255.0.0.0
#
interface Vlanif200
 ipv6 enable
 ipv6 address 2002:201:102:1::1/64
#
interface Eth-Trunk1
 service type tunnel
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface GigabitEthernet1/0/3
 eth-trunk 1
#
interface Tunnel1
 ipv6 enable
 ipv6 address 2002:201:102::1/64
 tunnel-protocol ipv6-ipv4 6to4
 source vlanif100
 eth-trunk 1
#
ipv6 route-static 2002:: 16 Tunnel1
#
return
```

11.7.3 Example for Configuring an ISATAP Tunnel

Networking Requirements

As shown in **Figure 11-3**, an IPv6 host on the IPv4 network runs Windows XP. The IPv6 host needs to be connected to the IPv6 network through a border device. The IPv6 host and border device support ISATAP. An ISATAP tunnel needs to be set up between the IPv6 host and the border device.

Figure 11-3 Networking diagram for configuring an ISATAP tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IPv4/IPv6 dual stack on the switch so that the switch can access the IPv4 network and IPv6 network.
2. Configure an ISATAP tunnel on the switch so that IPv6 hosts on the IPv4 network can communicate with IPv6 hosts on the IPv6 network.
3. Configure a static route from the IPv6 host to the ISATAP host so that the IPv6 host can forward packets directly over the tunnel.

Procedure

Step 1 Configure the ISATAP border device.

Enable the service loopback function on an Eth-Trunk.

CAUTION

The interface must be idle. That is, the interface does not transmit services.

```
<Quidway> system-view
[Quidway] interface eth-trunk 1
[Quidway-Eth-Trunk1] service type tunnel
[Quidway-Eth-Trunk1] quit
[Quidway] interface gigabitethernet 1/0/3
[Quidway-GigabitEthernet1/0/3] eth-trunk 1
[Quidway-GigabitEthernet1/0/3] quit
```

Enable the IPv4/IPv6 dual stack and configure an IP address for each interface.

```
[Quidway] ipv6
[Quidway] vlan batch 100 200
[Quidway] interface gigabitethernet 1/0/1
[Quidway-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[Quidway-GigabitEthernet1/0/1] port hybrid untagged vlan 100
[Quidway-GigabitEthernet1/0/1] quit
[Quidway] interface gigabitethernet 1/0/2
[Quidway-GigabitEthernet1/0/2] port hybrid pvid vlan 200
[Quidway-GigabitEthernet1/0/2] port hybrid untagged vlan 200
[Quidway-GigabitEthernet1/0/2] quit
```

```
[Quidway] interface vlanif 100
[Quidway-Vlanif100] ipv6 enable
[Quidway-Vlanif100] ipv6 address 3001::1/64
[Quidway-Vlanif100] quit
[Quidway] interface vlanif 200
[Quidway-Vlanif200] ip address 2.1.1.1 255.0.0.0
[Quidway-Vlanif200] quit
```

Configure an ISATAP tunnel.

```
[Quidway] interface tunnel 1
[Quidway-Tunnel1] tunnel-protocol ipv6-ipv4 isatap
[Quidway-Tunnel1] eth-trunk 1
[Quidway-Tunnel1] ipv6 enable
[Quidway-Tunnel1] ipv6 address 2001::/64 eui-64
[Quidway-Tunnel1] source vlanif 200
[Quidway-Tunnel1] undo ipv6 nd ra halt
[Quidway-Tunnel1] quit
```

Step 2 Configure the ISATAP host.

NOTE

The ISATAP host needs to run IPv6 and be enabled with the IPv6 function.

Run the following command to add a static route to the border device. The number of the pseudo interface on the host is 2. You can run the **ipv6 if** command to check the interface corresponding to Automatic Tunneling Pseudo-Interface.

```
C:\> netsh interface ipv6 isatap set router 2.1.1.1
```

Step 3 Configure the IPv6 host.

Configure a static route to the border device on the IPv6 host so that PCs on two different networks can communicate through the ISATAP tunnel.

```
C:\> netsh interface ipv6 set route 2001::/64 3001::1
```

Step 4 Verify the configuration.

Check the IPv6 status of Tunnel1 on the ISATAP device. You can see that the tunnel status is Up.

```
[Quidway] display ipv6 interface tunnel 1
Tunnel1 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::5EFE:201:101
  Global unicast address(es):
    2001::5EFE:201:101, subnet is 2001::/64
  Joined group address(es):
    FF02::1:FF01:101
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisement max interval 600 seconds, min interval 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses
```

Ping the global unicast address of the tunnel interface on the ISATAP host from the ISATAP device.

```
[Quidway] ping ipv6 2001::5efe:2.1.1.2
  PING 2001::5efe:2.1.1.2 : 56  data bytes, press CTRL_C to break
```

```
    Reply from 2001::5EFE:201:102
    bytes=56 Sequence=1 hop limit=64  time = 4 ms
    Reply from 2001::5EFE:201:102
    bytes=56 Sequence=2 hop limit=64  time = 3 ms
    Reply from 2001::5EFE:201:102
    bytes=56 Sequence=3 hop limit=64  time = 2 ms
    Reply from 2001::5EFE:201:102
    bytes=56 Sequence=4 hop limit=64  time = 2 ms
    Reply from 2001::5EFE:201:102
    bytes=56 Sequence=5 hop limit=64  time = 2 ms

  --- 2001::5efe:2.1.1.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 2/2/4 ms
```

Ping the global unicast address of the ISATAP device from the ISATAP host.

```
C:\> ping6 2001::5efe:2.1.1.1

Pinging 2001::5efe:2.1.1.1
from 2001::5efe:2.1.1.2 with 32 bytes of data:

Reply from 2001::5efe:2.1.1.1: bytes=32 time=1ms
Reply from 2001::5efe:2.1.1.1: bytes=32 time=1ms
Reply from 2001::5efe:2.1.1.1: bytes=32 time=1ms
Reply from 2001::5efe:2.1.1.1: bytes=32 time=1ms
Ping statistics for 2001::5efe:2.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Ping the IPv6 host from the ISATAP host. They can ping each other.

```
C:\> ping6 3001::2

Pinging 3001::2 with 32 bytes of data:

Reply from 3001::2: time<1ms
Reply from 3001::2: time<1ms
Reply from 3001::2: time<1ms
Reply from 3001::2: time<1ms

Ping statistics for 3001::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

----End

Configuration Files

Configuration file of the Switch

```
#
 sysname Quidway
#
vlan batch 100 200
#
ipv6
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3001::1/64
#
```

```
interface Vlanif200
 ip address 2.1.1.1 255.0.0.0
#
interface Eth-Trunk1
 service type tunnel
#
interface Tunnel1
 ipv6 enable
 ipv6 address 2001::/64 eui-64
 undo ipv6 nd ra halt
 tunnel-protocol ipv6-ipv4 isatap
 source Vlanif200
 eth-trunk 1
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface GigabitEthernet1/0/3
 eth-trunk 1
#
return
```

11.7.4 Example for Configuring 6PE

Networking Requirements

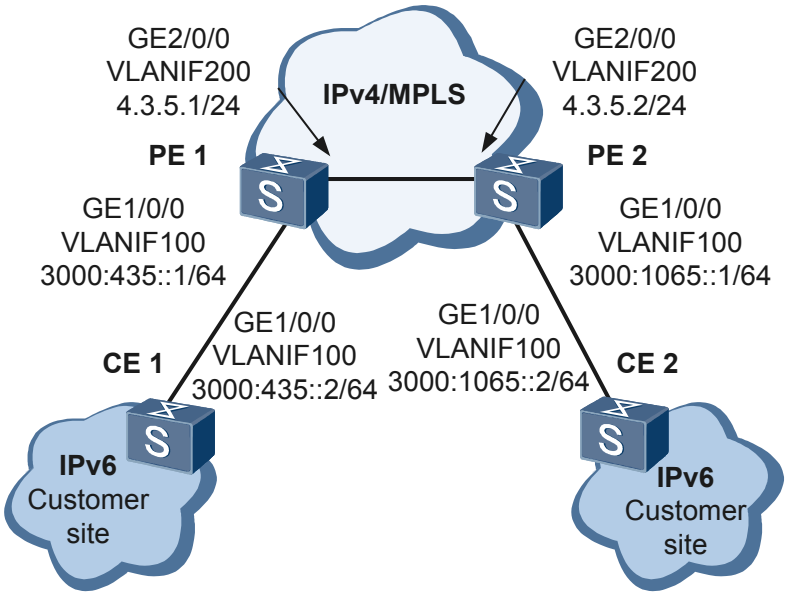
NOTE

To run MPLS-related commands, you need to purchase a license.

As shown in **Figure 11-4**, PE1 and PE2 support 6PE, and CE1 and CE2 support the IPv6 protocol. A carrier's IPv4/MPLS backbone network exists between PE1 and PE2. The IPv4/MPLS backbone network runs OSPF. IPv4 IBGP connections are set up between PEs. CE1 and CE2 are located on two IPv6 networks respectively. PEs and CEs use IPv6 addresses to exchange route information through static routes.

It is required that 6PE be used to connect IPv6 networks through the IPv4/MPLS backbone network.

Figure 11-4 Networking diagram for configuring the 6PE



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IPv6 and configure an IPv4/IPv6 dual stack.
2. Enable MPLS.
3. Configure a 6PE peer.
4. Configure IPv6 addresses for CE interfaces and static routes.

Configuration Procedures

1. Enable IPv6 and configure an IPv4/IPv6 dual stack.

   # Enable IPv6 on PE1.

   ```
   <Quidway> system-view
   [Quidway] sysname PE1
   [PE1] ipv6
   ```

   # Enable IPv6 on PE2.

   ```
   <Quidway> system-view
   [Quidway] sysname PE2
   [PE2] ipv6
   ```

   # Configure an IPv6 address for VLANIF 100 on PE1 and an IP address for Loopback0.

   ```
   [PE1] vlan batch 100 200
   [PE1] interface gigabitethernet 1/0/0
   [PE1-GigabitEthernet1/0/0] port hybrid pvid vlan 100
   [PE1-GigabitEthernet1/0/0] port hybrid untagged vlan 100
   [PE1-GigabitEthernet1/0/0] quit
   [PE1] interface vlanif 100
   [PE1-Vlanif100] ipv6 enable
   [PE1-Vlanif100] ipv6 address 3000:435::1 64
   [PE1-Vlanif100] quit
   [PE1] interface loopback 0
   ```

```
[PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack0] quit
```

Configure an IPv6 address for VLANIF 100 on PE2 and an IP address for Loopback0.

```
[PE2] vlan batch 100 200
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[PE2-GigabitEthernet1/0/0] port hybrid untagged vlan 100
[PE2-GigabitEthernet1/0/0] quit
[PE2] interface vlanif 100
[PE2-Vlanif100] ipv6 enable
[PE2-Vlanif100] ipv6 address 3000:1065::1 64
[PE2-Vlanif100] quit
[PE2] interface loopback 0
[PE2-LoopBack0] ip address 2.2.2.9 255.255.255.255
[PE2-LoopBack0] quit
```

2. Enable MPLS.

NOTE

PEs in this example are directly connected. You need to run the **label advertise** command to enable the egress node to assign labels to the penultimate hop.

Configure an IP address for VLANIF 200 on PE1 and enable MPLS and LDP.

```
[PE1] mpls lsr-id 1.1.1.9
[PE1] mpls
[PE1-mpls] lsp-trigger all
[PE1-mpls] label advertise non-null
[PE1-mpls] quit
[PE1] mpls ldp
[PE1-mpls-ldp] quit
[PE1] interface gigabitethernet 2/0/0
[PE1-GigabitEthernet2/0/0] port hybrid pvid vlan 200
[PE1-GigabitEthernet2/0/0] port hybrid untagged vlan 200
[PE1-GigabitEthernet2/0/0] quit
[PE1] interface vlanif 200
[PE1-Vlanif200] ip address 4.3.5.1 255.255.255.0
[PE1-Vlanif200] mpls
[PE1-Vlanif200] mpls ldp
[PE1-Vlanif200] quit
```

Configure an IP address for VLANIF 200 on PE2 and enable MPLS and LDP.

```
[PE2] mpls lsr-id 2.2.2.9
[PE2] mpls
[PE2-mpls] lsp-trigger all
[PE2-mpls] label advertise non-null
[PE2-mpls] quit
[PE2] mpls ldp
[PE2-mpls-ldp] quit
[PE2] interface gigabitethernet 2/0/0
[PE2-GigabitEthernet2/0/0] port hybrid pvid vlan 200
[PE2-GigabitEthernet2/0/0] port hybrid untagged vlan 200
[PE2-GigabitEthernet2/0/0] quit
[PE2] interface vlanif 200
[PE2-Vlanif200] ip address 4.3.5.2 255.255.255.0
[PE2-Vlanif200] mpls
[PE2-Vlanif200] mpls ldp
[PE2-Vlanif200] quit
```

Configure OSPF on PE1 to trigger LSP setup.

```
[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 4.3.5.0 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

Configure OSPF on PE2 to trigger LSP setup.

```
[PE2] ospf
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] network 2.2.2.9 0.0.0.0
[PE2-ospf-1-area-0.0.0.0] network 4.3.5.0 0.0.0.255
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
```

3. Configure a 6PE peer.

   # Configure IBGP on PE1, enable 6PE capability for the peer, and import IPv6 direct and static routes.

```
[PE1] bgp 65100
[PE1-bgp] peer 2.2.2.9 as-number 65100
[PE1-bgp] peer 2.2.2.9 connect-interface loopback 0
[PE1-bgp] ipv6-family
[PE1-bgp-af-ipv6] import-route direct
[PE1-bgp-af-ipv6] import-route static
[PE1-bgp-af-ipv6] peer 2.2.2.9 enable
[PE1-bgp-af-ipv6] peer 2.2.2.9 label-route-capability
[PE1-bgp-af-ipv6] quit
[PE1-bgp] quit
```

   # Configure IBGP on PE2, enable 6PE capability for the peer, and import IPv6 direct and static routes.

```
[PE2] bgp 65100
[PE2-bgp] peer 1.1.1.9 as-number 65100
[PE2-bgp] peer 1.1.1.9 connect-interface loopback 0
[PE2-bgp] ipv6-family
[PE2-bgp-af-ipv6] import-route direct
[PE2-bgp-af-ipv6] import-route static
[PE2-bgp-af-ipv6] peer 1.1.1.9 enable
[PE2-bgp-af-ipv6] peer 1.1.1.9 label-route-capability
[PE2-bgp-af-ipv6] quit
[PE2-bgp] quit
```

4. Configure IPv6 addresses for CE interfaces and static routes.

   # Configure CE1 to set up an IPv6 connection with PE1.

```
<Quidway> system-view
[Quidway] sysname CE1
[CE1] ipv6
[CE1] vlan batch 100
[CE1] interface gigabitethernet 1/0/0
[CE1-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[CE1-GigabitEthernet1/0/0] port hybrid untagged vlan 100
[CE1-GigabitEthernet1/0/0] quit
[CE1] interface vlanif 100
[CE1-Vlanif100] ipv6 enable
[CE1-Vlanif100] ipv6 address 3000:435::2 64
[CE1-Vlanif100] quit
[CE1] ipv6 route-static :: 0 vlanif 100
```

   # Configure CE2 to set up an IPv6 connection with PE2.

```
<Quidway> system-view
[Quidway] sysname CE2
[CE2] ipv6
[CE2] vlan batch 100
[CE2] interface gigabitethernet 1/0/0
[CE2-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[CE2-GigabitEthernet1/0/0] port hybrid untagged vlan 100
[CE2-GigabitEthernet1/0/0] quit
[CE2] interface vlanif 100
[CE2-Vlanif100] ipv6 enable
[CE2-Vlanif100] ipv6 address 3000:1065::2 64
```

```
[CE2-Vlanif100] quit
[CE2] ipv6 route-static :: 0 vlanif 100
```

5. Verify the configuration.

   # View LSP information on PE1.

```
[PE1] display mpls lsp
------------------------------------------------------------
                LSP Information: LDP LSP
------------------------------------------------------------
FEC             In/Out Label  In/Out IF                 Vrf Name
2.2.2.9/32      NULL/3        -/Vlanif200
------------------------------------------------------------
                LSP Information: BGP IPV6 LSP
------------------------------------------------------------
  FEC                 :  3000:435::/64
  In Label            :  109568             Out Label    :  -----
  In Interface        :  -----              OutInterface :  -----
  Vrf Name            :
```

   # View IPv6 routing information on PE1.

```
[PE1] display bgp ipv6 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: 5
 *>  Network  : ::1                                      PrefixLen : 128
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : ?

 *>  Network  : 3000:435::                               PrefixLen : 64
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    : NULL/109568
     Path/Ogn : ?

 *>  Network  : 3000:435::1                              PrefixLen : 128
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : ?

 *>i Network  : 3000:1065::                              PrefixLen : 64
     NextHop  : ::FFFF:2.2.2.9                           LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    : 109568/NULL
     Path/Ogn : ?

 *>  Network  : FE80::                                   PrefixLen : 10
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : ?
```

   # Ping the IPv6 address of CE2 from CE1, and the IPv6 address of CE2 can be pinged successfully.

```
[CE1] ping ipv6 3000:1065::2
PING 3000:1065::2 : 56  data bytes, press CTRL_C to break
    Reply from 3000:1065::2
    bytes=56 Sequence=1 hop limit=63  time = 50 ms
    Reply from 3000:1065::2
    bytes=56 Sequence=2 hop limit=63  time = 1 ms
```

```
    Reply from 3000:1065::2
    bytes=56 Sequence=3 hop limit=63  time = 1 ms
    Reply from 3000:1065::2
    bytes=56 Sequence=4 hop limit=63  time = 1 ms
    Reply from 3000:1065::2
    bytes=56 Sequence=5 hop limit=63  time = 1 ms

  --- 3000:1065::2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/10/50 ms
```

Configuration Files

- Configuration file of PE1

```
#
 sysname PE1
#
 ipv6
#
vlan batch 100 200
#
mpls lsr-id 1.1.1.9
 mpls
  lsp-trigger all
  label advertise non-null
#
mpls ldp
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3000:435::1/64
#
interface Vlanif200
 ip address 4.3.5.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet2/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface LoopBack0
 ip address 1.1.1.9 255.255.255.255
#
bgp 65100
 peer 2.2.2.9 as-number 65100
 peer 2.2.2.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  peer 2.2.2.9 enable
 #
 ipv6-family
  import-route direct
  import-route static
  peer 2.2.2.9 enable
  peer 2.2.2.9 label-route-capability
#
ospf 1
 area 0.0.0.0
  network 1.1.1.9 0.0.0.0
```

```
  network 4.3.5.0 0.0.0.255
#
return
```

- Configuration file of PE2

```
#
 sysname PE2
#
 ipv6
#
vlan batch 100 200
#
mpls lsr-id 2.2.2.9
 mpls
  lsp-trigger all
  label advertise non-null
#
mpls ldp
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3000:1065::1/64
#
interface Vlanif200
 ip address 4.3.5.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet2/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
#
bgp 65100
 peer 1.1.1.9 as-number 65100
 peer 1.1.1.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.9 enable
 #
 ipv6-family
  undo synchronization
  import-route direct
  import-route static
  peer 1.1.1.9 enable
  peer 1.1.1.9 label-route-capability
#
ospf 1
 area 0.0.0.0
  network 2.2.2.9 0.0.0.0
  network 4.3.5.0 0.0.0.255
#
return
```

- Configuration file of CE1

```
#
 sysname CE1
#
 ipv6
#
```

```
vlan batch 100
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3000:435::2 64
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
ipv6 route-static :: 0 Vlanif100
#
return
```

- Configuration file of CE2

```
#
 sysname CE2
#
 ipv6
#
interface Vlanif100
 ipv6 enable
 ipv6 address 3000:1065::2 64
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
ipv6 route-static :: 0 Vlanif100
#
return
```

12 IPv4 over IPv6 Tunnel Configuration

About This Chapter

During the later stage of IPv4 to IPv6 transition, the IPv4 over IPv6 tunnel is used to connect isolated IPv4 sites.

12.1 IPv4 over IPv6 Overview
You can create a tunnel on an IPv6 network to connect isolated IPv4 sites so that isolated IPv4 sites can access other IPv4 networks through the IPv6 public network.

12.2 Configuring an IPv4 over IPv6 Tunnel
To establish IPv4 over IPv6 tunnels, you need to enable the IPv4/IPv6 dual stack on border devices to forward IPv4 packets with the IPv6 header.

12.3 Maintaining the IPv4 over IPv6 Tunnel
Maintaining the IPv4 over IPv6 tunnel includes monitoring the running status of the IPv4 over IPv6 tunnel.

12.4 Configuration Examples
This section provides configuration examples of the IPv4 over IPv6 tunnel.

12.1 IPv4 over IPv6 Overview

You can create a tunnel on an IPv6 network to connect isolated IPv4 sites so that isolated IPv4 sites can access other IPv4 networks through the IPv6 public network.

During the later stage of IPv4 to IPv6 transition, a large number of IPv6 networks have been deployed and isolated IPv4 sites may exist. You can create a tunnel on an IPv6 network to connect isolated IPv4 sites, which is similar to deploying the VPN on the IP network using tunnel technology. The tunnel connecting IPv4 isolated sites on the IPv6 network is called an IPv4 over IPv6 tunnel.

12.2 Configuring an IPv4 over IPv6 Tunnel

To establish IPv4 over IPv6 tunnels, you need to enable the IPv4/IPv6 dual stack on border devices to forward IPv4 packets with the IPv6 header.

Pre-configuration Tasks

Before configuring an IPv4 over IPv6 tunnel, complete the following task:

- Configuring the IPv4/IPv6 dual stack

12.2.1 Enabling the Service Loopback Function on an Eth-Trunk

Context

Before enabling the service loopback function on an Eth-Trunk, pay attention to the following points:

- Ensure that an Eth-Trunk has been created, and member interfaces are added to the Eth-Trunk and in Up state.
- Only one interface enabled with the service loopback function is needed on a device.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface eth-trunk trunk-id
```

The Eth-Trunk interface view is displayed.

Step 3 Run:

```
service type tunnel
```

The Eth-Trunk is enabled with the service loopback function.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface interface-type interface-number
```

The interface view is displayed.

Step 6 Run:

```
eth-trunk trunk-id
```

The interface is added to the Eth-Trunk.

----End

12.2.2 Configuring a Tunnel Interface

Context

Configuring a tunnel interface includes configuring the protocol type, source address, and destination address and IP address.

NOTE

The device does not support fragmentation of packets that are transmitted over the IPv4 over IPv6 tunnel. Therefore, the IPv4 MTU of the tunnel interface must meet the following conditions:

IPv4 MTU of the tunnel interface < IPv6 MTU of the physical interface - Header length of IPv6 packets on the IPv4 over IPv6 tunnel

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface tunnel interface-number
```

A tunnel interface is created.

Step 3 Run:

```
tunnel-protocol ipv4-ipv6
```

The tunnel type is set to IPv4 over IPv6.

Step 4 Run:

```
source { source-ip-address | interface-type interface-number }
```

A source IPv6 address or source interface is configured.

Step 5 Run:

```
destination dest-ip-address
```

The destination address is configured.

Step 6 Run the following commands as required.

- Run:

  ```
  ip address ip-address { mask | mask-length } [ sub ]
  ```

An IPv4 address is configured for the tunnel interface.

- Run:

  ip address unnumbered interface *interface-type interface-number*

  The tunnel interface is configured to borrow an IPv4 address.

----End

12.2.3 Configuring a Tunnel Route

Context

Packets can be forwarded correctly only when devices at the two ends of a tunnel are configured with forwarding routes. Perform the following configurations on devices at the two ends of the tunnel.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Use either of the following methods to configure routes passing through a tunnel interface.

- Run:

  ip route-static *ip-address* { *mask* | *mask-length* } **tunnel** *interface-number*

  A static route is configured.

  The static route must be configured on both ends of the tunnel. The destination address is the destination IPv4 address of the packets that are not encapsulated into IPv6 packets, and the next hop is the local tunnel interface.

- Configure a dynamic route. Dynamic routes can be configured using IGP (excluding IS-IS) or BGP. The configuration method is not mentioned here.

  When configuring a dynamic routing protocol, enable the protocol on the tunnel interface and the interface on the link connecting the IPv4 network and IPv6 network.

----End

12.2.4 Performing Other IPv4 over IPv6 Tunnel Configurations

Context

You can perform one or more following configurations to optimize IPv4 over IPv6 tunnel performance.

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:

tunnel ipv4-ipv6 encapsulation-limit *encapsulation-limit*

The maximum encapsulation count of an IPv6 packet is specified.

By default, an IPv4-over-IPv6 packet can be encapsulated four times.

Step 4 Run:

tunnel ipv4-ipv6 flow-label *label-value*

The flow label value is set.

By default, the flow label value is 0.

Step 5 Run:

tunnel ipv4-ipv6 traffic-class { **original** | *class-value* }

The traffic class is set.

By default, the traffic class is 0.

----End

12.2.5 Checking the Configuration

Prerequisites

All configurations of the IPv4 over IPv6 tunnel are complete.

Procedure

- Run the **display interface tunnel** [*interface-number*] command to check the running status of the tunnel interface.
- Run the **display ip routing-table** command to check the routing table.

----End

12.3 Maintaining the IPv4 over IPv6 Tunnel

Maintaining the IPv4 over IPv6 tunnel includes monitoring the running status of the IPv4 over IPv6 tunnel.

12.3.1 Monitoring the Running Status of the IPv4 over IPv6 Tunnel

Context

In routine maintenance, you can run the following commands in any view to monitor the running status of the IPv4 over IPv6 tunnel.

Procedure

- Run the **display interface tunnel** [*interface-number*] command in any view to view the running status of the tunnel interface.

----End

12.4 Configuration Examples

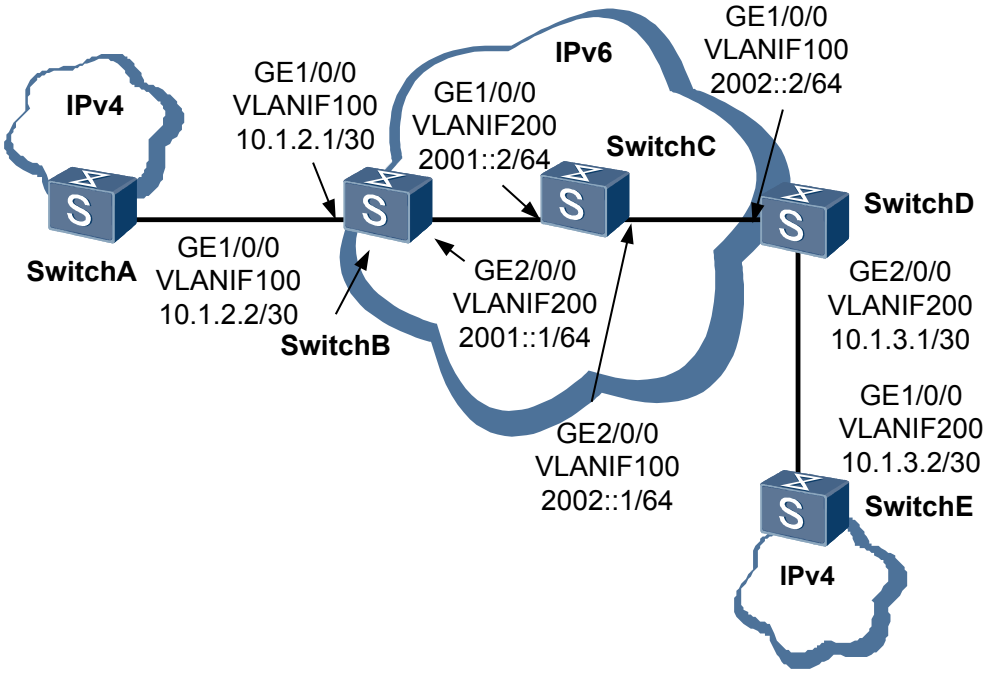
This section provides configuration examples of the IPv4 over IPv6 tunnel.

12.4.1 Example for Configuring an IPv4 over IPv6 Tunnel

Networking Requirements

As shown in **Figure 12-1**, two IPv4 networks are connected to an IPv6 network through SwitchA and SwitchE. Border devices SwitchB and SwitchD on the IPv6 network support the IPv4/IPv6 dual stack. An IPv4 over IPv6 tunnel needs to be set up between SwitchB and SwitchD so that physically isolated IPv4 networks can communicate.

Figure 12-1 Networking diagram for configuring an IPv4 over IPv6 tunnel



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IPv4 over IPv6 tunnel on the border devices at both ends of the IPv6 network.
2. Configure a route for the tunnel interface to forward packets.

Configuration Procedures

1. Configure an IPv6 address for the physical interface and enable IPv6 capability for IS-IS on the IPv6 network to implement IP connectivity of the IPv6 network.

   # Configure SwitchB.

```
<Quidway> system-view
[Quidway] sysname SwitchB
[SwitchB] ipv6
[SwitchB] vlan batch 100 200
[SwitchB] interface gigabitethernet 2/0/0
[SwitchB-GigabitEthernet2/0/0] port hybrid pvid vlan 200
[SwitchB-GigabitEthernet2/0/0] port hybrid untagged vlan 200
[SwitchB-GigabitEthernet2/0/0] quit
[SwitchB] interface vlanif 200
[SwitchB-Vlanif200] ipv6 enable
[SwitchB-Vlanif200] ipv6 address 2001::1 64
[SwitchB-Vlanif200] quit
[SwitchB] isis 1
[SwitchB-isis-1] network-entity 10.0000.0000.0001.00
[SwitchB-isis-1] ipv6 enable topology standard
[SwitchB-isis-1] quit
[SwitchB] interface vlanif 200
[SwitchB-Vlanif200] isis ipv6 enable 1
[SwitchB-Vlanif200] quit
```

   # Configure SwitchC.

```
<Quidway> system-view
[Quidway] sysname SwitchC
[SwitchC] ipv6
[SwitchC] vlan batch 100 200
[SwitchC] interface gigabitethernet 1/0/0
[SwitchC-GigabitEthernet1/0/0] port hybrid pvid vlan 200
[SwitchC-GigabitEthernet1/0/0] port hybrid untagged vlan 200
[SwitchC-GigabitEthernet1/0/0] quit
[SwitchC] interface gigabitethernet 2/0/0
[SwitchC-GigabitEthernet2/0/0] port hybrid pvid vlan 100
[SwitchC-GigabitEthernet2/0/0] port hybrid untagged vlan 100
[SwitchC-GigabitEthernet2/0/0] quit
[SwitchC] interface vlanif 200
[SwitchC-Vlanif200] ipv6 enable
[SwitchC-Vlanif200] ipv6 address 2001::2 64
[SwitchC-Vlanif200] quit
[SwitchC] interface vlanif 100
[SwitchC-Vlanif100] ipv6 enable
[SwitchC-Vlanif100] ipv6 address 2002::1 64
[SwitchC-Vlanif100] quit
[SwitchC] isis 1
[SwitchC-isis-1] network-entity 10.0000.0000.0002.00
[SwitchC-isis-1] ipv6 enable topology standard
[SwitchC-isis-1] quit
[SwitchC] interface vlanif 100
[SwitchC-Vlanif100] isis ipv6 enable 1
[SwitchC-Vlanif100] quit
[SwitchC] interface vlanif 200
[SwitchC-Vlanif200] isis ipv6 enable 1
[SwitchC-Vlanif200] quit
```

   # Configure SwitchD.

```
<Quidway> system-view
[Quidway] sysname SwitchD
[SwitchD] ipv6
[SwitchD] vlan batch 100 200
[SwitchD] interface gigabitethernet 1/0/0
[SwitchD-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[SwitchD-GigabitEthernet1/0/0] port hybrid untagged vlan 100
```

```
[SwitchD-GigabitEthernet1/0/0] quit
[SwitchD] interface vlanif 100
[SwitchD-Vlanif100] ipv6 enable
[SwitchD-Vlanif100] ipv6 address 2002::2 64
[SwitchD-Vlanif100] quit
[SwitchD] isis 1
[SwitchD-isis-1] network-entity 10.0000.0000.0003.00
[SwitchD-isis-1] ipv6 enable topology standard
[SwitchD-isis-1] quit
[SwitchD] interface vlanif 100
[SwitchD-Vlanif100] isis ipv6 enable 1
[SwitchD-Vlanif100] quit
```

2. Configure an IPv4 address for the physical interface, and configure OSPF on the IPv4 network to implement IP connectivity of the IPv4 network.

   # Configure SwitchA.

```
<Quidway> system-view
[Quidway] sysname SwitchA
[SwitchA] vlan batch 100
[SwitchA] interface gigabitethernet 1/0/0
[SwitchA-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[SwitchA-GigabitEthernet1/0/0] port hybrid untagged vlan 100
[SwitchA-GigabitEthernet1/0/0] quit
[SwitchA] interface vlanif 100
[SwitchA-Vlanif100] ip address 10.1.2.2 30
[SwitchA-Vlanif100] quit
[SwitchA] ospf 1
[SwitchA-ospf-1] area 0
[SwitchA-ospf-1-area-0.0.0.0] network 10.1.2.0 0.0.0.3
[SwitchA-ospf-1-area-0.0.0.0] quit
[SwitchA-ospf-1] quit
```

   # Configure SwitchB.

```
[SwitchB] interface gigabitethernet 1/0/0
[SwitchB-GigabitEthernet1/0/0] port hybrid pvid vlan 100
[SwitchB-GigabitEthernet1/0/0] port hybrid untagged vlan 100
[SwitchB-GigabitEthernet1/0/0] quit
[SwitchB] interface vlanif 100
[SwitchB-Vlanif100] ip address 10.1.2.1 30
[SwitchB-Vlanif100] quit
[SwitchB] ospf 1
[SwitchB-ospf-1] area 0
[SwitchB-ospf-1-area-0.0.0.0] network 10.1.2.0 0.0.0.3
[SwitchB-ospf-1-area-0.0.0.0] quit
[SwitchB-ospf-1] quit
```

   # Configure SwitchD.

```
[SwitchD] interface gigabitethernet 2/0/0
[SwitchD-GigabitEthernet2/0/0] port hybrid pvid vlan 200
[SwitchD-GigabitEthernet2/0/0] port hybrid untagged vlan 200
[SwitchD-GigabitEthernet2/0/0] quit
[SwitchD] interface vlanif 200
[SwitchD-Vlanif200] ip address 10.1.3.1 30
[SwitchD-Vlanif200] quit
[SwitchD] ospf 1
[SwitchD-ospf-1] area 0
[SwitchD-ospf-1-area-0.0.0.0] network 10.1.3.0 0.0.0.3
[SwitchD-ospf-1-area-0.0.0.0] quit
[SwitchD-ospf-1] quit
```

   # Configure SwitchE.

```
<Quidway> system-view
[Quidway] sysname SwitchE
[SwitchE] vlan batch 200
[SwitchE] interface gigabitethernet 1/0/0
[SwitchE-GigabitEthernet1/0/0] port hybrid pvid vlan 200
```

```
[SwitchE-GigabitEthernet1/0/0] port hybrid untagged vlan 200
[SwitchE-GigabitEthernet1/0/0] quit
[SwitchE] interface vlanif 200
[SwitchE-Vlanif200] ip address 10.1.3.2 30
[SwitchE-Vlanif200] quit
[SwitchE] ospf 1
[SwitchE-ospf-1] area 0
[SwitchE-ospf-1-area-0.0.0.0] network 10.1.3.0 0.0.0.3
[SwitchE-ospf-1-area-0.0.0.0] quit
[SwitchE-ospf-1] quit
```

3. Configure a tunnel interface.

   # Create a tunnel interface and configure an IPv4 address, a source IPv6 address (or source interface), and a destination IPv6 address for the tunnel interface.

   # Configure SwitchB.

```
[SwitchB] interface tunnel 2
[SwitchB-Tunnel2] tunnel-protocol ipv4-ipv6
[SwitchB-Tunnel2] ip address 10.1.1.1 30
[SwitchB-Tunnel2] source vlanif 200
[SwitchB-Tunnel2] destination 2002::2
[SwitchB-Tunnel2] quit
```

   # Configure SwitchD.

```
[SwitchD] interface tunnel 1
[SwitchD-Tunnel1] tunnel-protocol ipv4-ipv6
[SwitchD-Tunnel1] ip address 10.1.1.2 30
[SwitchD-Tunnel1] source vlanif 100
[SwitchD-Tunnel1] destination 2001::1
[SwitchD-Tunnel1] quit
```

4. Configure static routes.

   # Configure a static route between SwitchA and SwitchE.

   # Configure SwitchA.

```
[SwitchA] ip route-static 10.1.3.2 255.255.255.252 vlanif 100 10.1.2.1
```

   # Configure SwitchE.

```
[SwitchE] ip route-static 10.1.2.2 255.255.255.252 vlanif 200 10.1.3.1
```

   # Configure static routes between both ends of the tunnel.

   # Configure SwitchB.

```
[SwitchB] ip route-static 10.1.3.2 255.255.255.252 tunnel 1
```

   # Configure SwitchD.

```
[SwitchD] ip route-static 10.1.2.2 255.255.255.252 tunnel 1
```

5. Verify the configuration.

   After the preceding configurations are complete, check the status of the tunnel interface on SwitchB and SwitchD. You can see that the protocol status of the tunnel interface is Up.

```
[SwitchB] display interface tunnel 2
Tunnel2 current state : UP
Line protocol current state : UP
Last line protocol up time : 2012-02-28 02:46:05
Description :
HUAWEI, Quidway Series, Tunnel2 Interface
Route Port,The Maximum Transmit Unit is 1500
Internet Address is 10.1.1.1/30
Encapsulation is TUNNEL6, loopback not set
Tunnel protocol/transport (IPv6 or IPV4) over IPv6
Tunnel Source 2001::1 (Vlanif200)
Tunnel Destination 2002::2
Tunnel Encapsulation limit 4
Tunnel Traffic class not set
```

```
Tunnel Flow label not set
QoS maxBandwidth : 64 Kbps
Output queue : (Urgent queue : Size/Length/Discards)  0/50/0
Output queue : (Protocol queue : Size/Length/Discards) 0/1000/0
Output queue : (FIFO queue : Size/Length/Discards)  0/256/0
    5 minutes input rate 10 bits/sec, 0 packets/sec
    5 minutes output rate 14 bits/sec, 0 packets/sec
    493 packets input,  38480 bytes
    0 input error
    447 packets output,  53144 bytes
    0 output error
```

Check the IPv4 routing table on SwitchB and SwitchD. You can see that the outbound interface of the route to the remote IPv4 network is a tunnel interface.

```
[SwitchB] 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       InLoopBack0
       10.1.1.0/30  Direct 0    0            D   10.1.1.1        Tunnel2
       10.1.1.1/32  Direct 0    0            D   127.0.0.1       InLoopBack0
       10.1.2.0/30  Direct 0    0            D   10.1.2.1        Vlanif100
       10.1.2.1/32  Direct 0    0            D   127.0.0.1       InLoopBack0
       10.1.2.2/32  Direct 0    0            D   10.1.2.2        Vlanif100
      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
```

Ping VLANIF200 on SwitchE from SwitchA, SwitchA can receive a Reply packet from SwitchE.

```
[SwitchA] ping 10.1.3.2
  PING 10.1.3.2: 56  data bytes, press CTRL_C to break
    Reply from 10.1.3.2: bytes=56 Sequence=1 ttl=254 time=20 ms
    Reply from 10.1.3.2: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.1.3.2: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.1.3.2: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.1.3.2: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.1.3.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/4/20 ms
```

Configuration Files

- Configuration file of SwitchA

```
#
 sysname SwitchA
#
vlan batch 100
#
interface Vlanif100
 ip address 10.1.2.2 255.255.255.252
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
 ip route-static 10.1.3.2 255.255.255.252 Vlanif100 10.1.2.1
#
ospf 1
 area 0.0.0.0
  network 10.1.2.0 0.0.0.3
```

```
#
return
```

- Configuration file of SwitchB

```
#
 sysname SwitchB
#
 ipv6
#
vlan batch 100 200
#
isis 1
 network-entity 10.0000.0000.0001.00
 ipv6 enable topology standard
#
interface Vlanif100
 ip address 10.1.2.1 255.255.255.252
#
interface Vlanif200
 ipv6 enable
 ipv6 address 2001::1/64
 isis ipv6 enable 1
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet2/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface Tunnel2
 ip address 10.1.1.1 255.255.255.252
 tunnel-protocol ipv4-ipv6
 source  Vlanif200
 destination  2002::2
#
 ip route-static 10.1.3.2 255.255.255.252 Tunnel1
#
ospf 1
 area 0.0.0.0
  network 10.1.2.0 0.0.0.3
#
return
```

- Configuration file of SwitchC

```
#
 sysname SwitchC
#
 ipv6
#
vlan batch 100 200
#
isis 1
 network-entity 10.0000.0000.0002.00
 ipv6 enable topology standard
#
interface Vlanif100
 ivp6 enable
 ipv6 address 2002::1/64
 isis ipv6 enable 1
#
interface Vlanif200
 ivp6 enable
 ipv6 address 2001::2/64
 isis ipv6 enable 1
#
```

```
inteface GigabitEthernet1/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface GigabitEthernet2/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
return
```

- Configuration file of SwitchD

```
#
 sysname SwitchD
#
 ipv6
#
vlan batch 100 200
#
isis 1
 network-entity 10.0000.0000.0003.00
 #
  ipv6 enable topology standard
 #
#
interface Vlanif100
 ipv6 enable
 ipv6 address 2002::2/64
 isis ipv6 enable 1
#
interface Vlanif200
 ip address 10.1.3.1 255.255.255.252
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 100
 port hybrid untagged vlan 100
#
interface GigabitEthernet2/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
interface Tunnel1
 ip address 10.1.1.2 255.255.255.252
 tunnel-protocol ipv4-ipv6
 source  Vlanif100
 destination  2001::1
#
 ip route-static 10.1.2.2 255.255.255.252 Tunnel1
#
ospf 1
 area 0.0.0.0
  network 10.1.3.0 0.0.0.3
#
return
```

- Configuration file of SwitchE

```
#
 sysname SwitchE
#
vlan batch 200
#
interface Vlanif200
 ip address 10.1.3.2 255.255.255.252
#
interface GigabitEthernet1/0/0
 port hybrid pvid vlan 200
 port hybrid untagged vlan 200
#
```

```
 ip route-static 10.1.2.2 255.255.255.252 Vlanif200 10.1.3.1
#
ospf 1
 area 0.0.0.0
  network 10.1.3.0 0.0.0.3
#
return
```