# IP Protocol and Services

## QSR-1920, QSR-2920, QSR-3920

# Оглавление

# 1. FORWARDING

## 1.1. Overview

To improve the service packet forwarding performance for the data communication device, a fast forwarding technology is designed. It shifts down and completes the routing and service handling in the layer compression mode at a time. This greatly reduces resource consumption caused by internal system task switching and packet cache management and finally improves the data forwarding performance for the overall system. This technology is a data forwarding software platform that is extensible and irrelevant to the hardware platform. It not only supports the data forwarding software platform of different products, but also supports fast forwarding of data packets, such as IPv4 unicast, IPv4 multicast, IPv6, and MPLS. These software platforms are collectively called as the fast forwarding platform.

In the multi-core distribution system, service data will be balanced to multiple cores based on the multi-core load balancing policy. This improves the system data forwarding capability. There are two CPU multi-core load balancing policies: load policy per data flow and load policy per packet.

## 1.2. Forwarding Function Configuration

Table 1-1 Forwarding function list

| Configuration Task | |
|---|---|
| Configure the load balancing policy per packet | Configure the load balancing policy per packet |
| Configure the load balancing policy per data flow | Configure the load balancing policy per data flow |
| Configure the auto flow forwarding mode | Configure the auto flow forwarding mode |
| Configure the forced flow forwarding mode | Configure the forced flow forwarding mode |

### 1.2.1. **Configure Load Balancing Policy per Packet**

#### Configuration Condition

None

#### Configure Load Balancing Policy per Packet

Load policy per packet: Each core takes turns to handle a packet every time. This method cannot ensure the sequence of the service data flow, but it can ensure the CPU multi-core load balancing.

Table 1-2 Configure the load balancing policy per packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the load balancing policy per packet | **forwarding policy per-packet** | Optional<br>By default, the load balancing is performed per packet. |

## 1.2.2. Configure Load Balancing Policy per Data Flow

### Configuration Condition

None

### Configure Fast Forwarding Function over Interface

Load policy per data flow: Load is balanced by data flow. Different traffic distribution methods are adopted for service data flow of different types. For example, the IPv4 and IPv6 service data packet adopts IP quintuple as the key value for hash traffic distribution. The MPLS service data packet adopts the MPLS label as the key value for hash traffic distribution. Where, a data flow will be forwarded and handled only on a CPU core. This method can ensure the continuity of the same flow packet processing.

Table 1-3 Configure the load balancing policy per data flow

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the load balancing policy per data flow | **forwarding policy per-flow** | Optional<br>By default, the load balancing is performed per data flow. |

## 1.2.3. Configure Auto Flow Forwarding Mode

### Configuration Condition

None

### Configure Auto Flow Forwarding Mode

The service using flow acceleration (such as ACL) is enabled on the interface, and the shortest path forwarding packet uses flow acceleration.

Table 1-4 Configure the auto flow forwarding mode

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the auto flow forwarding mode | **forwarding flow auto-mode** | Optional<br>By default, the shortest path uses the forced flow forwarding. |

### 1.2.4. Configure Forced Flow Forwarding Mode

#### Configuration Condition

None

#### Configure Forced Flow Forwarding Mode

On the shortest path, forwarding packet is forced to use flow forwarding.

Table 1-5 Configure the forced flow forwarding mode

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the forced flow forwarding mode | **forwarding flow force-mode** | Optional<br>By default, shortest path uses the forced flow forwarding. |

# 2. IPV4 FORWARDING

## 2.1. Overview

The IPv4 forwarding includes the entire process from receiving the packet to sending out the packet. After receiving the packet, the interface uses the MEF to search for the routing, finds the corresponding outbound interface, and then sends out the packet. At the same time, if the ACL service is configured for the packet Rx and Tx interfaces, the flow forwarding records the packet handling result by the ACL. In this way, the packet can be searched for the corresponding handling result in the flow table entry and there is no need to deliver the packet to the ACL for handling. Thus, the flow forwarding accelerates the packet forwarding.

### 2.1.1. MEF Overview

The MEF mainly searches for the routing and obtains the forwarding information on the fast forwarding platform. The MEF table entry is the projection of the core routing table. It synchronously updates with the core routing table and searches for the routing using the LPM (Longest Prefix Match). The MEF table entry consists of two parts: FIB block and ADJ information. The MEF obtains the packet forwarding information by searching for the FIB block and ADJ information.

**The MEF has all the features and benefits of fast forwarding:**

- The searching speed is fast. No matter how large the forwarding table is, the routing searching can be completed upon a maximum of four times memory visits. The MEF is applicable to scenarios of all scales.
- The MEF has a stable performance. Because the complete projection of the core routing table is saved in the MEF table entry, if the driver receives the packet and the MEF fails to search the table, the packet is dropped directly. In this case, the upper CPU load is greatly relieved. This ensures that the CPU performance is given to the full play.
- The MEF is created before the packet flow. The cache entry is created without the progress switching. This improves the forwarding performance.
- The MEF will not be aging and updates synchronously with the core routing table. When the core routing table changes or the link information changes, the MEF table entry updates synchronously. No problems in the fast switching such as cache aging or invalid cache exist. Free from the routing instability, the MEF can be applied to multiple complex network scenarios.
- The link information is saved in the MEF. Complete forwarding information can be obtained by searching the forwarding table in the receiving progress, and then the forwarding is completed. There is no overhead related to the progress switching.
- The MEF provides effective load balancing mechanism and effectively utilizes the network resources by target flow, packet, and source flow.

### 2.1.2. Flow Forwarding Overview

The flow forwarding mainly records the forwarding and control result of the stable data flow on the fast forwarding platform and applies them to the subsequent packet forwarding of this data flow. In the common handling process, when the packet enters the router, actions such as packet resolution, link transmission protocol, ACL, NAT, PBR, and outbound routing are mandatory. Subsequently, the packet with the same header of the preceding packet will be

handled in the same process when it enters the router. The flow forwarding records the results that every module on the forwarding path handles the packet to the flow table when the packet is completely handled. When the subsequent packet reaches, directly search the flow table. If such flow already exists in the flow table, handle the packet using the result of the preceding packet instead of completely handling the packet. In this way, the forwarding speed accelerates.

In the packet forwarding process, after the packet is received, a corresponding table entry is searched in the flow table. If a corresponding table entry is found, go on the subsequent handling. If a corresponding table entry fails to be found, the packet is handled based on the process of creating the flow table, as shown in Figure 2-1.

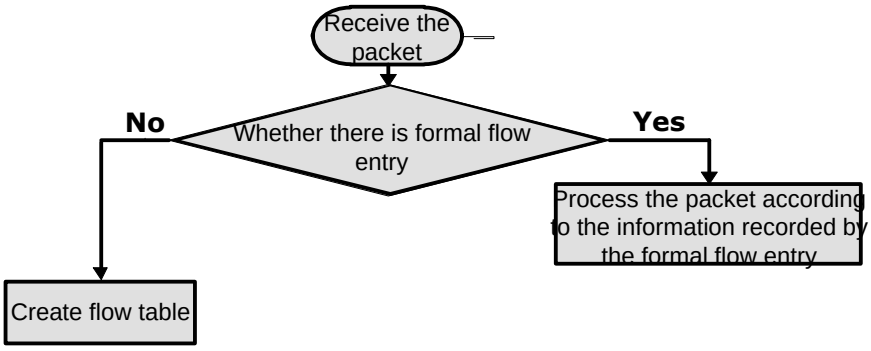



Figure 2-1 Handling process of the flow forwarding packet

## 2.2. IPV4 Forwarding Function Configuration

Table 2-1 Flow forwarding function list

| Configuration Task | |
|---|---|
| Configure basic functions of the flow forwarding | Enable the flow forwarding function |
| | Configure the maximum number of entries for the flow table |
| | Configure the aging time for the active flow |
| | Configure the aging time for the non-active flow |
| | Configure periodical reporting |
| | Configure fragment packet forwarding function |
| Configure the IP packet forwarding processing mode | Configure the processing mode for the IP packet with the options |
| | Configure the checksum for the IP packet |

## 2.2.1. Configure Basic Functions of Flow Forwarding

### Configuration Condition

None

### Enable Flow Forwarding Function

After enabling the flow forwarding, the packet handling results will be recorded in the flow forwarding table entry.

Table 2-2 Enable the flow forwarding function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the flow forwarding function | **ip flow enable** | Optional<br>By default, the flow forwarding function is enabled. |

### Configure Maximum Number of Entries for Flow Table

Because the flow table entry occupies large memory, it is recommended that the user not change the maximum number of entries for the flow table.

Table 2-3 Configure the maximum number of entries for the flow table

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Set the maximum number of entries for the flow table | **ip flow entries** *slot-number entry-number* | Mandatory<br>By default, the maximum number of the entries of the flow table is 256K. |

**Note:**

- The maximum number of entries of the flow table is set in the flow forwarding initialization based on the memory size of the current device. Therefore, manual configuration is not recommended.

### Configure Aging Time for Active Flow

If a flow table entry is hit by a packet flow in a certain time, the packet flow corresponding to the flow table entry is called the active flow. When the existence time of the active flow exceeds the aging time of the active time, the flow table entry corresponding to the active flow will be forced to age. That is, this flow table entry is deleted.

Table 2-4 Configure the aging time for the active flow

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Set the aging time for the active flow | **ip flow timeout active** *active-time* | Mandatory<br>By default, the aging time for the active flow is 3600 minutes. |

### Configure Aging Time for Non-active Flow

If a flow table entry is not hit by a packet flow in a certain time, the packet flow corresponding to the flow table entry is called the non-active flow. When the existence time of the non-active flow exceeds the aging time of the non-active time, the flow table entry will be forced to age. That is, this flow table entry is deleted.

Table 2-5 Configure the aging time for the non-active flow

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the aging time for the non-active flow | **ip flow timeout inactive** *inactive-time* | Mandatory<br>By default, the aging time of the non-active flow is 60s. |

### Configure Periodic Reporting

The command is used to control whether to report the IPv4 data flow to ipfix periodically. By default, it is reported periodically.

Table 2-6 Configure reporting periodically

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |

| Step | Command | Description |
|---|---|---|
| Set periodical reporting | **ip flow period upstream** | Optional<br>By default, the IP data flow is reported to ipfix periodically. |

### Configure Fragment Packet Flow Forwarding Function

The command is used to enable the flow forwarding function of the IPv4 fragment packets. By default, the function is disabled.

Table 2-7 Configure the fragment packet flow forwarding function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the fragment packet flow forwarding function | **ip flow fragment** | Optional<br>By default, the function is disabled. |

## 2.2.2. Configure IP Packet Forwarding Processing Mode

### Configuration Condition

None

### Configure Processing Mode for the IP Packets with Options

During forwarding, you can configure the processing mode for the IP packets with options. You can process or drop.

Table 2-8 Configure the processing mode for the IP packets with options

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the processing mode for the IP packets with options | **dfp ip options** { **process** \| **drop** } | Mandatory<br>By default, it is necessary to process the IP packet with options. |

**Configure Checksum for IP Packet**

During forwarding, you can configure checksum for the received IP packet.

Table 2-9 Configure checksum for the received IP packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure checksum for the received IP packet | **dfp ip recv-cksum** | Mandatory<br>By default, do not perform checksum for the received IP packet. |

## 2.2.3. IPv4 Forwarding Monitoring and Maintaining

Table 2-10 IPv4 forwarding monitoring and maintaining

| Command | Description |
|---|---|
| **show ip flow** { **interface** *interface-name* { **ingress** \| **egress** } \| **parameter \| entry \| user** } | Display the flow table entry information or the configuration parameters of the flow forwarding |

# 3. IPV6 FORWARDING

## 3.1. IPV6 Forwarding Function Configuration

Table 3-1 Flow forwarding function list

| Configuration Task | |
|---|---|
| Configure basic functions of the flow forwarding | Enable the flow forwarding function |
| | Configure the maximum number of entries for the flow table |
| | Configure the aging time for the active flow |
| | Configure the aging time for the non-active flow |
| | Configure the periodic reporting |
| | Configure the fragment packet flow forwarding function |
| Configure IPv6 packet forwarding processing mode | Configure the processing mode for the IPv6 packet with options |
| | Configure the checksum for the IPv6 packet |

### 3.1.1. Configure Basic Functions of Flow Forwarding

#### Configuration Condition

None

#### Enable Flow Forwarding Function

After enabling the flow forwarding, the packet handling results will be recorded in the flow forwarding table entry.

Table 3-2 Enable the flow forwarding function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the flow forwarding function | **ipv6 flow enable** | Optional<br>By default, the flow forwarding function is enabled. |

### Configure Maximum Number of Entries for Flow Table

Because the flow table entry occupies large memory, it is not recommended to modify the maximum number of entries for the flow table.

Table 3-3 Configure the maximum number of entries for the flow table

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Set the maximum number of entries for the flow table | **ipv6 flow entries** *lpu-number entry-number* | Mandatory<br>By default, the maximum items of the flow table is 4096K. |

**Note:**

- The maximum number of entries of the flow table is set in the flow forwarding initialization based on the memory size of the current device. Therefore, manual configuration is not recommended.

### Configure Aging Time for Active Flow

If a flow table entry is hit by a packet flow in a certain time, the packet flow corresponding to the flow table entry is called the active flow. When the existence time of the active flow exceeds the aging time of the active time, the flow table entry corresponding to the active flow will be forced to age. That is, this flow table entry is deleted.

Table 3-4 Configure the aging time for the active flow

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Set the aging time for the active flow | **Ipv6 flow-cache timeout active** *active-time* | Mandatory<br>By default, the aging time for the active flow is 3600 minutes. |

### Configure Aging Time for Non-active Flow

If a flow table entry is not hit by a packet flow in a certain time, the packet flow corresponding to the flow table entry is called the non-active flow. When the existence time of the non-active flow exceeds the aging time of the non-active time, the flow table entry will be forced to age. That is, this flow table entry is deleted.

Table 3-5 Configure the aging time for the non-active flow

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Set the aging time for the non-active flow | **Ipv6 flow-cache timeout inactive** *inactive-time* | Mnadatory<br>By default, the aging time for the non-active flow is 60 minutes. |

### Configure Periodic Reporting

The command is used to control whether the IPv6 data flow is periodically reported to ipfix. By default, it is reported periodically.

Table 3-6 Configure periodic reporting

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |

| Step | Command | Description |
|---|---|---|
| Configure periodic reporting | **ipv6 flow period upstream** | Optional<br>By default, the IPv6 data floe can be periodically reported to ipfix. |

**Configure Fragment Packet Flow Forwarding Function**

The command is used to enable the flow forwarding function of the fragment packet. By default, the flow forwarding function of the fragment packet is disabled.

Table 3-7 Configure the fragment packet forwarding function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the flow forwarding function of the fragment packet | **ipv6 flow fragment** | Optional<br>By default, the flow forwarding function of the fragment packet is disabled. |

### 3.1.2. IPv6 Forwarding Monitoring and Maintaining

Table 3-8 IPv6 forwarding monitoring and maintaining

| Command | Description |
|---|---|
| **show ipv6 flow** { **interface** *interface-name* { **ingress** \| **egress** } \| **parameter \| entry \| user** } | Display the configuration parameters of the flow entry information or flow forwarding |

# 4. ARP

## 4.1. Overview

Address Resolution Protocol (ARP) provides dynamic mapping from IP addresses to MAC addresses. The Ethernet frames to be transmitted in the Ethernet can be encapsulated properly only after MAC addresses are specified. The ARP protocol is used to obtain MAC addresses that correspond to IP addresses.

## 4.2. ARP Function Configuration

Table 4-1 RP function list

| Configuration Tasks | |
|---|---|
| Configure basic functions of ARP. | Configure a static ARP entry. |
| | Configure the maximum number of dynamic ARP entries. |
| | Configure the dynamic ARP aging time. |
| | Configure ARP receive queue depth. |
| | Configure ARP proxy. |
| Configure ARP anti-attack function | Configure ARP anti-spoofing attack |

### 4.2.1. Configure Basic Functions of ARP

#### Configuration Condition

None

#### Configure a Static ARP Entry

Configuring static ARP means that a user manually specifies the mapping between IP addresses and MAC addresses.

Table 4-2 Configuring static ARP

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |

| Step | Command | Description |
|---|---|---|
| Configure a static ARP entry. | **arp** [ **vrf** *vrf-name* ] { *ip-address* \| *host-name* } *mac-address* [ **alias** [ **advertise** ] \| **advertise** [ **alias** ] ] | Mandatory. |

**Note:**

- When the configured static ARP entry contains an alias, if an ARP request for this IP address is received, the MAC address in the static ARP entry is used for response.
- When the configured static ARP entry contains the advertise option, the static ARP is notified regularly if the static ARP notification is enabled.

### Configure the Maximum Number of Dynamic ARP Entries

The purpose of configuring the maximum number of dynamic ARP entries is to prevent dynamically learned ARP from occupying too many system resources.

Table 4-3 Configuring the maximum number of dynamic ARP entries

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum number of dynamic ARP entries. | **arp limited** *max-entries* | Mandatory.<br>By default, the maximum number of dynamic ARP entries is 2000. |

### Configure the Dynamic ARP Aging Time

The life cycle of a dynamically learned ARP entry is the aging time. Within the aging time, the device sends ARP requests periodically. If it receives an ARP response, it resets the aging time. If the aging time expires, the device deletes the dynamic ARP entry.

Table 4-4 Configuring the dynamic ARP aging time

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |

| Step | Command | Description |
|---|---|---|
| Configure the dynamic ARP aging time. | **arp timeout** { *second* \| **disable** } | Mandatory.<br>The default aging time is 1200 seconds. |

**Configure ARP Receive Queue Depth**

The received ARP packets will be first cached in the ARP receive queue and the system reads and handles the packets cached in the receive queue in turn. When the cached ARP packets reach the queue depth, the subsequent received ARP packets will be dropped. The user can adjust the ARP receive queue depth based on the network burst ARP.

Table 4-5 Configure the ARP receive queue depth

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the ARP receive queue depth | **arp queue-length** *length* | Mandatory<br>By default, the queue depth is 200 bytes. |

**Configure ARP Proxy**

An ARP request is sent by one network host to another network, and the intermediate device between the two networks can respond to the ARP request. This process is called ARP proxy.

Table 4-6 Configuring ARP proxy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Configure ARP proxy. | **ip proxy-arp** | Mandatory.<br>By default, the ARP proxy function is enabled. |

## 4.2.2. Configure ARP Anti-attack Function

### Configuration Condition

None

### Configure ARP Anti-proofing Attack

In terms of the ARP proofing on the Ethernet, configure the ARP anti-proofing attack to avoid such problems.

Configuring this function includes ARP scanning, ARP binding, and static ARP notification enabling.

- ARP scanning is used to complete the ARP resolution in a specified IP address segment.
- ARP binding is used to bind the dynamic ARP as static ARP.
- Static ARP notification enabling is used to notify the ARP entries with notification tags in the static ARP.

Table 4-7Configure the ARP anti-spoofing attack

| Step | Command | Description |
|---|---|---|
| Execute ARP scanning | **arp** [ **vrf** *vrf-name* ] **scan** *begin-ip-address end-ip-address* [ **bind** [ **alias** [ **advertise** ] \| **advertise** [ **alias** ] ] ] | Mandatory<br>When the bind is contained, the scanned dynamic ARP is bound as the static ARP. |
| Execute ARP binding | **arp** [ **vrf** *vrf-name* ] **bind** { all **\|** { *begin-ip-address end-ip-address* [ **alias** [ **advertise** ] \| **advertise** [ **alias** ] ] } } | Optional |
| Enter the global configuration mode | **configure terminal** | - |
| Configure the static ARP with notification tags | **arp** [ **vrf** *vrf-name* ] *ip-address mac-address* **advertise** | Optional<br>The static ARP is configured independently. |
| Enable the static ARP notification | **arp advertise** [ **interval** *interval-time* ] | Mandatory<br>By default, this function disabled. |

### 4.2.3. ARP Monitoring and Maintaining

Table 4-8 ARP monitoring and maintaining

| Command | Description |
|---|---|
| **clear arp attack-detection** | Clears the suspected host information of ARP attack |
| **clear arp-cache [***start-ip-addr end-ip-addr* **[force]] | [interface** *if-name* **[force]] | [vrf** *vrf-name* **[***start-ip-addr end-ip-addr* **[force]]] | [force]** | Clears the ARP cache |
| **show arp statistic** [**all** ] | Displays the ARP quantity statistics information |
| **show arp** [ **vrf** *vrf-name* ] | Displays the ARP table. |
| **show arp attack-detection** | Displays the information about the host which has been suspected of initiating static ARP attacks. |

## 4.3. ARP Typical Configuration Example

### 4.3.1. Configure ARP Proxy

#### Network Requirements

- Device is directly connected to PC1 and PC2 respectively. The network number of the LAN in which PC1 and PC2 is located is 10.0.0.0/16.
- The MAC address of the interface gigabitethernet0 of Device is 0001.7a13.0102.
- Through the ARP proxy of Device, PC1 can successfully ping PC2, and PC1 can learn the MAC address of PC2.

#### Network Topology

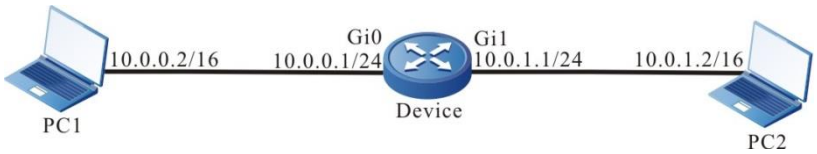



Figure 4-1 Networking for configuring ARP proxy

#### Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Check the result.

#Ping the PC2 IP address 10.0.1.2 from PC1.

```
C:\Documents and Settings>ping 10.0.1.2

Pinging 10.0.1.2 with 32 bytes of data:

Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255
Reply from 10.0.1.2: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.1.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#Query the ARP entry of Device.

```
Device#show arp
Protocol  Address     Age (min) Hardware Addr   Type   Interface          Switchport
Internet  10.0.0.1      -       0001.7a13.0102  ARPA   gigabitethernet0       N/A
Internet  10.0.0.2      1       B8AC.6F2D.4498  ARPA   gigabitethernet0       N/A
Internet  10.0.1.1      -       0001.7a13.0103  ARPA   gigabitethernet1       N/A
Internet  10.0.1.2      1       4437.e603.0d63  ARPA   gigabitethernet1       N/A
```

#Query the ARP entry of PC1.

```
C:\Documents and Settings>arp -a

Interface: 10.0.0.2 --- 0x5
  Internet Address      Physical Address      Type
  10.0.0.1              00-01-7a-13-01-02     dynamic
  10.0.1.2              00-01-7a-13-01-02     dynamic
```

#Ping from PC1 to PC2 is successful, the PC1 has learned the MAC address of PC2.

**Note:**

- By default, ARP proxy is enabled for a device.

### 4.3.2. Configure a Static ARP Entry

#### Network Requirements

- Device and PC are directly connected.
- The MAC address of PC is 4437.e603.0d63.
- The IP address and MAC address of PC is bound to Device.
- PC can successfully ping the address of the interface gigabitethernet1 of Device.

## Network Topology

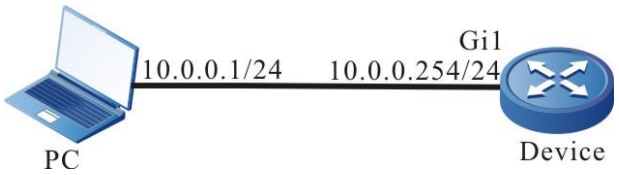



Figure 4-2 Networking for configuring a static ARP entry

## Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Bind the IP address and MAC address of PC to Device.

#Configure Device.

Bind the IP address and MAC address of PC to Device.

```
Device(config)#arp 10.0.0.1 4437.e603.0d63
```

**Step 3:** Check the result.

#Query the ARP entry of Device.

```
Device1#show arp
Protocol  Address     Age (min) Hardware Addr   Type   Interface          Switchport
Internet  10.0.0.1      -       4437.e603.0d63  ARPA   gigabitethernet1       NA
Internet  10.0.0.254    -       0001.7a13.0102  ARPA   gigabitethernet1       NA
```

#PC pings the IP address 10.0.0.254 of the interface gigabitethernet1 on Device.

```
C:\Documents and Settings>ping 10.0.0.254

Pinging 10.0.0.254 with 32 bytes of data:

Reply from 10.0.0.254: bytes=32 time<1ms TTL=255
Reply from 10.0.0.254: bytes=32 time<1ms TTL=255
Reply from 10.0.0.254: bytes=32 time<1ms TTL=255
Reply from 10.0.0.254: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.0.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#PC can successfully ping the address of the interface gigabitethernet1 of Device.

# 5. IP

## 5.1. Overview

The Internet Protocol (IP) is based on data packets. It is used in data exchange between computer networks. The protocols that are supported by the device include: IP, Internet Control Message Protocol (ICMP), Transfer Control Protocol (TCP), User Datagram Protocol (UDP), and Socket.

Among them, IP packets are the base of the TCP/IP protocol stack. The IP layer is responsible for addressing, fragmentation, reassembly, and protocol information partitioning. As the network layer protocol, the IP protocol performs route addressing and control packet transmission.

The UDP protocol and the UDP protocol is set up based on the IP protocol. They provide connection-based reliable data transmission services and non-connection-based unreliable data transmission services respectively.

ICMP is mainly used to provide network detection services. It also provides an error report if the network layer or transmission layer protocol becomes abnormal, and it informs the related device of the abnormality to facilitate network control management.

## 5.2. IP Function Configuration

Table 5-1 IP basic function list

| **Configuration Tasks** | |
|---|---|
| Configure an IP address. | Configure an IP address for an interface. |
| | Configure an unnumbered IP address for an interface. |
| Configure basic functions of the IP protocol. | Configure the depth of the IP packet receive queue. |
| | Configure the Time To Live (TTL) of transmitted IP packets. |
| | Configure timeout for packet reassembly. |
| | Enable IP packet receiving verification and check. |
| | Configure transmitted IP packets to calculate a checksum. |
| | Enable IP routing cache. |

| Configuration Tasks | |
|---|---|
| Configure basic functions of the ICMP protocol. | Enable global ICMP redirection. |
| | Enable global ICMP redirection. |
| | Enable ICMP destination unreachable. |
| | Configure ICMP rate limit. |
| Configure basic functions of the TCP protocol. | Configure the size of the TCP receiving cache. |
| | Configure the size of the TCP transmitting cache. |
| | Configure the maximum number of TCP retransmissions. |
| | Configure the maximum length of TCP packets. |
| | Configure the maximum TCP round-trip time. |
| | Configure the TCP connection idle time. |
| | Configure TCP connection setup waiting time. |
| | Configure the maximum number of TCP keep-alive times. |
| | Enable the TCP timestamp. |
| | Enable TCP selective retransmission. |
| Configure the TCP protocol anti-attack function | Enable the TCP syncache function |
| | Enable the TCP syncookies function |
| | Enable the TCP connection accelerated aging function |

| Configuration Tasks | |
|---|---|
| Configure basic functions of the UDP protocol. | Configure TTL of UDP packets. |
| | Configure the size of the UDP receiving cache. |
| | Configure the size of the UDP transmitting cache. |
| | Enable UDP verification and check. |
| | Fill in UDP packet checksum. |

### 5.2.1. Configure an IP Address

An IP address is a 32-bit number. It is used to uniquely identify a network device that runs the IP protocol on the Internet.

**An IP address consists of the following two parts:**

- Network number (Net-id): It identifies the network in which the device is located.
- Host number (Host-id): It specifies the host number in the device network.

To facilitate IP address management, IP addresses are categorized into five classes, and each IP address class has its own functions. IP addresses of classes A to C are used for address allocation, IP addresses of class D is used in multicast applications, and IP addresses of class E are used for test purpose. The following table shows the IP addresses classes and their ranges.

Table 5-2 IP address classes and their ranges

| Address Type | Available Network Address Range | Description |
|---|---|---|
| A | 1.0.0.0-127.0.0.0 | Network number 127 is used for loopback interfaces. |
| B | 128.0.0.0-191.255.0.0 | - |
| C | 192.0.0.0-223.255.255.0 | - |
| D | 224.0.0.0-239.255.255.255 | Class D addresses are used for multicast. |

| Address Type | Available Network Address Range | Description |
|---|---|---|
| E | 240.0.0.0-255.255.255.255 | 255.255.255.255 is used for broadcast address and the other class E addresses are used for test purpose. |

With the development of the Internet, IP address resources have gradually been consumed up. Address allocation based on classes causes address waste, so the concept of "subnet" is introduced. "Subnet" takes some host numbers in the IP addresses as subnet numbers. In this way, a large network is divided into multiple subnets. This facilitates network planning and deployment.

The three address segments, 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255, and 192.168.0.0-192.168.255.255 are private and reserved addresses, and they cannot be allocated to the public network.

This section describes how to configure an interface IP address and how to configure an unnumbered interface IP address.

### Configuration Condition

None

### Configure an IP Address for an Interface

An IP address can only be configured for an interface that supports the IP protocol. One interface can only be configured with one primary IP address but it can be configured with multiple secondary IP addresses.

Table 5-3 Configuring an IP address for an interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Configure an IP address for an interface. | **ip address** *ip-address* { *network-mask* \| *mask-len* } [ **secondary** ] | Mandatory |

**Note:**

- One interface can only be configured with one primary IP address, therefore, the newly configured primary IP address replaces the original primary IP address.
- Before an interface is configured with secondary IP addresses, the interface must be configured a primary IP address. An interface can be configured with a maximum of 100 secondary IP addresses.
- The IP addresses of different interfaces must not in the same network segment, but the primary and secondary IP addresses of one interface can be in the same network segment.

### Configure an Unnumbered IP Address for an Interface

Unnumbered IP addresses save IP addresses. In the case of unnumbered IP addresses, the IP addresses of other interfaces can be borrowed instead of allocated independently. If an unnumbered interface generates an IP packet, the source IP address of the packet is the primary IP address of a borrowed interface. In configuring an unnumbered IP address for an interface, the interface to be borrowed must be specified, so that the IP address of the interface can be borrowed.

Table 5-4 Configuring an unnumbered IP address for an interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Configure an unnumbered IP address for an interface. | **ip unnumbered** *reference-interface* | Mandatory |

**Note:**

- The borrowed interface must be configured with the primary IP address, and the interface must not be configured with an unnumbered IP address.
- The primary IP address of an interface can be borrowed by multiple interfaces, but only the primary IP address of the interface can be borrowed.

## 5.2.2. Configure Basic Functions of the IP Protocol

In the TCP/IP protocol stack, the IP protocol is the network layer core protocol that is responsible for network interconnection. The IP protocol is a connectionless protocol. Before transmitting data, you need not set up a connection. The IP protocol tries best to deliver packets, but is does not ensure that all packets can reach the destination orderly.

### Configuration Condition

None

### Configure the Depth of the IP Packet Receive Queue

The IP packets received by a device are first cached in the IP packet receive queue of an interface. The system reads packets orderly in the queue for processing. If the cached IP packets reach the specified queue depth, the later IP packets are discarded. You can adjust the depth of the IP packet receive queue according to burst of IP packets in the network.

Table 5-5 Configuring the depth of the IP packet receive queue

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the depth of the IP packet receive queue to a specified value. | **ip option queue-length** *queue-size* | Mandatory.<br>By default, the depth of the IP packet receive queue is 200 bytes. |
| Configure the depth of the IP packet receive queue to the default value. | **default ip option queue-length** | Optional |

### Configure the TTL of Transmitted IP Packets

The header of an IP packet contains the Time-To-Live (TTL) field, which is decreased by one once the IP packet passes a routing device. When the TTL is 0, the device discards the IP packet. By default, the TTL of IP packets transmitted by the device is 255, that is, the packet can only be transmitted for up to 255 times. If you want to limit the number of packet forwarding times, adjust the TTL value for the transmitted IP packets.

Table 5-6 Configuring the TTL of transmitted IP packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the TTL of transmitted IP packets to a specified value. | **ip option default-ttl** *ttl - value* | Mandatory.<br>By default, the TTL of transmitted IP packets is 255. |
| Configure the TTL of transmitted IP packets to a default value. | **default ip option default-ttl** | Optional |

### Configure Timeout for Packet Reassembly

If an IP packet is fragmented during the transmission, after the fragments reach the destination, they need to be reassembled to form a complete IP packet. Before all fragments are received, the received fragments are cached temporarily. If reassembly times out before all fragments reach the destination, the received fragments are discarded.

Table 5-7 Configuring timeout for packet reassembly

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configuring timeout for packet reassembly to a specified value. | **ip option fragment-ttl** *ttl-value* | Mandatory.<br>By default, the timeout for packet reassembly is 60, and the unit is 0.5 second. |
| Configuring timeout for packet reassembly to the default value. | **default ip option fragment-ttl** | Optional |

**Note:**

- The unit for timeout of packet reassembly is 0.5 second.

### Enable IP Packet Receiving Verification and Check

You can enable this function to verify and check the received IP packets. If the checksum is incorrect, the packet will be discarded.

Table 5-8 Enabling IP packet receiving verification and check

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable IP packet receiving verification and check. | **ip option recv-checksum** | Mandatory.<br>By default, the function is enabled. |
| Configure the method for verifying and checking the received IP packets to the default value. | **default ip option recv-checksum** | Optional |

### Enable IP Routing Cache

After a packet is sent from socket to the IP layer, if the destination address is the same as the previous packet and the route is valid, the packet directly use the route in the cache without the need of searching for another route.

Table 5-9 Enabling IP routing cache

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable IP routing cache. | **ip upper-cache** | Mandatory.<br>By default, the IP routing cache function is enabled. |

## 5.2.3. Configure Basic Functions of the ICMP Protocol

In the TCP/IP protocol stack, ICMP is mainly used to provide network detection services. It also provides an error report if the network layer or transmission layer protocol becomes abnormal, and it informs the related device of the abnormality to facilitate network control management.

### Configuration Condition

None

### Enable Global ICMP Redirection

After a device receives an IP packet to be forwarded, if it is found that the receiving interface of the packet and the transmitting interface of the packet are the same through route selection, the device forwards the packet and sends back an ICMP redirection packet to the source end, requesting the source end to reselect the correct next hop for transmission of later packets. By default, a device can send ICMP redirection packets. In some special cases, you can prohibit a device from sending ICMP redirection packets.

Table 5-10 Enabling global ICMP redirection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable global ICMP redirection. | **ip redirect** | Mandatory.<br>By default, the global ICMP redirection function is enabled. |

### Enable Global ICMP Redirection

In sending ICMP redirection packets, if you need to send ICMP redirection packets, you need to enable the ICMP redirection function on the interface.

Table 5-11 Enabling global ICMP redirection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Enable ICMP redirection on an interface. | **ip redirects** | Mandatory.<br>By default, the ICMP redirection function is enabled on an interface. |

**Note:**

- You can send ICMP redirection packets only when the ICMP redirection function is enabled globally and on the interface.

### Enable ICMP Destination Unreachable

After a device receives IP data packets, if the destination is unreachable, the packet is discarded and the ICMP destination unreachable error packet is sent back to the source end.

- If route selection of a forwarded IP packet fails, a host unreachable ICMP error packet is sent back to the source end.
- For an IP packet that can be forwarded, if you need to fragment the IP packet but a Don't Fragment (DF) bit is set in the packet, an ICMP error packet indicating that "segmentation is required but a DF bit is set" is sent to the source end.
- For an IP packet whose destination address is the local device, if the device does not support the upper-layer protocol of the device, it sends a "protocol unreachable" ICMP error packet to the source end.
- For an IP packet whose destination address is the local device, if the transport layer port of the packet of the packet does not match the port that the device process monitors, the device sends back a "port unreachable" ICMP error packet to the source end.

If a device encounters a malicious attack by a large number of ICMP destination unreachable packets, the device performance is degraded, and network traffic is increased. To prevent such case, you can disable the function of sending ICMP destination unreachable packets.

Table 5-12 Enabling ICMP destination unreachable

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Enable ICMP destination unreachable. | **ip unreachables** | Optional.<br>By default, the ICMP destination unreachable function is enabled. |

### Configure ICMP Rate Limit

If a device encounters a malicious attack by a large number of ICMP error packets, the device performance is degraded, and network traffic is increased. To prevent such case, you can configure ICMP packet rate limit. The types of ICMP error packets include: unreachable packets, redirection packets, TTL timeout packets, and parameter error packets. The default rate limit of the packets is 10pps, while the transmitting rate of the other types of packets is 0, that is, no rate limit. In addition, you can configure the transmitting rates for different types of packets independently. If no rate is configured for a type of packets, the default value is used.

Table 5-13 Configuring ICMP rate limit

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable ICMP rate limit. | **ip icmp ratelimit enable** | Mandatory.<br>By default, the function is enabled. |
| Configure ICMP rate limit. | **ip icmp ratelimit** { **default** *pps* \| **echo-reply** { *pps* \| **unlimit** } \| **mask-reply** { *pps* \| **unlimit** } \| **param-problem** { *pps* \| **unlimit** } \| **redirect** { *pps* \| **unlimit** } \| **time-exceed** { *pps* \| **unlimit** } \| **time-stamp-reply** { *pps* \| **unlimit** } \| **unreach** { *pps* \| **unlimit** } } | Mandatory.<br>By default, the ICMP rate limit function is enabled. |

## 5.2.4. **Configure Basic Functions of the TCP Protocol**

In the TCP/IP protocol stack, TCP is a connection-oriented transport layer protocol. Before sending data through the TCP protocol, you must first set up a connection. The TCP protocol provides congestion control and ensures reliable data transmission.

### Configuration Condition

None

### Configure the Size of the TCP Receiving Cache

In some special network environment, you can configure both the size of receiving cache and the size of transmitting cache of a TCP connection so that the network can reach the optimal performance. If the TCP connection receiving cache is not configured, the size of the receiving cache is the default value.

Table 5-14 Configuring the size of the TCP receiving cache

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the size of the TCP receiving cache. | **ip tcp recvbuffers** *buff-size* | Mandatory.<br>By default, the size of the receiving cache is 8192 bytes. |

### Configure the Size of the TCP Transmitting Cache

In some special network environment, you can configure both the size of receiving cache and the size of transmitting cache of a TCP connection so that the network can reach the optimal performance. If the TCP connection transmitting cache is not configured, the size of the transmitting cache is the default value.

Table 5-15 Configuring the size of the TCP transmitting cache

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the size of the TCP transmitting cache. | **ip tcp sendbuffers** *buff-size* | Optional.<br>By default, the size of the transmitting cache is 8192 bytes. |

### Configure the Maximum Number of TCP Retransmissions

After the server sends a SYN-ACK packet, if it does not receive a response packet from the client, the server retransmits the packet. If the number of retransmissions exceeds the maximum number of retransmissions defined by the system, the system disconnects the TCP connection.

Table 5-16 Configuring the maximum number of TCP retransmissions

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum number of TCP retransmissions. | **ip tcp retransmits** *retransmits-count* | Mandatory.<br>By default, the maximum number of TCP retransmissions is 3. |

### Configure the Maximum Length of TCP Packets

The maximum length of TCP packets is the maximum length of data blocks that are sent by the transmitting end of a TCP connection to the receiving end. When a connection is set up, the smaller maximum packet length of the two ends is used as the maximum packet length in sending TCP packets by the two ends. If a TCP packet exceeds the maximum packet length, the transmitting ends fragments the packet before sending it.

Table 5-17 Configuring the maximum length of TCP packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum length of TCP packets. | **ip tcp segment-size** *seg-size* | Optional.<br>By default, the maximum length of TCP packets is 512 bytes. |

### Configure the Maximum TCP Round-Trip Time

The TCP round trip time refers to the time between the time point at which the transmitting end sends a TCP packet and the time point at which the transmitting end receives the response packet. The maximum TCP round-trip time that is configured during TCP connection setup is taken as the initial value of the TCP round-trip time. The later TCP round-trip time is calculated according to the actual round-trip time. By default, the maximum TCP round-trip time is 3 seconds.

Table 5-18 Configuring the maximum TCP round-trip time

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum TCP round-trip time. | **ip tcp round-trip** *round-trip-time* | Mandatory.<br>By default, the maximum TCP round-trip time is 3 seconds. |

### Configure the TCP Connection Idle Time

After a TCP connection is set up, if no data is exchanged, the TCP connection idle time times out. Then TCP performs a keep-alive test. After the maximum number of keep-alive times is reached, the TCP connection is disconnected. By default, the TCP connection idle time is 2 hours.

Table 5-19 Configuring the TCP connection idle time

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the TCP connection idle time. | **ip tcp idle-timeout** *idle-time* | Mandatory.<br>By default, the TCP connection idle time is 14400, and the unit is 0.5 second. |

**Note:**

- The unit of the TCP connection idle time is 0.5 second.

### Configure TCP Connection Setup Waiting Time

The setup of a TCP connection requires three handshakes. After a TCP client sends a connection request packet, it waits for the response from the TCP server before completing connection setup. After the time for waiting for connection setup timeout before a response is received, connection setup is terminated. By default, the time for waiting for setting up a TCP connection is 75 seconds.

Table 5-20 Configuring TCP connection setup waiting time

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure TCP connection setup waiting time. | **ip tcp init-timeout** *init-time* | Mandatory.<br>By default, the time for waiting for setting up a TCP connection is 150 seconds, and the unit is 0.5 second. |

**Note:**

- The unit of the TCP connection setup waiting time is 0.5 second.

**Configure the Maximum Number of TCP Keep-Alive Times**

If no data is exchanged on a TCP connection for TCP connection idle time, a TCP keep-alive packet is sent for keep-alive test. If the keep-alive test fails, a keep-alive test is performed again. If the maximum number of TCP keep-alive times exceeds the threshold, the TCP connection will be disconnected. By default, the maximum number of TCP keep-alive times is 3.

Table 5-21 Configuring the maximum number of TCP keep-alive times

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum number of TCP keep-alive times. | **ip tcp keep-count** *keep-count* | Mandatory.<br>By default, the maximum number of TCP keep-alive times is 3. |

**Enable the TCP Timestamp**

TCP automatically calculates the packet round-trip time according to the serial number of the request packet and that of the response packet. However, the calculation is not accurate. Use of TCP timestamps can revise the problem. The transmitting end adds a timestamp into a packet, and the receiving end sends back the timestamp in the response packet. The transmitting end calculates the packet round-trip time according to the returned timestamp. By default, the function is disabled.

Table 5-22 Enabling the TCP timestamp

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable the TCP timestamp. | **ip tcp timestamp** | Mandatory.<br>By default, the function is disabled. |

### Enable TCP Selective Retransmission

After TCP sends a series of packets, if the transmission of one packet fails, the series of packets need to be retransmitted. After TCP selective transmission is enabled, only the packet that fails to be transmitted needs to be retransmitted. This reduces the system and line cost. By default, the function is disabled.

Table 5-23 Enable TCP selective retransmission

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure TCP selective retransmission. | **ip tcp selective-ack** | Mandatory.<br>By default, the function is disabled. |

## 5.2.5. Configure TCP Anti-Attack Function

If the TCP server receives lots of SYN packets, but the peer end does not answer the SYN+ACK response of the server. As a result, lots of the server memory is consumed, occupying a half of the connection queue of the server, and the TCP server cannot serve for the normal request. As for the attack, you can configure the TCP anti-attack function to prevent.

### Configuration Condition

None

### Enable the TCP syncache Function

When the SYN packet is received, there is no rush to distribute TCB, but first reply one SYN+ACK packet and save the half-open connection information in a private cache until the correct response ACK packet is received.

Table 5-24 Enable the TCP syncache function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the TCP syncache function | **ip tcp syncache** | Mandatory<br>By default, the function is not enabled. |

### Enable the TCP syncookies Function

The function does not use any storage resource, but adopts one special algorithm to generate Sequence Number. The algorithm considers the fixed information of the peer IP and port, and local IP and port, as well as the local fixed information that the peer party does not know, such as MSS and time. After the peer ACK packet is received, re-calculate and view whether it is the same as the Sequence Number-1 in the peer response packet, so as to decide whether to distribute the TCB resource.

Table 5-25 Enable the TCP syncookies function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the TCP syncookies function | **ip tcp syncookies** | Mandatory<br>By default, the function is not enabled. |

### Enable TCP Connection Aging Function

This function checks the number of TCP connections to the current system and compares it with the expected connection threshold; if the number of current accessed TCP connections reaches two-thirds of the expected connection threshold, check and accelerate the aging of TCP connections to avoid the existence of dead connections and recover resources as soon as possible.

Table 5-26 Enable the TCP connection aging function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the function of accelerating the TCP connection | **ip tcp naptha threshold** *num* | Mandatory<br>By default, the function is not enabled. |

## 5.2.6. Configure TCP Max. Segment Size

This function sets the maximum segment size of the TCP packet on the interface. When the connection is established through TCP, the mss value is encapsulated into the header to set the maximum segment size of the packet sent by the connection to the mss value.

### Configuration Conditions

None

### Configure TCP Packet MSS

The command is used to control the size of the packet transmitted by connection as mss.

Table 5-27 Configure TCP connection mss

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration | **interface** *interface-name* | - |
| Configure the *mss* value | **ip tcp adjust-mss** *mss-value* | Mandatory<br>By default, it is not configured. |

## 5.2.7. Configure Basic Functions of the UDP Protocol

In the TCP/IP protocol stack, UDP is a connectionless-oriented transport layer protocol. Before sending data through the TCP protocol, you need not set up a connection. The UDP protocol provides unreliable data transmission without congestion control.

### Configuration Condition

None

### Configure TTL of UDP Packets

Configuring TTL of UDP packets means to fill in the TTL value in the IP header of UDP packets. The header of an IP packet contains the Time-To-Live (TTL) field, which is decreased by one once the IP packet passes a routing device. When the TTL is 0, the device discards the IP packet. By default, the TTL value of the IP packet of a UDP packet is 64.

Table 5-28 Configuring TTL of UDP packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure TTL of UDP packets. | **ip udp default-ttl** *time-to-live* | Mandatory.<br>By default, the TTL value of the IP packet of a UDP packet is 64. |

### Configure the Size of the UDP Receiving Cache

In some special network environment, you can configure both the size of receiving cache and the size of transmitting cache of a UDP connection so that the network can reach the optimal performance. If the UDP connection receiving cache is not configured, the size of the receiving cache is the default value, 41600 bytes.

Table 5-29 Configuring the size of the UDP receiving cache

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the size of the UDP receiving cache. | **ip udp recvbuffers** *buffer-size* | Mandatory.<br>By default, the size of the UDP receiving cache is 41600 bytes. |

### Configure the Size of the UDP Transmitting Cache

In some special network environment, you can configure both the size of receiving cache and the size of transmitting cache of a UDP connection so that the network can reach the optimal performance. If the UDP connection transmitting cache is not configured, the size of the transmitting cache is the default value, 9216 bytes.

Table 5-30 Configuring the size of the UDP transmitting cache

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the size of the UDP transmitting cache. | **ip udp sendbuffers** *buffer-size* | Mandatory.<br>By default, the size of the UDP transmitting cache is 9216 bytes. |

### Enable UDP Verification and Check

To prevent errors that occur during transmission of UDP packets, after UDP packets are received, UDP verification and check need to be performed. The system compares the UDP packet verification field calculated by the receiving end and the UDP packet header checksum field. If the two values are different, the system determines that a transmission error has occurred, and then discards the packet. By default, the function is enabled.

Table 5-31 Enabling UDP verification and check

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable UDP verification and check. | **ip udp recv-checksum** | Mandatory.<br>By default, the function is enabled. |

### Fill in UDP Packet Checksum

To prevent UDP packets from encountering transmission errors, in transmitting UDP packets, the transmitting end fills in the UDP packet checksum to be calculated into the UDP packet header checksum field for the receiving end to perform checksum check. By default, the function is enabled.

Table 5-32 Filling in UDP packet checksum

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure to fill in packet checksum in transmitting UDP packets. | **ip udp send-checksum** | Mandatory.<br>By default, the function is enabled. |

### 5.2.8. IP Basics Monitoring and Maintaining

Table 5-33 IP basics monitoring and maintaining

| Command | Description |
|---|---|
| **clear ip icmpstat** | Clears ICMP protocol statistics. |
| **clear ip statistics** | Clears IP protocol statistics. |
| **clear ip tcpstat** | Clears TCP protocol statistics. |
| clear ip tcp syncache statistics | Clears the syncache statistics information of the TCP protocol |
| **clear ip udpstat** | Clears UDP protocol statistics. |
| **show ip icmpstat** | Displays ICMP protocol statistics. |
| **show ip interface** [ *interface-name* **\| brief** ] | Displays the interface IP address. |
| **show ip sockets** | Displays the Socket details. |
| **show ip statistics [interface** *interface-name* **]** | Displays IP protocol statistics. |
| **show ip tcpstat** | Displays TCP protocol statistics. |

| Command | Description |
|---|---|
| **show ip tcp syncache detail** | Displays the syncache entity information of the TCP protocol |
| **show ip tcp syncache statistics** | Displays the TCP syncache statistics information |
| **show ip udpstat** | Displays UDP protocol statistics. |
| **show tcp tcb** [ **detail** ] | Displays TCP protocol control block details. |

# 6. IPV6 BASIS

## 6.1. Overview

IPv6 (Internet Protocol Version 6) is the second generation standard protocol of the network layer protocol, also known as IPng (IP Next Generation). It is a set of specifications designed by IETF (Internet Engineering Task Force) and an upgraded version of IPv4.

L3 interface ND proxy

The L3 interface ND agent is used to realize the interworking of different network segments connected through two interfaces. Usually, the device will not respond to the NS request of the network segment corresponding to the non packet receiving interface, so different network segments cannot communicate directly. When the nd proxy function is enabled for the L3 interface, if the device receives ns belonging to other interface network segments from the interface, it can also respond to NA. So that hosts belonging to different network segments can also establish neighbor entries and communicate normally. Its typical application scenario is the networking of large and small network segments.

## 6.2. IPv6 Basic Function Configuration

Table 6-1 IPv6 basic function configuration list

| **Configuration Tasks** | |
|---|---|
| Configure the IPv6 address | Configure the interface IPv6 address |
| Configure IPv6 basic functions | Enable the IPv6 unicast forwarding function |
| | Enable the interface IPv6 function |
| | Configure the IPv6 packet hop limit |
| | Configure the IPv6 MTU of the interface |
| Configure the IPv6 neighbor discovery protocol | Configure the IPv6 static neighbor |
| | Configure the aging time of the IPv6 neighbor entry in the STALE state Configure the interval of re-transmitting the NS packet |
| | Configure the times of sending the NS packet when IPv6 repeats the address detection |
| | Configure the related parameters of the RA packet |

| Configuration Tasks | |
|---|---|
| Configure the IPv6 neighbor discovery protocol | Enable the function of the interface sending the re-direct packet |
| Configure the ICMPv6 function | Configure the rate of sending the ICMPv6 error packet |
| | Enable the function of sending the ICMPv6 packet with unreachable destination |
| Configure the IPv6 TCP anti-attack function | Enable the TCP syncache function |
| | Enable the TCP syncookies function |
| | Enable the function of accelerating the aging of the TCP connection |
| Enable the ND fast pick-up function | Enable the ND fast pick-up function |
| Configure the ND proxy function of the L3 interface | Enable the ND proxy function of the L3 interface |

### 6.2.1. **Configure Interface IPv6 Address**

The most striking difference between IPv6 and IPv4 is that the length of the IP address increases from 32 bits to 128 bits. The IPv6 address is represented as a series of 16-bit hex numbers separated by colon (:). Each IPv6 address is divided into eight groups, each group of 16 bits represented by four hexadecimal digits, and the groups are separated by colons, such as 2000:0000:240F:0000:0000:0CB0:123A:15AB.

**In order to simplify the representation of the IPv6 address, process the “0” in the IPV6 address as follows:**

- The preamble “0” in each group can be omitted, that is, the above address can be represented as 2000:0:240F:0:0:CB0:123A:15AB.
- If the address contains two or more consecutive groups of zero, it can be replaced by a double colon "::" that is, the above address can be represented as 2000:0:240F:: CB0:123A:15AB.
- The double colon "::" can only be used once in an IPv6 address. Otherwise, the number of zeros represented by "::" cannot be determined when the device converts "::" to zero to recover 128-bit addresses.

The IPv6 address consists of two parts: address prefix and interface identifier. The address prefix is equivalent to the network number field in the IPv4 address and the interface identifier is equivalent to the host number field in the IPv4 address.

The IPv6 address prefix is expressed as: IPv6 address/prefix length. The IPv6 address is any of the forms listed above, and the prefix length is a decimal number that indicates how many bits in front of the IPv6 address is the address prefix.

**There are three kinds of IPv6 addresses, that is, unicast address, multicast address, and anycast address:**

- Unicast address: used to uniquely identify an interface, similar to IPv4 unicast address. The packets sent to one unicast address will be sent to the interface identified by this address.
- Multicast address: used to identify a set of interfaces, similar to IPv4 multicast address. The packets sent to one multicast address are sent to all the interfaces identified by this address.
- Anycast Address: used to identify a group of interfaces and the packet whose destination is an anycast address is sent only to one interface in the group. According to the routing protocol, the interface of receiving the packet is the closest interface from source.

The IPv6 address type is specified by the first few bits of the address, called the format prefix. The corresponding relationship between the main address type and the format prefix is shown in Table 1-2.

Table 6-2 The corresponding relationship between the IPv6 address type and the format prefix

| **Address Type** | | **Format Prefix (Binary)** | **Prefix ID** |
|---|---|---|---|
| Unicast address | Un-specified address | 00...0 (128 bits) | ::/128 |
| | Loopback address | 00...1 (128 bits) | ::1/128 |
| | The local address of the link | 1111111010 | FE80::/10 |
| | Global unicast address | Other forms | - |
| Multicast address | | 11111111 | FF00::/8 |

| Address Type | Format Prefix (Binary) | Prefix ID |
|---|---|---|
| Anycast address | Distributed from the unicast address space, use the format of the unicast address | |

IPv6 unicast addresses can be of various types, including global unicast addresses, link local addresses, and site local addresses.

- The global unicast address is equivalent to the IPv4 public network address, which is provided to the Internet service provider. This type of addresses allows the aggregation of routing prefixes, thus limiting the number of global routing entries.
- Link local addresses are used for the communication between the local nodes on the link in the neighborhood discovery protocol and stateless automatic configuration. The packet using the link local address as the source or destination address is not forwarded to other links.
- Loopback address: Unicast address 0:0:0:0:0:0:0:0:1 (simplified as: 1) is called a loopback address and cannot be assigned to any physical interface. Its function is the same as the loopback address in IPv4, that is, the node sends IPv6 packets to itself.
- Un-specified address: The address "::" is called an unspecified address and cannot be assigned to any node. Before a node obtains a valid IPv6 address, it can be entered in the source address field of the IPv6 packet sent, but not as the destination address of the IPv6 packet.

The special multicast addresses reserved by IPv6 are shown in table 1-3.

Table 6-3 The special multicast address list of IPv6

| Address | Usage |
|---|---|
| FF01::1 | The multicast address of all nodes in the local scope of the node |
| FF02::1 | The multicast address of all nodes in the local scope of the link |
| FF01::2 | The multicast address of all routers in the local scope of the node |
| FF02::2 | The multicast address of all routers in the local scope of the link |

**Configuration Condition**

None

### Configure the IPv6 Address of the Interface

Table 6-4 Configure the IPv6 address of the interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the IPv6 address of the interface | **ipv6 address** { *linklocal-address* **link-local** \| *prefix-address* [ **anycast** \| **eui-64** ] \| **autoconfig** } | Mandatory<br>By default, the interface is not configured with the IPv6 address. |

**Note:**

- One interface can be configured with multiple IPv6 addresses.
- After one interface is configured with the IPv6 address, automatically enable the IPv6 function.

## 6.2.2. Configure IPv6 Basic Functions

### Configuration Condition

None

### Enable IPv6 Unicast Forwarding Function

By default, the IPv6 unicast forwarding function is enabled. In some special cases, the user can disable the IPv6 unicast forwarding function. After disabling the function, do not forward the IPv6 packet.

Table 6-5 Enable IPv6 unicast forwarding function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable IPv6 unicast forwarding function | **ipv6 unicast-routing** | Mandatory<br>By default, the IPv6 unicast forwarding function is enabled. |

### Enable IPv6 Function of the Interface

Before performing the IPv6 configuration on one interface, first enable the IPv6 function. Otherwise, some configuration will not take effect.

Table 6-6 Enable the IPv6 function of the interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the IPv6 function of the interface | **ipv6 enable** | Mandatory<br>By default, the IPv6 function of the interface is disabled. |

### Configure the Hop Limit of the IPv6 Packet

The IPv6 header contains the Hop Limit field, whose function is the same as the TTL field in the IPv4 header, representing the times the packet can be forwarded by the router over the network.

With the command, you can configure the hop limit in the IPv6 packet header generated by the device.

Table 6-7 Configure the hop limit of the IPv6 packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the hop limit of the IPv6 packet | **ipv6 hop-limit** *value* | Mandatory<br>By default, the hop limit of the IPv6 packet sent by the device is 64. |

**Configure the IPv6 MTU of the Interface**

Table 6-8 Configure the IPv6 MTU of the interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the IPv6 MTU of the interface | **ipv6 mtu** *value* | Mandatory<br>By default, do not configure the IPv6 MTU of the interface |

## 6.2.3. Configure IPv6 Neighbor Discovery Protocol

IPv6 Neighbor Discovery protocol includes the following functions: address resolution, neighbor unreachable detection, duplicate address detection, router discovery / prefix discovery, address auto configuration and redirection.

The ICMPv6 packet type used by the ND protocol and its functions are shown in the following table.

Table 6-9 The ICMPv6 packet type used by the ND protocol and its functions

| ICMPv6 Packet Type | Type No. | Function |
|---|---|---|
| Router request packet (RS: Router Solicitation) | 133 | After a node starts, it sends a request to the router v configuration information, used for auto configuratio |
| Router advertisement packet (RA: Router Advertisement) | 134 | Respond for the RS packet<br>Without suppressing the sending of RA packets, the router periodically sends RA packets, including prefix information options and some flag bits. |
| Neighbor request packet (NS: Neighbor Solicitation) | 135 | Get the link-layer address of the neighbor<br>Verify whether the neighbor is reachable<br>Perform the repeated address detection |
| Neighbor advertisement packet (NA: Neighbor Advertisement) | 136 | Respond for the NS packet<br>The node sends the NA packet automatically when the link layer changes, advertising the change information of the node to the neighbor node. |
| Re-direction packet (Redirect) | 137 | When meeting a certain condition, the default gateway sends a redirect packet to the source host, making the host re-select the correct next hop address for subsequent packet transmission. |

- Address resolution

Get the link-layer address of the neighbor node on one link, which is realized by the NS packet and NA packet

- Neighbor unreachable detection

After getting the link-layer address of the neighbor node, verify whether the neighbor node is reachable via the NS packet and NA packet.

1. The node sends the NS packet, whose the destination address is the IPv6 address of the neighbor node
2. If receiving the confirm packet of the neighbor node, it is regarded that the neighbor is reachable. Otherwise, it is regarded that the neighbor is not reachable.

- Duplicate address detection

After the node gets one IPv6 address, it is necessary to use the duplicate address detection function to confirm whether the address is used by other nodes.

- Router discovery/prefix discovery and address auto configuration

Router discovery/prefix discovery indicates the node gets the neighbor router and its network prefix from the received RA packet, as well as other configuration parameters.

Address stateless auto configuration indicates that the node automatically configures the IPv6 address according to the information obtained by router discovery / prefix discovery.

Router discovery/prefix discovery is achieved through RS packets and RA packets.

- Re-direction

When the host starts, there may be only one default route to the default gateway in its routing table. When meeting certain conditions, the default gateway sends ICMPv6 redirect packets to the source host, informing the host to choose a better next hop for sending the subsequent packets.

### Configuration Condition

None

### Configure IPv6 Static Neighbor

Resolving the IPv6 address of the neighbor node into the link layer address can be realized by the address resolution function of the IPv6 ND protocol, or by configuring the static neighbor manually.

The IPv6 neighbor is uniquely identified by the IPv6 address of the neighbor node and the L3 interface connected to the neighbor node.

Table 6-10 Configure the IPv6 static neighbor

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the IPv6 static neighbor | **ipv6 neighbor** *ipv6-address interface-name mac-address* | Mandatory<br>By default, do not configure the IPv6 static neighbor. |

### Configure the Age Time of the IPv6 Neighbor Entry in the STALE State

IPv6 neighbor entries have five reachability states: INCOMPLETE, REACHABLE, STALE, DELAY and PROBE. The STALE state indicates not knowing whether the neighbor is reachable or not. The neighbor entry in STALE state has an aging time, and the neighbor entries in STALE state reaching the aging time will migrate to the DELAY state.

Table 6-11 Configure the age time of the IPv6 neighbor entry in the STALE state

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |

| Step | Command | Description |
|---|---|---|
| Configure the age time of the IPv6 neighbor entry in the STALE state | **ipv6 neighbor stale-aging** *aging-time* | Optional<br>By default, the age time of the IPv6 neighbor entry in the STALE state is 7200s. |

### Configure the Re-transmission Interval of the NS Packet

When the device sends an NS packet, and if it does not receive a response within a specified time interval, it will resend the NS packet. The interval for re-sending NS packet can be configured by the following command.

Table 6-12 Configure the interval of re-sending the NS packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the re-transmission interval of the NS packet | **ipv6 nd ns-interval** *value* | Mandatory<br>By default, the interval of sending the NS packet is 1000ms. |

### Configure the Times of the IPv6 Duplicate Address Detection Sending the NS Packet

After the interface is configured with the IPv6 address, the NS packet is sent for duplicate address detection. If no response is received within a certain period of time, the NS packet is continued to be sent. When the number of NS packets sent reaches the set value, no response is received, the address is considered available.

Table 6-13 Configure the times of the IPv6 duplicate address detection sending the NS packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the times of the IPv6 duplicate address detection sending the NS packet | **ipv6 nd dad attempts** *value* | Mandatory<br>By default, the times of the IPv6 duplicate address detection sending the NS packet is 1. |

### Configure the Related Parameters of the RA Packet

Users can configure whether the interface sends the RA packet and the interval of sending the RA packet according to the actual situation, and can configure the parameters of the RA packet to inform the host. When the host receives the RA message, it can use these parameters to do the corresponding operation.

Table 6-14 The parameters and descriptions in the RA packet

| Parameter | Description |
|---|---|
| Hop Limit | When sending the IPv6 packet, the host will fill the Hop Limit field in the IPv6 header using this parameter value. At the same time, the parameter value is also used as the Hop Limit field value in the device reply packet. |
| MTU | The MTU of the released link, which can be used to ensure that all nodes on one link adopt the same MTU value |
| Router Lifetime | Used to set the time of the router that sends the RA packet serving as the default router of the host. The host can determine whether to take the router sending the RA packet as the default router based on the router lifetime parameter value of the received RA packet. |

| Parameter | Description |
|---|---|
| The time of the neighbor keeping the reachable state (Reachable Time) | When the neighbor reachability detection confirms that the neighbor is reachable, the device assumes that the neighbor is reachable within the set reachable time; if a packet needs to be sent to the neighbor after the set time, reconfirm that the neighbor is reachable. |

Table 6-15 Configure the related parameters of the RA packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the prefix option information in the RA packet | **ipv6 nd prefix** { *ipv6-prefix* \| **default** } [ *valid-lifetime* \| **infinite** \| **no-advertise** \| **no-autoconfig** \| **off-link** ] [ *prefered-lifetime* \| **infinite** ] | Mandatory<br>By default, do not configure the prefix option information. |
| Configure the value of the Hop Limit field in the RA packet sent by the interface to be got from the global configuration | **ipv6 nd ra hop-limit** | Optional<br>By default, do not configure the value of the Hop Limit field in the RA packet sent by the interface to be got from the global configuration, and the value of the Hop Limit field is 0. |
| Configure the maximum interval and minimum interval of sending the RA packet | **ipv6 nd ra interval** *max-value* [ *min-value* ] | Optional<br>By default, the maximum interval of sending the RA packet is 600s, and the minimum interval is 198s. |

| Step | Command | Description |
|---|---|---|
| Configure the RA packet to carry the MTU option | **ipv6 nd ra mtu** | Optional<br>By default, the RA packet does not carry the MTU option. |
| Configure the lifetime of the router in the RA packet | **ipv6 nd ra-lifetime** *value* | Optional<br>By default, the lifetime of the router in the RA packet is 1800s. |
| Prohibit the interface from sending the RA packet periodically | **ipv6 nd suppress-ra period** | Optional<br>By default, the interface does not send the RA packet periodically. |
| Prohibit the interface from replying the RS packet | **ipv6 nd suppress-ra response** | Optional<br>By default, the interface does not reply the RA packet when receiving the RS packet. |

### Enable the Interface to Send the Re-direct Packet

After receiving the IPv6 packet that needs to be forwarded, the device finds that the receiving interface of the packet is the same as the sending interface by selecting the route. At this time, the device forwards the packet and sends back the redirect packet to the source, informing the source to re-select the correct next hop for sending the subsequent packets. By default, a device can send a redirect packet, but in some specific cases, the user can prevent the device from sending a redirect packet.

Table 6-16 Enable the function of the interface sending the re-direct packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | configure terminal | - |
| Enter the interface configuration mode | interface interface-name | - |

| Step | Command | Description |
|---|---|---|
| Enable the function of the interface sending the re-direction packet | ipv6 redirects | Optional<br>By default, the function of the interface sending the re-direction packet is enabled. |

## 6.2.4. Configure ICMPv6 Function

In IPv6 protocol stack, Internet Control Message Protocol is mainly used to provide network detection services, and provide error reports to inform the corresponding devices when the network layer or transport layer protocol is abnormal, so as to control and manage the network.

### Configuration Condition

None

### Configure the Rate of Sending the ICMPv6 Error Packet

If there are too many ICMPv6 error packets sent in the network, it may lead to network congestion. To avoid this, users can configure the maximum number of ICMPv6 error packets sent within a specified time.

Table 6-17 Configure the rate of sending the ICMPv6 error packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the rate of sending the ICMPv6 error packets | **ipv6 icmp error-interval** *interval* [ *buckets* ] | Optional<br>By default, the period of sending the ICMPv6 error packets is 100ms, and the maximum number of the ICMPv6 error packets sent in the period is 10. |

### Enable the Function of Sending the ICMPv6 Packet with Unreachable Destination

The function of sending the ICMPv6 packet with the unreachable destination indicates that after receiving one IPv6 packet and if its destination is reachable, the device discards the packet and sends the ICMPv6 unreachable error packet to the source.

The device will send ICMPv6 unreachable error packet when meeting the following conditions:

- When forwarding packets, and if failed to find the route, the device sends the ICMPv6 error packet "No route to the destination address" to the source.

- When a device forwards a packet, and if it is unable to send it due to a management policy (such as firewall, ACL), it sends an ICMPv6 error packet "the communication with destination address is prohibited by management policy" to the source.
- If the destination IPv6 address of the packet exceeds the range of the source IPv6 address (for example, the source IPv6 address of the packet is the link local address, and the destination IPv6 address of the packet is the global unicast address) when forwarding a packet, and as a result, the packet cannot reach the destination, the device will send the ICMPv6 error packet "out of the source address range" to the source.
- If the device fails to resolve the link layer address of the destination IPv6 address when forwarding the packet, it sends the "address unreachable" ICMPv6 error packet to the source.
- When a device receives an IPv6 packet whose destination address is the local and transport layer protocol is UDP, and if the destination port number of the packet does not match the process in use, it sends a "port unreachable" ICMPv6 error packet to the source.

Because the information transmitted to the user process by ICMPv6 destination unreachable error packet is unreachable, if there is a malicious attack, it may affect the normal use of the terminal users. To avoid these phenomena, the user can disable the function of sending the ICMPv6 destination unreachable error packet.

Table 6-18 Enable the function of sending the ICMPv6 packet with the unreachable destination

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the function of sending the ICMPv6 packet with the unreachable destination | **ipv6 unreachables** | Optional<br>By default, the function of sending the ICMPv6 packet with the unreachable destination is enabled. |

### 6.2.5. **Configure the IPv6 TCP Anti-Attack Function**

If the IPv6 TCP server receives a large number of SYN packets, but the peer does not reply the SYN+ACK response to the server, this will lead to a large amount of memory consumption on the server, occupying the semi-connected queue of the server, and as a result, the IPv6 TCP server cannot serve the normal request. This attack can be avoided by configuring the IPv6 TCP anti-attack function.

### Configuration Condition

None

### Enable IPv6 TCP syncache Function

Instead of rushing to allocate TCB when receiving SYN packets, the function first replies a SYN + ACK packet and stores this semi-open connection information in a dedicated cache until the correct ACK packet is received, and then reallocates the TCB.

Table 6-19 Enable IPv6 TCP syncache function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the IPv6 TCP syncache function | **ipv6 tcp syncache** | Mandatory<br>By default, the IPv6 TCP syncache function is disabled. |

### Enable IPv6 TCP syncookies Function

This function does not use any storage resources at all. It uses a special algorithm to generate Sequence Number. This algorithm takes into account the IPv6 address and port of the peer party, the IPv6 address and port fixed information of one's own party, and some fixed information of one's own party that the peer party cannot know, such as MSS and time. After receiving the ACK packet of the peer party, recalculate it to see whether it is the same as the Sequence Number-1 in the response packet of the peer party, so as to decide whether to allocate TCB resources.

Table 6-20 Enable the IPv6 TCP syncookies function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the IPv6 TCP syncookies function | **ipv6 tcp syncookies** | Mandatory<br>By default, the IPv6 TCP syncookies function is disabled. |

### Enable the Function of Accelerating IPv6 TCP Connection Aging

This function checks the number of IPv6 TCP connections to the current system and compares it with the expected connection threshold; if the number of current accessed IPv6 TCP connections reaches two-thirds of the expected connection threshold, check and accelerate the aging of IPv6 TCP connections to avoid the existence of dead connections and recover resources as soon as possible.

Table 6-21 Enable the function of accelerating the IPv6 TCP connection aging

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the function of accelerating the IPv6 TCP connection aging | **ipv6 tcp naptha threshold** *num* | Mandatory<br>By default, the function is not enabled. |

## 6.2.6. Configure TCP Max. Segment Size

This function sets the maximum segment size of the TCP packet on the interface. When the connection is established through TCP, the mss value is encapsulated into the header to set the maximum segment size of the packet sent by the connection to the mss value.

### Configuration Condition

None

### Configure TCP Packet MSS

The command is used to control the size of the packet transmitted by connection as mss.

Table 6-22 Configure TCP connection mss

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the *mss* value | **ipv6 tcp adjust-mss** *mss-value* | Mandatpry<br>By default, it is not configured. |

### 6.2.7. **Enable ND Quick Pick-up Function**

#### Configuration Condition

None

#### Enable ND Quick Pick-up Function

Table 6-23 Enable the ND quick pick-up function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the ND quick pick-up function | **nd fast-response** | Mnadatory<br>By default, the global ND quick pick-up function is enabled. |

### 6.2.8. **Configure L3 Interface ND Proxy**

#### Configuration Condition

None

#### Configure L3 Interface ND Proxy

Table 6-24 Configure the L3 ND proxy function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the L3 interface configuration mode | **interface** *interface-name* | Mandatory<br>After entering the L3 interface configuration mode, subsequent configuration takes effect only on the current interface. |
| Configure the L3 interface ND proxy function | **nd proxy enable** | Mandatory<br>By default, the interface does not enable the proxy function. |

### 6.2.9. IPv6 Basic Monitoring and Maintaining

Table 6-25 IPv6 basic monitoring and maintaining

| Command | Description |
|---|---|
| **clear ipv6 icmp6stat** | Clear the ICMPv6 statics information |
| **clear ipv6 interface statistics** | Clear the IPv6 packet statistics information of the interface |
| **clear ipv6 mtu** | Clear the MTU information of the IPv6 path |
| **clear ipv6 neighbors** | Clear the IPv6 dynamic neighbor entry |
| **clear ipv6 statistics** | Clear the IPv6 basic statistics information |
| **clear ipv6 tcp syncache statistics** | Clear the syncache statistics information of the IPv6 TCP protocol |

| Command | Description |
|---|---|
| **clear ipv6 tcp6stat** | Clear the IPv6 TCP statistics information |
| **clear ipv6 udp6stat** | Clear the IPv6 UDP statistics information |
| **clear nd fast-response statistics** | Clear the ND quick pick-up statistics |
| **show ipv6 hop-limit** | Show the IPv6 global Hop Limit value |
| **show ipv6 frag-queue** | Show the cached IPv6 fragment packet |
| **show ipv6 icmp6state** | Show the ICMPv6 statistics information |
| **show ipv6 interface** | Show the IPv6 information of the interface |
| **show ipv6 interface statistics** | Show the IPv6 statistics information of the interface |
| **show ipv6 max-mtu** | Show the IPv6 MTU maximum value supported by the system |
| **show ipv6 mtu** | Show the MTU information of the IPv6 path |
| **show ipv6 neighbors** | Show the IPv6 neighbor information |
| **show ipv6 prefix** | Show the IPv6 address prefix information |
| **show ipv6 sockets** | Show the IPv6 socket information |
| **show ipv6 statistics** | Show the IPv6 basic statistics information |
| **show ipv6 tcp syncache detail** | Show the syncache entry information of the IPv6 TCP protocol |
| **show ipv6 tcp syncache statistics** | Show the syncache statistics information of the IPv6 TCP protocol |

| Command | Description |
|---|---|
| **show ipv6 tcp6state** | Show the IPv6 TCP statistics information |
| **show ipv6 udp6state** | Show the IPv6 UDP statistics information |

## 6.3. IPv6 Basic Configuration Example

### 6.3.1. Configure the IPv6 Address of the Interface

#### Network Requirements

- Two devices are connected via the Ethernet port, configure the IPv6 global unicast address for the interface, and verify the connectivity between them.

#### Network Topology

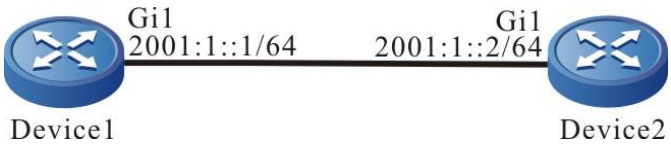



Figure 6-1 Networking for configuring the Ipv6 address of the interface

#### Configuration Steps

**Step 1:** Configure the global unicast address of the interface.

#Configure the global unicast address of Device1 interface gigabitethernet1 as 2001:1::1/64.

```
Device1(config)#interface gigabitethernet1
Device1(config-if-gigabitethernet1)#ipv6 address 2001:1::1/64
Device1(config-if-gigabitethernet1)#exit
```

#Configure the global unicast address of Device2 interface gigabitethernet1 as 2001:1::2/64.

```
Device2(config)# interface gigabitethernet1
Device2(config-if-gigabitethernet1)#ipv6 address 2001:1::2/64
Device2(config-if-gigabitethernet1)#exit
```

**Step 2:** Check the result.

#View the Device1 interface information.

```
Device1#show ipv6 interface gigabitethernet1
gigabitethernet1 is up
  VRF: global
  IPv6 is enable, link-local address is fe80::0201:7aff:fe46:a64d
  Global unicast address(es):
    2001:0001::0001, subnet is 2001:0001::/64
  Joined group address(es):
```

```
    ff02::0001:ff00:0001
    ff02::0001:ff00:0
    ff02::0002
    ff02::0001
    ff02::0001:ff46:a64d
  ND control flags: 0x1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
```

After configuring the IPv6 address, enable the IPv6 protocol function on the interface automatically, generate the local address of the link automatically, and add into the corresponding multicast group.

#View the interface information of Device2.

```
Device2#show ipv6 interface gigabitethernet1
gigabitethernet1 is up
  VRF: global
  IPv6 is enable, link-local address is fe80::0201:7aff:fe22:e222
  Global unicast address(es):
    2001:0001::0002, subnet is 2001:0001::/64
  Joined group address(es):
    ff02::0001:ff00:0002
    ff02::0001:ff00:0
    ff02::0002
    ff02::0001
    ff02::0001:ff22:e222
  ND control flags: 0x1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
```

#Ping the link local address of Device2 fe80::0201:7aff:fe22:e222 on Device1.

```
Device1#ping  fe80::0201:7aff:fe22:e222

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to fe80::201:7aff:fe22:e222 , timeout is 2 seconds:

Output Interface: gigabitethernet1
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/96/483 ms.
```

**Note:**

- When pinging the link local address, it is necessary to specify the egress interface, which is the interface on the same link of the ping link local address.

#On Device1, ping the global unicast address of Device2 2001:1::2.

```
Device1#ping 2001:1::2

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2001:1::2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/36/183 ms.
```

Device1 and Device2 can ping each other.

## 6.3.2. Configure IPv6 Neighbor Discovery

### Network Requirements

- Device and PC belong to one LAN.
- Configure the interface of Device gigabitethernet1 with the EUI-64 address.
- PC gets the IPv6 address prefix via the IPv6 neighbor discovery protocol, configure the IPv6 address according to the got address automatically. Realize the communication of the IPv6 protocol between PC and Device.

### Network Topology

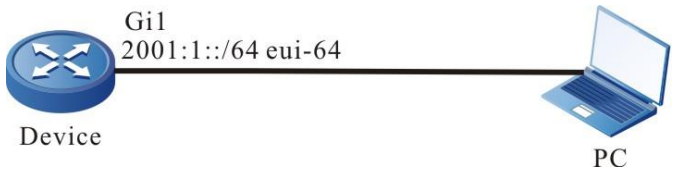



Figure 6-2 Networking for configuring IPv6 neighbor discovery

### Configuration Steps

**Step 1:** Configure the EUI-64 unicast address, and enable the RA advertising function.

#Configure Device gigabitethernet1 with the EUI-64 address, and enable the RA advertising function of gigabitethernet1.

```
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#ipv6 address 2001:1::/64 eui-64
Device(config-if-gigabitethernet1)#no ipv6 nd suppress-ra period
Device(config-if-gigabitethernet1)#no ipv6 nd suppress-ra response
Device(config-if-gigabitethernet1)#exit
```

**Note:**

- By default, the RA advertising function is disabled.

#View the interface information of Device.

```
Device#show ipv6 interface gigabitethernet1
gigabitethernet1 is up
 VRF: global
 IPv6 is enable, link-local address is fe80::0201:7aff:fe5d:e7d3
 Global unicast address(es):
  2001:0001::0201:7aff:fe5d:e7d3, subnet is 2001:0001::/64 [EUI]
 Joined group address(es):
  ff02::0001:ff00:0
  ff02::0002
  ff02::0001
  ff02::0001:ff5d:e7d3
 ND control flags: 0x85
 MTU is 1500 bytes
 ICMP redirects are enabled
 ICMP unreachables are enabled
 ND DAD is enabled, number of DAD attempts: 1
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 (unspecified)
 ND advertised retransmit interval is 0 (unspecified)
 ND config flags is 0x0
 ND MaxRtrAdvInterval is 600
 ND MinRtrAdvInterval is 198
 ND AdvDefaultLifetime is 1800"
```

**Step 2:** Configure PC.

#On the PC, install the Ipv6 protocol. The Ipv6 configuration depends on the operation system. This text takes Windows XP as an example to describe.

```
C:\>ipv6 install
Installing...
```

```
Succeeded.
```

**Step 3:** Check the result.

#View the PC interface information.

```
C:\>ipconfig
……(some displayed information is omitted)
Ethernet adapter 130:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . . . . . . . . : 130.255.128.100
    Subnet Mask . . . . . . . . . . . : 255.255.0.0
    IP Address. . . . . . . . . . . . : 2001:1::15b3:d4:f13d:c3da
    IP Address. . . . . . . . . . . . : 2001:1::3a83:45ff:feef:c724
    IP Address. . . . . . . . . . . . : fe80::3a83:45ff:feef:c724%6
    Default Gateway . . . . . . . . . : fe80::201:7aff:fe5e:cfc1%6
```

You can see that the PC gets the Ipv6 address prefix 2001:1::/64, and generates the global unicast address according to the prefix automatically.

**Note:**

- After the Windows XP host gets the address prefix, it generates two global unicast addresses. The interface ID of one address is generated according to the MAC address of the interface, and the interface ID of the other address is generated randomly.

#On Device, ping the link local address of the PC fe80::3a83:45ff:feef:c724.

```
Device#ping fe80::3a83:45ff:feef:c724

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to fe80::3a83:45ff:feef:c724 , timeout is 2 seconds:

Output Interface: gigabitethernet1
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/29/149 ms.
```

#On Device, ping the auto generated global unicast address 2001:1::15b3:d4:f13d:c3da and 2001:1::3a83:45ff:feef:c724 on the PC.

```
Device#ping 2001:1::15b3:d4:f13d:c3da

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2001:1::15b3:d4:f13d:c3da , timeout is 2 seconds:
!!!!!
```

```
Success rate is 100% (5/5). Round-trip min/avg/max = 0/36/183 ms.


Device#ping 2001:1::3a83:45ff:feef:c724


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2001:1::3a83:45ff:feef:c724 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/26/133 ms.
```

PC and Device can ping each other.

**Note:**

- When pinging the link local address, it is necessary to specify the egress interface, which is the interface on the same link of the ping link local address.

### 6.3.3. Configure L3 ND Proxy

#### Network Requirements

- Device is directly connected with PC1 and PC2 respectively. The network prefix of PC1 and PC2 is the same, which is 2001:1:1:: / 48.
- The MAC address of Device interface gigabitethernet0 is 0001.7a6a.01f0
- Through the device's L3 ND agent, PC1 can ping PC2.

#### Network Topology

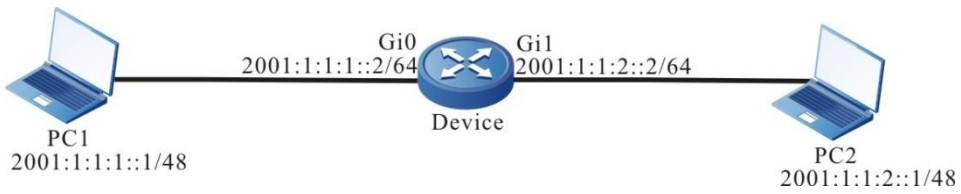



Figure 6-3 Networking of configuring L3 ND proxy

#### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the L3 ND proxy.

#On the L3 interface gigabitethernet0 of Device, enable L3 ND proxy.

```
Device(config)#interface  gigabitethernet0
Device(config-if-gigabitethernet0)#nd proxy enable
Device(config-if-gigabitethernet0)#exit
```

#On the L3 interface gigabitethernet1 of Device, enable L3 ND proxy.

```
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#nd proxy enable
```

```
Device(config-if-gigabitethernet1)#exit
```

**Step 3:** Check the result.

# PC1 ping PC2 address 2001:1:1:2::1.

```
C:\Documents and Settings>ping 2001:1:1:2::1

Pinging 2001:1:1:2::1 with 32 bytes of data:
Reply from 2001:1:1:2::1: bytes=32 time=9971ms TTL=255
Reply from 2001:1:1:2::1: bytes=32 time<1ms TTL=255
Reply from 2001:1:1:2::1: bytes=32 time<1ms TTL=255
Reply from 2001:1:1:2::1: bytes=32 time<1ms TTL=255
Ping statistics for 2001:1:1:2::1
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 997ms, Average = 249ms
```

#View the neighbor table entry of Device.

```
Device#show ipv6 neighbors
IPv6 Address                 Age  Link-layer Addr   flags    State   Interface
2001:1:1:1::1                0    00e0.4c6b.f753    405      STALE   gigabitethernet0
2001:1:1:2::1                1    0857.00da.4715    405      STALE   gigabitethernet1
```

#View the neighbor table entry of PC1.

```
C:\Documents and Settings>netsh interface ipv6 show neighbors
Internet Address                          Physical Address   Type
----------------------------------------- ----------------- ----------
-
2001:1:1:1::2                             00-01-7a-6a-01-f0  Stale (Router)
2001:1:1:2::1                             00-01-7a-6a-01-f0  Stale (Router)
```

#PC1 can ping PC2, and PC1 learned the neighbor table entry of PC2. The MAC address in the neighbor table entries is the MAC address of gigabitethernet0 0001.7a6a.01f0.

**Note:**

- By default, the device does not enable the L3 ND proxy.

# 7. VFR

## 7.1. Overview

IP fragmentation packets may be out-of-order in transmission. Because the out-of-order packets do not have the transmission layer information, these packets cannot be correctly handled by some service modules, such as ACL. To tackle this problem, the VRF (Virtual Fragmentation Reassembly) caches the fragmented packets, analyze the packets, extract the transmission layer information, and provide the information to ACL. The VRF supports anti-tiny fragment attack. If the IP payload length of the first fragment of the arrived packet is less than the length of the transmission layer protocol header, the tiny fragment attack is suffered. The anti-attack function can reduce the system resource consumption.

## 7.2. VFR Function Configuration

Table 7-1 IP VFR function list

| Configuration Task | |
|---|---|
| Enable the IP VFR | Enable the IP VFR |
| Configure the IP VFR parameter | Configure the aging time for the IP VFR |
| | Configure the automatic updating for the IP VFR |
| | Configure the max. cache packets |
| | Configure the max. packets cached by each fragment record |
| Configure the anti-tiny fragment attack | Configure the anti-tiny fragment attack |

### 7.2.1. Enable IP VFR

#### Configuration Condition

None

#### Enable IP VFR

Enable the IP VFR on the inbound interface. When the IP fragmented packet is received, the VFR attempts to analyze the packet fragmentation. If the first fragmentation of the fragmented packet arrives at first, the transmission layer information of the fragmented packet is saved in the fragmentation records. If other fragmentation of the fragmented packet arrives at first (the fragmented packet is out-of-order), the transmission layer information cannot be obtained. The VFR attempts to cache the fragmentation, obtains the transmission layer information when the first fragment of the packet arrives, and then sends out the packet. When the first fragment of the packet does not arrive in a long time, sends out all the cached packets.

Table 7-2 Enable the IP VFR

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the IP VFR | **ip virtual-reassembly** | Mandatory<br>By default, the IP VFR is not enabled. |

**Note:**

When the fragmented packet arrives out of order, the VFR may increase the forwarding delay of the fragmentation, reducing the fragmentation forwarding performance.

### 7.2.2. Configure IP VFR Parameter

The IP VFR parameter can configure the aging time and automatic updating.

#### Configuration Condition

Before configuring the IP VFR parameter, first complete the following task:

- Enable the IP VFR.

#### Configure Aging Time for IP VFR

When the device receives the IP fragmented packet, the fragmentation record is created. When the aging time times out, the fragmentation record is deleted.

Table 7-3 Configure the aging time for the IP VFR

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the aging time for the IP VFR | **ip virtual-reassembly timeout** *seconds* | Mandatory<br>By default, the aging time is 1s. |

#### Configure Auto Updating for IP VFR

If the IP VFR automatic updating is configured, the aging time of the subsequent received fragmented packet will be updated automatically in the aging process.

Table 7-4 Configure the automatic updating for the aging time of the fragmented records

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the automatic updating for the IP VFR | **ip virtual-reassembly auto-update** | Mandatory<br>By default, the automatic updating for the IP VFR is not configured. |

### Configure Max. Packets Cached by IP VFR

After configuring the maximum number of packets cached in IP VFR, if the cached packets exceed the configured maximum in processing fragment packets, the packets will not be cached.

Table 7-5 Configure the maximum cached packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the maximum number of packets cached in IP VFR | **ip virtual-reassembly hold global** *global* | Mandatory<br>By default, the maximum number of the cached packets is 128. |

### Configure Max. Packets Cached by IP VFR Fragmentation Record

After configuring the maximum number of the packets cached by the fragmentation records of the IP VFR, if the cached packets in one fragment record exceed the configured maximum in processing fragment packets, the fragment records will not cache the packet.

Table 7-6 Configure the maximum number of the packets cached by fragment records

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the maximum number of the packets recorded by IP VFR fragmentation records | **ip virtual-reassembly hold single** *single* | Mandatory<br>By default, the maximum number of the packets cached by fragment records is 6. |

### 7.2.3. Configure Anti-tiny Fragment Attack

#### Configuration Condition

Before configuring the anti-tiny fragment attack function, first complete the following task:

- Enable the IP VFR.

#### Configure Anti-tiny Fragment Attack

If the IP payload length of the first fragment of the arrived packet is less than the length of the transmission layer protocol header, the tiny fragment attack is suffered. When a large number of such fragmented packets are received, abundant fragmentation records need to be created, resulting in excessive system resource consumption. If the anti-tiny fragment attack is configured, the fragmentation record for this tiny fragment will not be created, thus reducing the system resource consumption.

Table 7-7 Configure the anti-tiny fragment attack

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the anti-tiny fragment attack | **ip virtual-reassembly arbitrary** | Mandatory<br>By default, the anti-tiny fragment attack is not configured. |

# 8. IPV6 VFR

## 8.1. Overview

IPv6 fragmentation packets may be out-of-order in transmission. Because the out-of-order packets do not have the transmission layer information, these packets cannot be correctly handled by some service modules, such as ACL. To tackle this problem, the IPv6 VRF (ipv6 virtual-reassembly, IPv6 VFR) caches the fragmented packets, analyze the packets, extract the transmission layer information, and provide the information to the ACL and other service modules. IPv6 VRF supports anti-tiny fragment attack function. If the IPv6 payload length of the first packet received is less than the header length of the transport layer protocol, it is considered that it has been attacked by tiny fragment, and the anti-attack function can reduce the consumption of system resources.

## 8.2. IPv6 VFR Function Configuration

Table 8-1 IPv6 VFR function list

| Configuration Task | |
|---|---|
| Enable the IPv6 VFR | Enable the IPv6 VFR |
| Configure the IPv6 VFR parameter | Configure the aging time for the IPv6 VFR |
| | Configure the IPv6 VRF auto updating |
| | Configure the maximum number of the cached packets |
| | Configure the maximum packets cached by each fragmentation record |
| Configure the anti-Tiny Fragment attack | Configure the anti-Tiny Fragment attack |

### 8.2.1. **Enable IPv6 VFR**

#### Configuration Condition

None

#### Enable IPv6 VFR

Enable the IPv6 VFR on the inbound interface. When the IPv6 fragmented packet is received, the VFR attempts to analyze the packet fragmentation. If the first fragmentation of the fragmented packet arrives at first, the transmission layer information of the fragmented packet is saved in the fragmentation records. If other fragmentation of the fragmented packet arrives at first (the fragmented packet is out-of-order), the transmission layer information cannot be obtained. The VFR attempts to cache the fragmentation, obtains the transmission layer

information when the first fragment of the packet arrives, and then sends out the packet. When the first fragment of the packet does not arrive in a long time, sends out all the cached packets.

Table 8-2 Enable the IPv6 VFR

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the IPv6 VFR | **Ipv6 virtual-reassembly** | Mandatory<br>By default, the IPv6 VFR is not enabled. |

**Note:**

When the fragmented packet arrives out of order, the IPv6 VFR may increase the forwarding delay of the fragmentation, reducing the fragmentation forwarding performance.

### 8.2.2. Configure IPv6 VFR Parameter

The IPv6 VFR parameter can configure the aging time and automatic updating and auto updating.

#### Configuration Condition

Before configuring the IPv6 VFR parameter, first complete the following task:

- Enable the IPv6 VFR.

#### Configure Aging Time for IPv6 VFR

When the device receives the IPv6 fragmented packet, the fragmentation record is created. When the aging time times out, the fragmentation record is deleted.

Table 8-3 Configure the aging time for the IPv6 VFR

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the aging time for the IPv6 VFR | **Ipv6 virtual-reassembly timeout** *seconds* | Mandatory<br>By default, the aging time is 1s. |

#### Configure Auto Updating of IPv6 VFR

If the IPv6 VFR auto updating is configured, the aging time of the subsequent received fragmented packet will be updated automatically in the aging process.

Table 8-4 Configure the auto updating for the aging time of the fragmentation records

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure IPv6 VFR auto updating | **ipv6 virtual-reassembly auto-update** | Mandatory<br>By default, do not configure the IPv6 VFR auto updating. |

### Configure Max. Packets Cached by IPv6 VFR

After configuring the maximum number of packets cached in IPv6 VFR, if the cached packets exceed the configured maximum in processing fragment packets, the packets will not be cached.

Table 8-5 Configure the maximum cached packets

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the maximum number of the packets cached by IPv6 VFR | **ipv6 virtual-reassembly hold global** *global* | Mandatory<br>By default, the maximum number of the cached packets is 128. |

### Configure Max. Packets Cached by IPv6 VFR Fragmentation Record

After configuring the maximum number of the packets cached by the fragmentation records of the IPv6 VFR, if the cached packets in one fragment record exceed the configured maximum in processing fragment packets, the fragment records will not cache the packet.

Table 8-6 Configure the maximum number of the packets cached by fragment records

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the maximum number of the packets cached by the fragmentation records of the IPv6 VFR | **ipv6 virtual-reassembly hold single** *single* | Mandatory<br>By default, the maximum number of the packets cached by the fragmentation records is 6. |

### 8.2.3. Configure Anti-tiny Fragment Attack

#### Configuration Condition

Before configuring the anti-tiny fragment attack function, first complete the following task:

- Enable the IPv6 VFR.

#### Configure Anti Tiny Fragment Attack

If the IPv6 payload length of the first received fragment packet is less than the length of the transmission layer protocol header, the tiny fragment attack is suffered. When a large number of such fragmented packets are received, abundant fragmentation records need to be created, resulting in excessive system resource consumption. If the anti-tiny fragment attack is configured, the fragmentation record for this tiny fragment will not be created, thus reducing the system resource consumption.

Table 8-7 Configure the anti-tiny fragment attack

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the anti-tiny fragment attack | **ipv6 virtual-reassembly arbitrary** | Mandatory<br>By default, do not configure the anti-tiny fragment attack. |

# 9. GRE

## 9.1. Overview

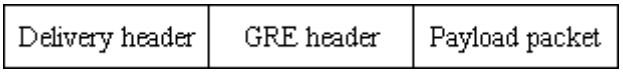
Generic Routing Encapsulation (GRE) is a generic tunnel encapsulation protocol which defined how to use a network protocol to encapsulate another network protocol.

GRE tunnel is one tunnel technology among a lot of tunnel technologies. The starting point and end point of a tunnel need to be manually configured. The tunnel is a virtual end-to-end connection. It provides a transmission channel for the encapsulated packets. The two ends of the tunnel encapsulates and de-capsulates data packets respectively.

- GRE encapsulation:

If a data packet needs to be transmitted through a GRE tunnel, the tunnel adds a GRE header to the packet header, and adds an IP header to the GRE header. Set the protocol number of the IP header to 47 (GRE protocol number in the IP header), set the source address of the IP header to the source address of the tunnel, and set the destination address of the IP header to the destination address of the tunnel.

- GRE packet structure




Payload packet: The network layer packet (such as IP packet) before it enters the tunnel is taken as the valid payload of the tunnel packet. The protocol of the packet is called the passenger protocol of the GRE tunnel.

GRE header: It refers to the GRE header that is added to the payload packet after the payload packet enters the tunnel. The GRE header contains the GRE protocol and some information related to the passenger protocol.

Delivery header: The encapsulated external protocol header (such as IP header) is the header of the protocol for the network in which the tunnel is located. It is a transmission tool which helps one protocol packet to traverse the network of another protocol.

- GRE packet forwarding

After a packet is encapsulated at the starting point of a GRE tunnel, it selects a route according to the destination after encapsulation, and then it is sent out through the corresponding network interface. Intermediate devices take the packet as a common packet until the packet reaches the end of the tunnel.

- GRE decapsulation:

Decapsulation is the reverse process of encapsulation. After the end point of the tunnel receives the packet, it analyzes Delivery header. If the end point of the tunnel finds that the destination is its own address, it checks the protocol field of the IP header. If the protocol field is 47 (GRE protocol number), the end point of the tunnel hands over the packet to the GRE tunnel for processing. The tunnel first removes Delivery header and then checks the protocol number, checksum, and keyword in the GRE header. After required processing, the tunnel removes the GRE header, and hands over the Payload packet to the passenger protocol for later processing. Then, the decapsulation is completed.

## 9.2. GRE Function Configuration

Table 10-1 GRE function list

| Configuration Tasks | |
|---|---|
| Configure a GRE tunnel. | Configure a GRE tunnel. |

### 9.2.1. Configure a GRE Tunnel

#### Configuration Condition

Before configuring a GRE tunnel, ensure that:

- Interface IP addresses have been configured so that neighbor nodes are reachable at the network layer.
- Tunnel interfaces have been created and the basic parameters have been configured. (Refer to tunnel interface configuration manual.)
- A unicast protocol has been configured so that the routes at the two ends of the tunnel are reachable.

#### Configure a GRE Tunnel

Table 10-2 Configuring a GRE tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode. | | **configure terminal** | - |
| Enter the tunnel interface configuration mode. | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address. | Configure IPv4 unicast address | **ip address** *ip-address* { *mask* \| *mask-length* } | Mandatory.<br>By default, no address is configured for the tunnel interface. |
| | Configure the IPv6 global unicast address or anycast address or local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig** } | |

| Step | | Command | Description |
|---|---|---|---|
| Configure the tunnel interface address. | Configure the local address of the IPv6 link | **ipv6 address** *IPv6-address* **link-local** | Optional<br>By default, after the interface enables IPv6, generate the local address of the link automatically. |
| Configure the tunnel interface mode to GRE | | **tunnel mode gre** [ **ip \| ipv6** ] | Optional.<br>By default, the tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface name on the tunnel interface. | | **tunnel source** { *ip-address* \| *ipv6-address* \| *interface-name* } | Mandatory.<br>By default, the source address and interface name are not configured on the tunnel interface. |
| Configure the destination address or host name on the tunnel interface. | | **tunnel destination** { *ip-address*\| *ipv6-address* } | Mandatory.<br>By default, the destination address is not configured on the tunnel interface. |
| Configure the verification function on the tunnel interface | | **tunnel checksum** | Optional<br>By default, the verification function on the tunnel interface is not configured.<br>The verification function cannot be configured at the both ends of the tunnel. |

| Step | Command | Description |
|---|---|---|
| Configure the keywords on the tunnel interface | **tunnel key** *key-number* | Optional<br>By default, the keyword on the tunnel interface is not configured.<br>The keywords on the both ends of the tunnel must be consistent. Otherwise, the tunnel transmission will fail. |
| Configure the keep-alive action on the tunnel interface | **keepalive** [ *period* [ *retries* ] ] | Optional<br>By default, the keep-alive function on the tunnel interface is not configured. |

**Note:**

- On the two end of a tunnel, the source addresses and destination addresses must be configured. The source address of one end is the destination end of the other end.
- In configuring the source of a tunnel, if the interface mode is adopted, the primary address of the source interface is used as the source address of the tunnel.
- The two ends of a tunnel must be configured with the same tunnel mode; otherwise, transmission through the tunnel fails.
- The keep-alive function only takes effect on the GRE over IPv4 tunnel. When the keep-alive function is configured, the tunnel will send the keep-alive packet periodically. If consecutive N keep-alive packets (N indicates the configured retries value) are not responded, the peer end is considered as not working normally. Then, disable the local tunnel interface. The tunnel will also send the keep-alive packet periodically when the tunnel is disabled. The keep-alive function is not required to be configured at the both ends of the tunnel.

### 9.2.2. GRE Monitoring and Maintaining

Table 10-3 GRE monitoring and maintaining

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***]** | Displays the configuration information of all tunnels or a specified tunnel. |

## 9.3. GRE Typical Configuration Example

### 9.3.1. Configure GRE Basic Functions

#### Network Requirements

- IP Network1 and IP Network2 are two private networks of Device1 and Device3.
- IP Network1 and IP Network2 communicate with each other through the GRE tunnel between Device1 and Device3.

#### Network Topology

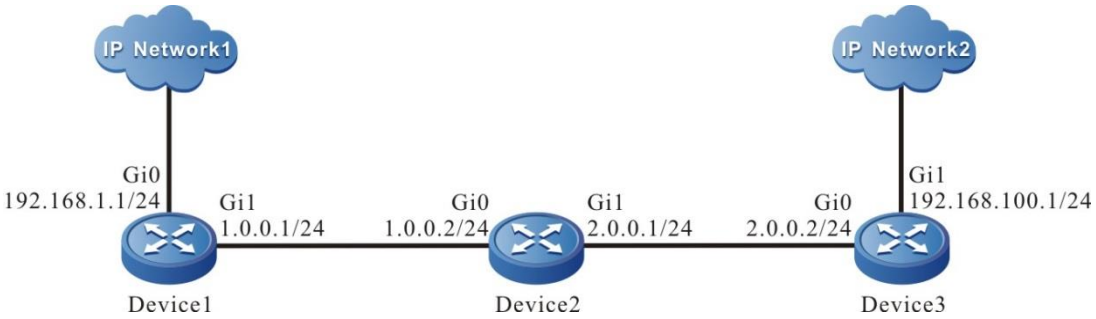



Figure 9-1 Networking for configuring GRE basic functions

#### Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure Open Shortest Path First (OSPF).

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
```

```
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#Query the routing table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:18:40, gigabitethernet0
C   2.0.0.0/24 is directly connected, 00:22:27, gigabitethernet0
C   192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1
```

**Note:**

- The method for querying the routing table of Device 1 and Device2 is the same as that for Device 3, so the processes are not described here.

**Step 3:** Configure a GRE tunnel.

#On Device1, configure GRE tunnel tunnel1, set the source address to 1.0.0.1, destination address to 2.0.0.2, and IP address to 10.0.0.1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#tunnel destination 2.0.0.2
Device1(config-if-tunnel1)#ip address 10.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#exit
```

#On Device3, configure GRE tunnel tunnel1, set the source address to 2.0.0.2, destination address to 1.0.0.1, and IP address to 10.0.0.2.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel source 2.0.0.2
Device3(config-if-tunnel1)#tunnel destination 1.0.0.1
Device3(config-if-tunnel1)#ip address 10.0.0.2 255.255.255.0
Device3(config-if-tunnel1)#exit
```

#Query the GRE tunnel information of Device3.

```
Device3#show tunnel 1
```

Tunnel 1:

Tunnel mode is gre ip

Gre checksum validation is disabled

Gre key is not set

Gre keepalive is disabled

Source ipv4 address is 2.0.0.2 (Source ipv4 address is up on source interface gigabitethernet0)

Destination ipv4 address is 1.0.0.1

Tunnel state is up

Encapsulation vrf is global(0x0)

TTL(time-to-live) is 255

TOS(type of service) is not set

total(1)

**Note:**

- The method for querying the GRE tunnel information of Device 1 is the same as that for Device 3, so the process is not described here.
- If a tunnel is located at different network segments, both the two devices at the two ends of the tunnel must be configured with a static route that reaches the peer tunnel with the tunnel interface as the output interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to the egress interface tunnel1 of IP Network2.

Device1(config)#ip route 192.168.100.0 255.255.255.0 tunnel1

#On Device3, configure the static route to the egress interface tunnel1 of IP Network1.

Device3(config)#ip route 192.168.1.0 255.255.255.0 tunnel1

#View the route table of Device3.

Device3#show ip route

Codes: C - connected, S - static, R - RIP, O - OSPF, OE-OSPF External, M - Management

D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O 1.0.0.0/24 [110/65536] via 2.0.0.1, 00:43:30, gigabitethernet0

C 2.0.0.0/24 is directly connected, 00:47:17, gigabitethernet0

C 10.0.0.0/24 is directly connected, 00:17:12, tunnel1

S 192.168.1.0/24 [1/100000] is directly connected, 00:00:10, tunnel1

```
C   192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1
```

**Note:**

- The method for querying the routing table of Device 1 is the same as that for Device 3, so the process is not described here.

## 9.3.2. Configure GRE over IPv6 Basic Functions

### Network Requirements

- IP Network1 and IP Network2 are the private IP networks of Device1 and Device3 respectively.
- IPv6 Network1 and IPv6 Network2 are the private IPv6 networks of Device1 and Device3 respectively.
- IP Network1 and IP Network2 communicate with each other through the GRE over IPv6 tunnel between Device1 and Device3.
- IPv6 Network1 and IPv6 Network2 communicate with each other through the GRE over IPv6 tunnel between Device1 and Device3.

### Network Topology

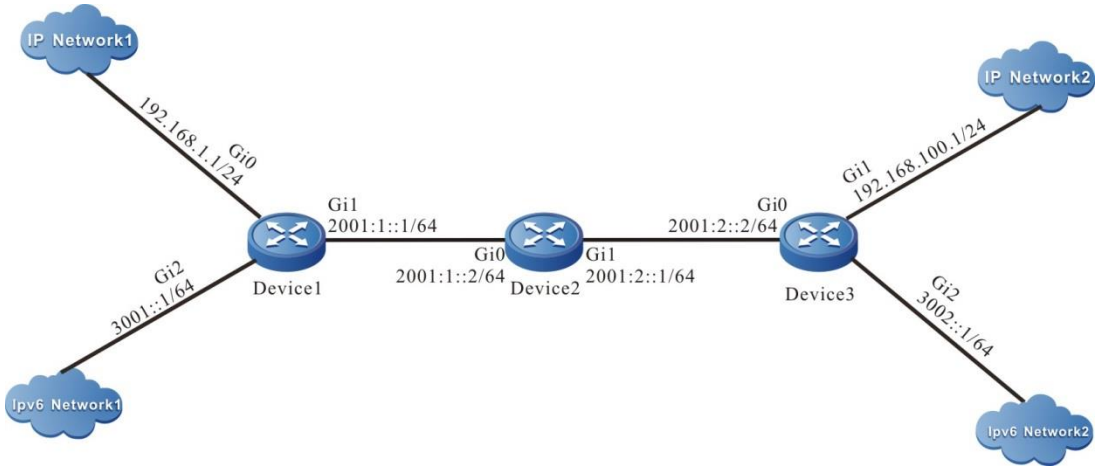



Figure 9-2Networking of configuring GRE over IPv6 basic functions

### Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure OSPFv3, making Device1, Device2, Device3 be able to communicate with each other.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#ipv6 router ospf 100
Device1(config-ospf6)#router-id 1.1.75.1
Device1(config-ospf6)#exit
Device1(config)#interface gigabitethernet1
Device1(config-if-gigabitethernet1)#ipv6 router ospf tag 100 area 0
```

```
Device1(config-if-gigabitethernet1)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#ipv6 router ospf 100
Device2(config-ospf6)#router-id  1.2.75.1
Device2(config-ospf6)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#ipv6 router ospf tag 100 area 0
Device2(config-if-gigabitethernet0)#exit
Device2(config)#interface gigabitethernet1
Device2(config-if-gigabitethernet1)#ipv6 router ospf tag 100 area 0
Device2(config-if-gigabitethernet1)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#ipv6 router ospf 100
Device3(config-ospf6)#router-id 1.1.73.1
Device3(config-ospf6)#exit
Device3(config)#interface gigabitethernet0
Device3(config-if-gigabitethernet0)#ipv6 router ospf tag 100 area 0
Device3(config-if-gigabitethernet0)#exit
```

#View the IPv6 route table of Device3.

```
Device3#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management


L   ::1/128 [0/0]
    via ::, 6w0d:23:09:31, lo0
O   2001:1::/64 [110/2]
    via fe80::508b:fff:fee4:ff6, 00:08:37, gigabitethernet0
C   2001:2::/64 [0/0]
    via ::, 00:15:51, gigabitethernet0
L   2001:2::2/128 [0/0]
    via ::, 00:15:50, lo0
C   3002::/64 [0/0]
    via ::, 00:15:06, gigabitethernet2
```

```
L   3002::1/128 [0/0]
    via ::, 00:15:04, lo0
```

**Note:**

- The method for querying the routing table of Device 1 and Device2 is the same as that for Device 3, so the processes are not described here.

**Step 3:** Configure GRE over IPv6 tunnel.

#Configure GRE over IPv6 Tunnel 1 on Device1, the source address is 2001:1:: 1, the destination address is 2001:2:: 2, the IP address is 10.0.0.1, and the IPv6 address is 10:: 1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode gre IPv6
Device1(config-if-tunnel1)#tunnel source 2001:1::1
Device1(config-if-tunnel1)#tunnel destination 2001:2::2
Device1(config-if-tunnel1)#ip address 10.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#ipv6 address 10::1/64
Device1(config-if-tunnel1)#exit
```

#Configure GRE over IPv6 Tunnel 1 on Device3 with source address of 2001:2:: 2, destination address of 2001:1:: 1, IP address of 10.0.0.2 and IPv6 address of 10:: 2.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode gre IPv6
Device3(config-if-tunnel1)#tunnel source 2001:2::2
Device3(config-if-tunnel1)#tunnel destination 2001:1::1
Device3(config-if-tunnel1)#ip address 10.0.0.2 255.255.255.0
Device3(config-if-tunnel1)#ipv6 address 10::2/64
Device3(config-if-tunnel1)#exit
```

#View the GRE tunnel information of Device3.

```
Device3#show tunnel 1


Tunnel 1:
  Tunnel mode is gre ipv6
    Gre checksum validation is disabled
    Gre key is not set
  Source ipv6 address is 2001:2::2(Source ipv6 address is up on source interface gigabitethernet0)
  Destination ipv6 address is 2001:1::1
  Tunnel state is up
  Encapsulation vrf is global(0x0)
  TTL(time-to-live) is 255
  TOS(type of service) is not set
```

total(1)

**Note:**

- The viewing method of Device1 is the same as that of Device3, and the viewing process is omitted.
- When the tunnel is not in the same network segment, the static route to the peer end tunnel needs to be configured on the devices at both ends of the tunnel, and the egress interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to the egress interface of IP Network2 tunnel1.

```
Device1(config)#ip route 192.168.100.0 255.255.255.0 tunnel1
```

#On Device1, configure the static route to the egress interface of IPv6 Network2 tunnel1.

```
Device1(config)#IPv6 route 3002::/64 tunnel1
```

#On Device3, configure the static route to the egress interface of IP Network1 tunnel1.

```
Device3(config)#ip route 192.168.1.0 255.255.255.0 tunnel1
```

#On Device3, configure the static route to the egress interface of IPv6 Network1 tunnel1

```
Device3(config)#IPv6 route 3001::/64 tunnel1
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C   10.0.0.0/24 is directly connected, 00:17:12, tunnel1
S   192.168.1.0/24 [1/100000] is directly connected, 00:00:10, tunnel1
C   192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1
```

#View the IPv6 route table of Device3.

```
Device3#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
    U - Per-user Static route
    O - OSPF, OE-OSPF External, M - Management

L   ::1/128 [0/0]
    via ::, 6w0d:23:50:28, lo0
```

C 10::/64 [0/0]
via ::, 00:12:23, tunnel1
L 10::2/128 [0/0]
via ::, 00:12:22, lo0
O 2001:1::/64 [110/2]
via fe80::508b:fff:fee4:ff6, 00:49:34, gigabitethernet0
C 2001:2::/64 [0/0]
via ::, 00:56:48, gigabitethernet0
L 2001:2::2/128 [0/0]
via ::, 00:56:46, lo0
S 3001::/64 [1/100000]
via ::, 00:00:14, tunnel1
C 3002::/64 [0/0]
via ::, 00:56:02, gigabitethernet2
L 3002::1/128 [0/0]
via ::, 00:56:01, lo0

**Note:**

- The viewing method of Device1 is the same as that of Device3, and the viewing process is omitted.

# 10. IPIP

## 10.1. Overview

The IPIP (IPv4 over IPv4) tunnel is one of the tunnel technologies. Similar to the GRE (Generic Routing Encapsulation) tunnel, the start and end of the IPIP tunnel are manually configured. It serves a virtual PTP link and provides a transmission tunnel for the encapsulated packet. The start and end of the tunnel encapsulate and decapsulated the data packet. The IPIP tunnel encapsulates only the IPv4 data packets and enables the encapsulated data packets to be transmitted over other IPv4 network.

- IPIP encapsulation
- When the IP data packet is transmitted over the IPIP tunnel, the tunnel adds an IP packet header. In the IP packet header, the protocol number is set to 4, the source IP address is set to the source IP address of the tunnel, and the destination IP address is set to the destination IP address of the tunnel.
- IPIP packet structure

| Outer IP header | IP header | IP Payload |
|---|---|---|

IP Payload: indicates the IP packet payload before the packet entering the tunnel. It is a valid payload of the tunnel packet.

IP header: indicates the IP packet heeder before the packet entering the tunnel.

Outer IP header: indicates the encapsulated outer IP packet header. It is a transmission tool which enables the IP packet to be transmitted over another IP network.

- IPIP packet forwarding

After the packet is encapsulated at the start of the IPIP tunnel, a routing is selected for the packet based on the encapsulated destination IP address and then the packet is sent out from the corresponding network interface. The intermediate equipment forwards the packet as the common IP packet until the packet reaches the end of the tunnel.

- IPIP decapsulation

Contrary to the encapsulation process, in the decapsulation process, the end of the tunnel first analyzes the outer IP header when it receives the packet. If the destination IP address is the IP address of the end of the tunnel, check the protocol field of the IP packet header. If the protocol field is 4, transmit the packet to the IPIP tunnel for handling. The tunnel removes the outer IP header of the packet and chooses a routing for the packet based on the destination IP address of the decapsulated packet. The subsequent handling is performed based on the routing result.

## 10.2. IPIP Function Configuration

Table 10-1 IPIP function list

| Configuration Task | |
|---|---|
| Configure the IPIP tunnel | Configure the IPIP tunnel |

### 10.2.1. Configure IPIP Tunnel

#### Configuration Condition

Before configuring the IPIP tunnel, first complete the following tasks:

- Configure the interface IP address, enabling that the adjacent nodes on the network layer are reachable.
- Create the tunnel interface and configure the basic parameters. For details, refer to the *Tunnel Interface Configuration Manual*.
- Configure any unicast routing protocol, enabling the routing to the both ends of the tunnel is reachable.

#### Configure IPIP Tunnel

Table 10-2 Configure the IPIP tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the tunnel interface configuration mode | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface IP address | **ip address** *ip-address* { *mask* \| *mask-length* } | Mandatory<br>By default, the IP address on the tunnel interface is not configured. |
| Configure the tunnel interface mode as IPv4 over IPv4 | **tunnel mode ipip** | Mandatory<br>By default, the tunnel interface mode is GRE over IPv4. |

| Step | Command | Description |
|---|---|---|
| Configure the source IP address or the interface name for the tunnel interface | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, the source IP address and the interface name are not configured for the tunnel interface. |
| Configure the destination IP address or the host name for the tunnel interface | **tunnel destination** { *ip-address* } | Mandatory<br>By default, the destination IP address is not configured for the tunnel interface. |

**Note:**

- The start and end of the tunnel must be configured with the source IP address and the destination IP address. The IP address at the start and end of the tunnel are complementary source IP address and destination IP address for each other.
- When configuring the tunnel source IP address, the source IP address of the tunnel adopts the master IP address of the source IP address if the interface mode is used.
- The start and end of the tunnel must be configured with the same tunnel mode. Otherwise, the tunnel transmission fails.
- Two or more tunnels with the same tunnel mode, source IP address, and destination IP address cannot be configured on the same device.

### 10.2.2. IPIP Monitoring and Maintaining

Table 10-3 The IPIP monitoring and maintaining

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***]** | Display the configuration information of all tunnels and the specified tunnel |

## 10.3. IPIP Typical Configuration Example

### 10.3.1. Configure IPIP Basic Function

#### Network Requirements

- IP Network1 and IP Network2 are two private networks for Device1 and Device3, respectively.
- IP Network1 communicates with IP Network2 through the IPIP tunnel between Device1 and Device3.

## Network Topology

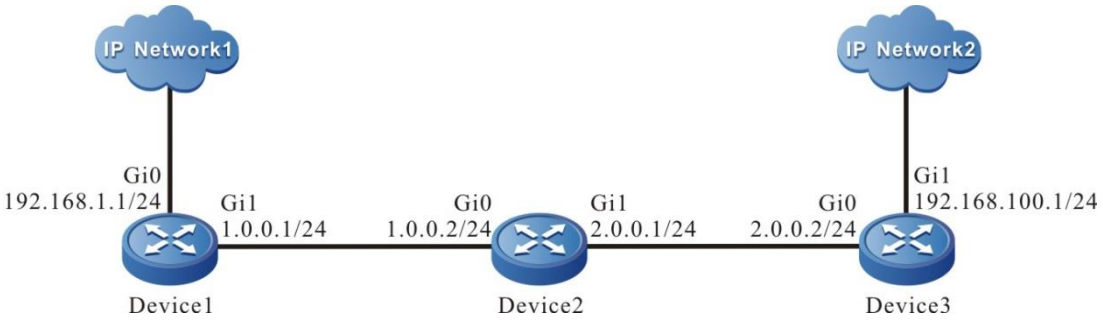



Figure 10-1 Networking of configuring the IPIP basic function

## Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the OSPF.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#View the routing table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:18:40, gigabitethernet0
```

C 2.0.0.0/24 is directly connected, 00:22:27, gigabitethernet0

C 192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1

**Note:**

- Device1 and Device2 are viewed in the same way as Device3. The viewing process is omitted.

**Step 3:** Configure the IPIP tunnel.

#Configure the IPIP tunnel, tunnel1 on Device1 with source IP address as 1.0.0.1, destination IP address as 2.0.0.2, and the IP address as 10.0.0.1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode ipip
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#tunnel destination 2.0.0.2
Device1(config-if-tunnel1)#ip address 10.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#exit
```

#Configure the IPIP tunnel, tunnel1 on Device3 with source IP address as 2.0.0.2, destination IP address as 1.0.0.1, and the IP address as 10.0.0.2.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode ipip
Device3(config-if-tunnel1)#tunnel source 2.0.0.2
Device3(config-if-tunnel1)#tunnel destination 1.0.0.1
Device3(config-if-tunnel1)#ip address 10.0.0.2 255.255.255.0
Device3(config-if-tunnel1)#exit
```

#View the IPIP tunnel information of Device3.

```
Device3#show tunnel 1

Tunnel 1:
  Tunnel mode is ipip
  Source ipv4 address is 2.0.0.2(Source ipv4 address is up on source interface gigabitethernet0)
  Destination ipv4 address is 1.0.0.1
  Tunnel state is up
  Encapsulation vrf is global(0x0)
  TTL(time-to-live) is 255
  TOS(type of service) is not set
total(1)
```

**Note:**

- Device1 is viewed in the same way as Device3. The viewing process is omitted.

- When the tunnel does not exist in the same network segment, devices at the both ends of the tunnel are configured with a static routing to the peer end. And the outbound interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to the egress interface tunnel1 of IP Network2.

```
Device1(config)#ip route 192.168.100.0 255.255.255.0 tunnel1
```

#On Device3, configure the static route to the egress interface tunnel1 of IP Network1.

```
Device3(config)#ip route 192.168.1.0 255.255.255.0 tunnel1
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:43:30, gigabitethernet0
C   2.0.0.0/24 is directly connected, 00:47:17, gigabitethernet0
C   10.0.0.0/24 is directly connected, 00:17:12, tunnel1
S   192.168.1.0/24 [1/100000] is directly connected, 00:00:10, tunnel1
C   192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1
```

**Note:**

- Device1 is viewed in the same way as Device3. The viewing process is omitted.

# 11. TRANSITION TUNNEL

## 11.1. Overview

The transition tunnel (IPv6 over IPv4) technology provides a way to transfer the IPv6 data with the existing IPv4 routing system: the IPv6 packet is encapsulated in the IPv4 packet as unstructured data and transmitted through the IPv4 network. According to the different setup modes, the transitional tunnels can be divided into manual tunnels and automatic tunnels. The transition tunnel technology skillfully utilizes the existing IPv4 network, its significance lies in providing a way to enable IPv6 nodes to communicate during the transition period, but it cannot solve the intercommunication problem between IPv6 nodes and IPv4 nodes. Transition tunnels are divided into manual type and automatic type. The corresponding relationship between the types and the modes is as follows:

Table 11-1 Transition tunnel mode

| Tunnel Type | Tunnel Mode | Tunnel Source/Destination Address | Tunnel Interface Address |
|---|---|---|---|
| Manual tunnel | IPv6 over IPv4 manual tunnel | The source/destination address is the manual configured IPv4 address | IPv6 address |
| Auto tunnel | IPv4 compatible IPv6 auto tunnel | The source address is the manual configured IPv4 address, and the destination address does not need to be configured. | IPv4 compatible IPv6 address, its format: ::a.b.c.d/96 |
| | 6to4 tunnel | | 6to4 address, the format is:<br>2002:a.b.c.d::/48 |
| | ISATAP tunnel | | ISATAP address, the format is:<br>Prefix:0:5EFE:a.b.c.d/64 |

- IPv6 over IPv4 manual tunnel

This kind of tunnel is established manually. The terminal address of the tunnel is determined by the configuration. It is not necessary to assign special IPv6 addresses to the nodes. It is suitable for the IPv6 nodes that often communicate.

- IPv4 compatible with IPv6 auto tunnel

IPv4 compatible IPv6 auto tunnel is a point-to-multipoint tunnel. A special IPv6 address is used at both ends of the tunnel. Its format is: a.b.c.d/96, in which a.b.c.d is the IPv4 address. In the process of tunnel encapsulation, the embedded IPv4 address is automatically used as the end of the tunnel, which makes the establishment of the tunnel very convenient. However, due to the fact that the IPv4 compatible IPv6 address still depends on the IPv4 address in the

application, this limitation can not be changed. Therefore, IETF has abandoned this kind of address in the new standard, and this kind of tunnel will be phased out.

- 6to4 tunnel

The 6to4 tunnel is a point-to-multipoint tunnel. Special IPv6 addresses are required at both ends of the tunnel in the format of 2002:a.b.c.d:/48, in which 2002 represents a fixed IPv6 address prefix and a.b.c.d represents the unique 32-bit IPv4 address corresponding to the 6to4 tunnel. This embedded IPv4 address is automatically used as the end point of the tunnel in the process of tunnel encapsulation, which makes the establishment of the tunnel very convenient. Therefore, the nodes using the 6to4 mechanism must have at least one unique IPv4 address in the world. This mechanism is suitable for the intercommunication between the nodes running IPv6. Since the first 48 bits in the IPv6 address prefix of 6to4 have been determined by the fixed number plus the IPv4 address, the remaining 16 bits subnet number can be defined by the user himself, which makes it more flexible to use the IPv4 network to realize the interconnection of the IPv6 network. 6to4 overcomes the limitation of IPv4 compatible IPv6 automatic tunnel.

- ISATAP tunnel

ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) tunnels are point-to-multipoint automatic tunnels. Special IPv6 addresses are required at both ends of the tunnel in the format of Prefix: 0:5EFE: a.b.c.d/64, where Prefix represents any valid IPv6 unicast address prefix, and a.b.c.d represents 32-bit IPv4 address (not required to be globally unique). The embedded IPv4 address is automatically used as the end point of the tunnel during the tunnel encapsulation, so this kind of tunnel is also built automatically.

- Transition tunnel encapsulation

When IPv6 packets are sent through transitional tunnels, an IP header is added to the tunnel header, the protocol number in the IP header is 41, the source address in the IP header is set to the source address of the tunnel, and the destination address in the IP header is set according to the type of the tunnel: if it is a manual tunnel, it is set as the configured tunnel destination address, and if it is an automatic tunnel, it is set as the IPv4 address embedded in the IPv6 address.

- The structure of the transition tunnel packet

| IPv4 Header | IPv6 Header | IPv6 Packet Payload |
|---|---|---|

IPv6 Packet Payload: The payload of the IPv6 packet before entering the tunnel, which serves as the valid payload of the tunnel packet.

IPv6 Header: The header of the IPv6 packet before entering the tunnel.

IPv4 Header: The encapsulated outer IPv4 header, which is the transmission tool of the IPv6 packet across the IPv4 network.

- The forwarding of the transition tunnel packet

After the packet is encapsulated at the beginning of the transition tunnel, select the route according to the encapsulated destination address, and then, send the packet from the corresponding network interface. The intermediate device forwards it as an ordinary IP packet until the packet reaches the end of the tunnel.

- The encapsulation/de-capsulation of the transition tunnel packet

The de-capsulation process and the encapsulation process are opposite. The tunnel end first analyzes the IPv4 header after receiving the packet. If the destination address is its own address, check the protocol field of the IP header. If the protocol field is 41, hand over the packet to the transition tunnel for processing. After the tunnel removes the IPv4 header of the packet, select the route according to the destination address of the packet after de-capsulation, and perform the subsequent processing according to the result of the route selection.

## 11.2. Transition Tunnel Function Configuration

Table 11-2 Transition tunnel function configuration list

| Configuration task | |
|---|---|
| Configure IPv6 over IPv4 manual tunnel | Configure IPv6 over IPv4 manual tunnel |
| Configure the IPv4 compatible IPv6 auto tunnel | Configure the IPv4 compatible IPv6 auto tunnel |
| Configure the 6to4 tunnel | Configure the 6to4 tunnel |
| Configure the ISATAP tunnel | Configure the ISATAP tunnel |

**Note:**

- For the configuration commands of the transition tunnel, refer to the chapter of IP Tunnel in the configuration manual.

### 11.2.1. Configure the IPv6 over IPv4 Manual Tunnel

#### Configuration Conditions

Before configuring the IPv6 over IPv4 manual tunnel, complete the following tasks:

- Configure the IP address of the physical interface, making the neighboring nodes reachable at the network layer
- Create one tunnel interface, and configure the basic parameters (refer to the configuration manual of the tunnel interface)
- Configure any unicast routing protocol, making the route at the two ends of the tunnel reachable

### Configure IPv6 over IPv4 Manual Tunnel

Table 11-3 Configure IPv6 over IPv4 manual tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode | | **configure terminal** | - |
| Enter the tunnel configuration mode | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address | Configure IPv6 global unicast address or anycast address or the local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig }** | Mandatory<br>By default, do not configure the IPv6 address on the tunnel interface. |
| | Configure the local address of the IPv6 link | **ipv6 address** *ipv6-address* **link-local** | Optional<br>By default, after enabling the IPv6 on the interface, automatically generate the local address of the link. |
| Configure the tunnel interface mode as the IPv6 over IPv4 manual tunnel | | **tunnel mode ipv6ip** | Mandatory<br>By default, the tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface name of the tunnel interface | | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the tunnel interface. |

| Step | Command | Description |
|---|---|---|
| Configure the destination address or host name of the tunnel interface | **tunnel destination** { *ip-address* \| *hostname* } | Mandatory<br>By default, do not configure the destination address or host name on the tunnel interface. |

**Note:**

- The source address and destination address must be configured at both ends of the tunnel, and the addresses of the two ends are mutually the source address and destination address.
- If adopting the interface mode when configuring the tunnel source, the source address of the tunnel is the master address of the source interface.
- The two ends of the tunnel should be configured as the same tunnel mode. Otherwise, transmitting via the tunnel fails.
- On one device, you cannot configure multiple tunnels whose tunnel mode, source address, and destination address are all the same.

## 11.2.2. **Configure IPv4 Compatible IPv6 Auto Tunnel**

### Configuration Conditions

Before configuring the IPv4 over IPv6 manual tunnel, complete the following tasks:

- Configure the IP address of the physical interface, making the neighboring nodes reachable at the network layer
- Create one tunnel interface, and configure the basic parameters (refer to the configuration manual of the tunnel interface)
- Configure any unicast routing protocol, making the route at the two ends of the tunnel reachable

### Configure IPv4 Compatible IPv6 Auto Tunnel

Table 11-4 Configure IPv4 compatible IPv6 auto tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode | | **configure terminal** | - |
| Enter the tunnel configuration mode | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address | Configure IPv6 global unicast address or anycast address or the local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig }** | Mandatory<br>By default, do not configure the IPv6 address on the tunnel interface. |
| | Configure the local address of the IPv6 link | **ipv6 address** *ipv6-address* **link-local** | Optional<br>By default, after enabling the IPv6 on the interface, automatically generate the local address of the link. |
| Configure the tunnel interface mode as the IPv4 compatible IPv6 auto tunnel | | **tunnel mode ipv6ip auto-tunnel** | Mandatory<br>By default, the mode of the tunnel interface is GRE over IPv4. |
| Configure the source address or interface name of the tunnel interface | | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address and interface name on the tunnel interface. |

**Note:**

- IPv4 compatible IPv6 auto tunnel does not need to be configured with the destination address. When encapsulating the tunnel, the used destination address is the embededed IPv4 address auto got from IPv4 compatible IPv6 address.
- If adopting the interface mode when configuring the tunnel source, the source address of the tunnel is the master address of the source interface.
- The two ends of the tunnel should be configured as the same tunnel mode. Otherwise, transmitting via the tunnel fails.
- On one device, you can only configure one IPv4 compatible IPv6 auto tunnel.

### 11.2.3. Configure the 6to4 Tunnel

#### Configuration Conditions

Before configuring the 6to4 tunnel, complete the following tasks:

- Configure the IP address of the physical interface, making the neighboring nodes reachable at the network layer
- Create one tunnel interface, and configure the basic parameters (refer to the configuration manual of the tunnel interface)
- Configure any unicast routing protocol, making the route at the two ends of the tunnel reachable

#### Configure the 6to4 Tunnel

Table 11-5 Configure the 6to4 tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode | | **configure terminal** | - |
| Enter the tunnel configuration mode | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address | Configure IPv6 global unicast address or anycast address or the local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig** } | Mandatory<br>By default, do not configure the IPv6 address on the tunnel interface. |

| Step | | Command | Description |
|---|---|---|---|
| Configure the tunnel interface address | Configure the local address of the IPv6 link | **ipv6 address** *ipv6-address* **link-local** | Optional<br>By default, after enabling IPv6 on the interface, automatically generate the local address of the link. |
| Configure the tunnel interface mode as 6to4 tunnel | | **tunnel mode ipv6ip 6to4** | Mandatory<br>By default, the tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface name of the tunnel interface | | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the tunnel interface. |

**Note:**

- It is not necessary to configure the destination address for the 6to4 tunnel. The destination address used during tunnel encapsulation is automatically got from the imbedded IPv4 address in the 6to4 tunnel IPv6 address.
- If adopting the interface mode when configuring the tunnel source, the source address of the tunnel is the master address of the source interface.
- The two ends of the tunnel should be configured as the same tunnel mode. Otherwise, transmitting via the tunnel fails.
- On one device, you can only configure one 6to4 tunnel.

## 11.2.4. **Configure the ISATAP Tunnel**

### Configuration Conditions

Before configuring the ISATAP tunnel, complete the following tasks:

- Configure the IP address of the physical interface, making the neighboring nodes reachable at the network layer
- Create one tunnel interface, and configure the basic parameters (refer to the configuration manual of the tunnel interface)
- Configure any unicast routing protocol, making the route at the two ends of the tunnel reachable

### Configure the ISATAP Tunnel

Table 11-6 Configure the ISATAP tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode | | **configure terminal** | - |
| Enter the tunnel configuration mode | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address | Configure IPv6 global unicast address or anycast address or the local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig** } | Mandatory<br>By default, do not configure the IPv6 address on the tunnel interface. |
| | Configure the local address of the IPv6 link | **ipv6 address** *ipv6-address* **link-local** | Optional<br>By default, after enabling IPv6 on the interface, auto generate the local address of the link. |
| Configure the tunnel interface mode as the IPv4 compatible IPv6 auto tunnel | | **tunnel mode ipv6ip isatap** | Mandatory<br>By default, the tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface name of the tunnel interface | | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the tunnel interface. |

**Note:**

- It is not necessary to configure the destination address for the 6to4 tunnel. The destination address used during tunnel encapsulation is automatically got from the imbedded IPv4 address in the 6to4 tunnel IPv6 address.
- If adopting the interface mode when configuring the tunnel source, the source address of the tunnel is the master address of the source interface.
- The two ends of the tunnel should be configured as the same tunnel mode. Otherwise, transmitting via the tunnel fails.
- On one device, you cannot configure multiple ISATAP tunnels with the same source address at the same time.

### 11.2.5. Monitoring and Maintaining of Transition Tunnel

Table 11-7 Monitoring and maintaining of the transition tunnel

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***]** | Displays the configuration information of all tunnels or a specified tunnel. |

## 11.3. Typical Configuration Examples of Transition Tunnel

### 11.3.1. Configure Basic Functions of IPv6 over IPv4 Manual Tunnel

#### Network Requirements

- IPv6 Network1 and IPv6 Network2 are the private IPv6 network of Device1 and Device3 respectively.
- IPv6 Network1 and IPv6 Network2 communicate via the IPv6 over IPv4 manual tunnel between Device1 and Device3.

#### Network Topology

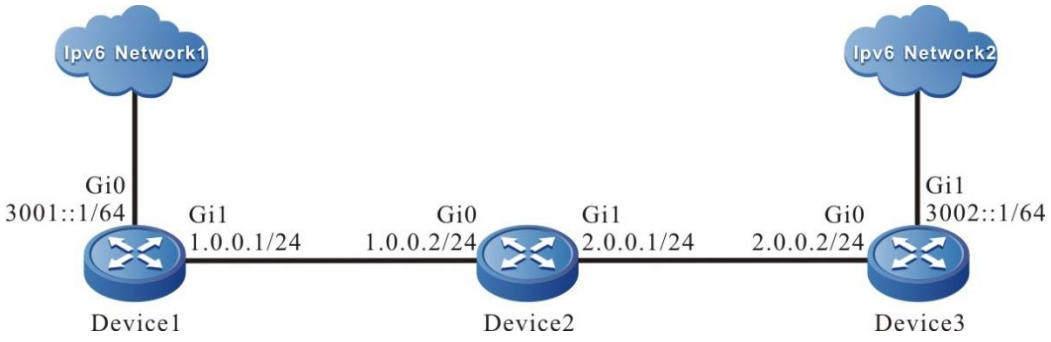



Figure 11-1 Networking for Configuring the basic functions of IPv6 over IPv4 manual tunnel

#### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure OSPF, making Device1, Device2, and Device3 communicate with each other.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#Query the route table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:18:40, gigabitethernet0
C   2.0.0.0/24 is directly connected, 00:22:27, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the querying process is omitted.

**Step 3:** Configure the IPv6 over IPv4 manual tunnel.

#On Device1, configure IPv6 over IPv4 manual tunnel tunnel1, the source address is 1.0.0.1, the destination address is 2.0.0.2, and the IPv6 address is 10::1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode ipv6ip
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#tunnel destination 2.0.0.2
Device1(config-if-tunnel1)#ipv6 address 10::1/64
```

```
Device1(config-if-tunnel1)#exit
```

#On Device3, configure IPv6 over IPv4 manual tunnel tunnel1, the source address is 2.0.0.2, the destination address is 1.0.0.1, and the IPv6 address is 10::2.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode ipv6ip
Device3(config-if-tunnel1)#tunnel source 2.0.0.2
Device3(config-if-tunnel1)#tunnel destination 1.0.0.1
Device3(config-if-tunnel1)#ipv6 address 10::2/64
Device3(config-if-tunnel1)#exit
```

#Query the IPv6 over IPv4 manual tunnel information of Device3.

```
Device3#show tunnel 1

Tunnel 1:
  Tunnel mode is ipv6ip
  Source ipv4 address is 2.0.0.2(Source ipv4 address is up on source interface gigabitethernet0)
  Destination ipv4 address is 1.0.0.1
  Tunnel state is up
  Encapsulation vrf is global(0x0)
  TTL(time-to-live) is 255
  TOS(type of service) is not set
total(1)
```

**Note:**

- The querying method of Device1is the same as that of Device3, so the querying process is omitted.
- When the tunnel is not in the same network segment, it is necessary to configure the static route to the peer tunnel on the devices at both ends of the tunnel, and the output interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to IPv6 Network2 with the egress interface tunnel1.

```
Device1(config)#ipv6 route 3002::/64 tunnel1
```

#On Device3, configure the static route to IPv6 Network1 with the egress interface tunnel1.

```
Device3(config)#ipv6 route 3001::/64 tunnel1
```

#Query the IPv6 route table of Device3.

```
Device3#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
```

```
O - OSPF, OE-OSPF External, M - Management

L   ::1/128 [0/0]
    via ::, 1w6d:20:35:50, lo0
C   10::/64 [0/0]
    via ::, 00:03:31, tunnel1
L   10::2/128 [0/0]
    via ::, 00:03:29, lo0
S   3001::/64 [1/100000]
    via ::, 00:00:01, tunnel1
C   3002::/64 [0/0]
    via ::, 00:00:06, gigabitethernet1
L   3002::1/128 [0/0]
    via ::, 00:00:04, lo0
```

**Note:**

- The querying method of Device1 is the same as that of Device3, so the querying process is omitted.

## 11.3.2. Configure Basic Functions of IPv4 Compatible IPv6 Auto Tunnel

### Network Requirements

- Device1 and Device2 have IPv4 and IPv6 dual protocol stack, and they communicate with each other through IPv4 network.
- An IPv4 compatible IPv6 automatic tunnel is established between Device1 and Device2, through which Device1 and Device2 communicate with each other.

### Network Topology


Figure 11-2 Networking of configuring the basic functions of IPv4 compatible IPv6 auto tunnel

### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the IPv4 compatible IPv6 auto tunnel.

#On Device1, configure the IPv4 compatible IPv6 auto tunnel tunnel1, and the source address is 1.0.0.1.

```
Device1#configure terminal
Device1(config)#interface tunnel 1
```

```
Device1(config-if-tunnel1)#tunnel mode ipv6ip auto-tunnel
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#exit
```

#On Device2, configure IPv4 compatible IPv6 auto tunnel (tunnel1), and the source address is 2.0.0.2.

```
Device2#configure terminal
Device2(config)#interface tunnel 1
Device2(config-if-tunnel1)#tunnel mode ipv6ip auto-tunnel
Device2(config-if-tunnel1)#tunnel source 2.0.0.2
Device2(config-if-tunnel1)#exit
```

#View the interface information of Device2 tunnel1.

```
Device2#show ipv6 interface tunnel1
tunnel1 is up
  VRF: global
  IPv6 is enable, link-local address is fe80::0201:7aff:fe5e:d029
  Global unicast address(es):
::0200:0002, subnet is ::/96
  Joined group address(es):
    ff02::0001:ff00:0002
    ff02::0001:ff00:0
    ff02::0002
    ff02::0001
    ff02::0001:ff5e:d029
  ND control flags: 0x1
  MTU is 1480 bytes
  ICMP redirects are enabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
```

You can see that the interface of Device2 tunnel1 automatically generates IPv6 address::0200:0002.

**Note:**

- IPv4 compatible IPv6 auto tunnel generates IPv4 compatible IPv6 address according to tunnel source address, and Qtech devices display it in hexadecimal format.

#View the IPv6 route table of Device2.

```
Device2#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management
C   ::/96 [0/0]
     via ::, 00:06:00, tunnel1
L   ::1/128 [0/0]
   via ::, 1w6d:20:35:50, lo0
L   ::200:2/128 [0/0]
     via ::, 00:05:59, lo0
```

#View the IPv4 compatible IPv6 auto tunnel information of Device2.

```
Device2#show tunnel 1

Tunnel 1:
   Tunnel mode is ipv6ip auto-tunnel
   Source ipv4 address is 2.0.0.2(Source ipv4 address is up on source interface gigabitethernet0)
   Tunnel state is up
   Encapsulation vrf is global(0x0)
   TTL(time-to-live) is 255
   TOS(type of service) is not set
total(1)
```

**Note:**

- The querying method of Device1 is the same as that of Device2, so the querying process is omitted.

**Step 3:** Check the result.

#On Device2, ping the IPv6 address of Device1 tunnel1 ::0100:0001.

```
Device2#ping ::0100:0001

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to ::100:1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/3/16 ms.
```

#Device2 can ping the IPv6 address of Device1 tunnel1 :0100:0001.

## 11.3.3. **Configure the Basic Functions of the 6to4 Tunnel**

### Network Requirements

- IPv6 Network1 and IPv6 Network2 are the private IPv6 network of Device1 and Device3 respectively.
- PC1 in IPv6 Network1 and PC2 in IPv6 Network2 communicate via the 6to4 tunnel between Device1 and Device3.

### Network Topology

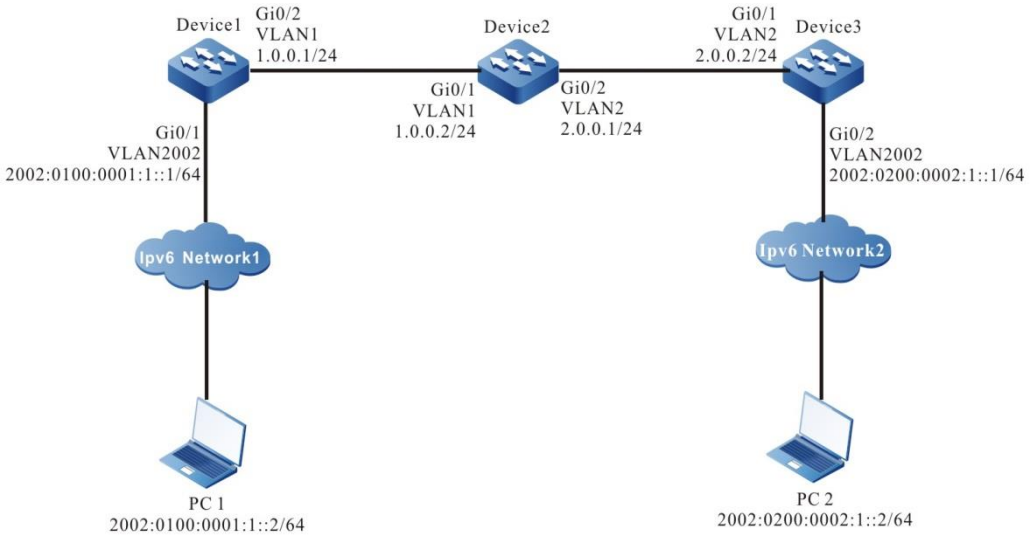



Figure 11-3 Networking for Configuring the basic functions of the 6to4 tunnel

### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure OSPF, making Device1, Device2, and Device3 communicate with each other.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
```

```
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#Query the route table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:18:40, gigabitethernet0
C   2.0.0.0/24 is directly connected, 00:22:27, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the querying process is omitted.

**Step 3:** Configure the 6to4 tunnel.

#On Device1, configure the 6to4 tunnel (tunnel1), the source address is 1.0.0.1, and the IPv6 address is 2002:100:1::1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode ipv6ip 6to4
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#ipv6 address 2002:100:1::1/64
Device1(config-if-tunnel1)#exit
```

#On Device3, configure the 6to4 tunnel (tunnel1), the source address is 2.0.0.2, and the IPv6 address is 2002:200:2::1.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode ipv6ip 6to4
Device3(config-if-tunnel1)#tunnel source 2.0.0.2
Device3(config-if-tunnel1)#ipv6 address 2002:200:2::1/64
Device3(config-if-tunnel1)#exit
```

#Query the 6to4 tunnel information of Device3.

```
Device3#show iptl kernel
IP tunnel kernel information:
IP Tunnel Interface 1 (0x10001b3):
  Tunnel mode is ipv6ip 6to4
  Tunnel state is up
  Destination address is Unknown family
```

```
    Source address is 2.0.0.2
    Source interface is vlan2 (0x4000299)
    Time To Live (TTL) is 255
    Type Of Service (TOS) is not set
    VRF is global (0x0)
    Source interface VRF is global (0x0)
    Internal flags is 0x30000

  total(1)
```

**Note:**

- The querying method of Device1is the same as that of Device3, so the querying process is omitted.
- When the tunnel is not in the same network segment, it is necessary to configure the static routing to the peer tunnel on the devices at both ends of the tunnel, and the output interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to the segment 2002::/16 with the egress interface (tunnel1).

```
Device1(config)#ipv6 route 2002::/16 tunnel1
```

#On Device3, configure the static route to the segment 2002::/16 with the egress interface (tunnel1).

```
Device3(config)#ipv6 route 2002::/16 tunnel1
```

#Query the IPv6 route table of Device3.

```
Device3#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
    U - Per-user Static route
    O - OSPF, OE-OSPF External, M - Management


L   ::1/128 [0/0]
    via ::, 6w1d:02:11:13, lo0
S   2002::/16 [1/100000]
    via ::, 00:10:31, tunnel1
C   2002:200:2::/64 [0/0]
    via ::, 00:12:51, tunnel1
L   2002:200:2::1/128 [0/0]
    via ::, 00:12:49, lo0
C   2002:200:2:1::/64 [0/0]
```

```
    via ::, 00:12:15, gigabitethernet1
L   2002:200:2:1::1/128 [0/0]
    via ::, 00:12:13, lo0
```

**Note:**

- The querying method of Device1is the same as that of Device3, so the querying process is omitted.

**Step 5:** Check the result.

#On PC1, ping the PC2 address 2002:200:2:1::2.

```
C:\>ping6 2002:200:2:1::2 –s 2002:0100:0001:1::2

Pinging 2002:200:2:1::2 with 32 bytes of data:

Reply from 2002:200:2:1::2: time<1ms
Reply from 2002:200:2:1::2: time<1ms
Reply from 2002:200:2:1::2: time<1ms
Reply from 2002:200:2:1::2: time<1ms

Ping statistics for 2002:200:2:1::2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#PC1 can ping PC2 address 2002:200:2:1::2.

## 11.3.4. **Configure 6to4 Tunnel Relay**

### Network Requirements

- Device1 is 6to4 relay device.
- IPv6 Network1 and IPv6 Network2 are the private IPv6 network of Device1 and Device3 respectively.
- PC1 in IPv6 Network1 and PC2 in IPv6 Network2 communicate via the 6to4 tunnel between Device1 and Device3.

## Network Topology

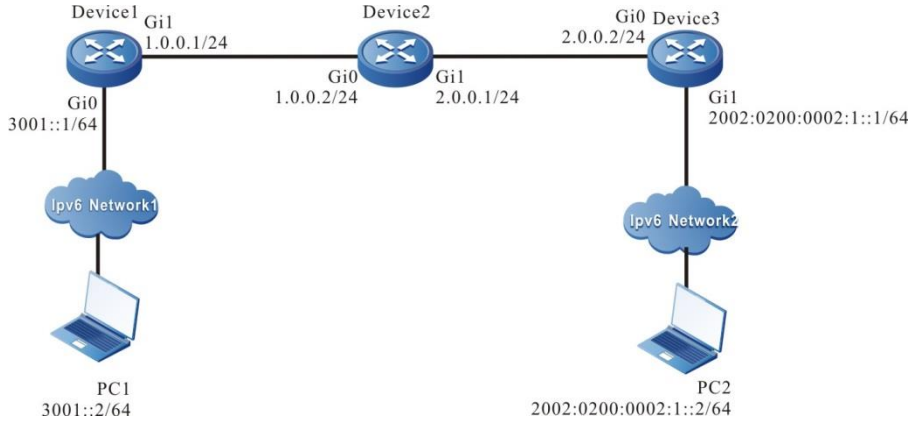



Figure 11-4 Networking of configuring the 6to4 tunnel relay

## Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure OSPF, making Device1, Device2, and Device3 communicate with each other.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
       D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS
```

```
Gateway of last resort is not set

O   1.0.0.0/24 [110/65536] via 2.0.0.1, 00:18:40, gigabitethernet0
C   2.0.0.0/24 is directly connected, 00:22:27, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the querying process is omitted.

**Step 3:** Configure the 6to4 tunnel.

#On Device1, configure the 6to4 tunnel (tunnel1), the source address is 1.0.0.1, and IPv6 address is 2002:100:1::1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode ipv6ip 6to4
Device1(config-if-tunnel1)#tunnel source 1.0.0.1
Device1(config-if-tunnel1)#ipv6 address 2002:100:1::1/64
Device1(config-if-tunnel1)#exit
```

#On Device3, configure the 6to4 tunnel (tunnel1), the source address is 2.0.0.2, and IPv6 address is 2002:200:2::1.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode ipv6ip 6to4
Device3(config-if-tunnel1)#tunnel source 2.0.0.2
Device3(config-if-tunnel1)#ipv6 address 2002:200:2::1/64
Device3(config-if-tunnel1)#exit
```

#View the 6to4 tunnel information of Device3.

```
Device3#show tunnel 1

Tunnel 1:
  Tunnel mode is ipv6ip 6to4
  Source ipv4 address is 2.0.0.2(Source ipv4 address is up on source interface gigabitethernet0)
  Tunnel state is up
  Encapsulation vrf is global(0x0)
  TTL(time-to-live) is 255
  TOS(type of service) is not set
total(1)
```

**Note:**

- The viewing method of Device1 is the same as that of Device3, and the viewing process is omitted.
- When the tunnel is not in the same network segment, the static route to the peer end tunnel needs to be configured on the devices at both ends of the tunnel, and the egress interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to the egress interface tunnel1 of the segment 2002::/16.

```
Device1(config)#IPv6 route 2002::/16 tunnel1
```

#On Device3, configure the static route to the egress interface tunnel1 of the segment 2002::/16, and configure the static route to the gateway 2002:100:1::1 of the segment 3001::/64.

```
Device3(config)#IPv6 route 2002::/16 tunnel1
Device3(config)#IPv6 route 3001::/64 2002:100:1::1
```

#View the Ipv6 route table of Device3.

```
Device3#show IPv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management
L   ::1/128 [0/0]
    via ::, 6w1d:02:11:13, lo0
S   2002::/16 [1/100000]
    via ::, 00:10:31, tunnel1
C   2002:200:2::/64 [0/0]
    via ::, 00:12:51, tunnel1
L   2002:200:2::1/128 [0/0]
    via ::, 00:12:49, lo0
C   2002:200:2:1::/64 [0/0]
    via ::, 00:12:15, gigabitethernet1
L   2002:200:2:1::1/128 [0/0]
    via ::, 00:12:13, lo0
S   3001::/64 [1/100000]
    via 2002:100:1::1, 00:00:53, tunnel1
```

**Note:**

- The querying method of Device1 is the same as that of Device3, so the querying process is omitted.

**Step 4:** Configure the static route.

#On PC2, ping the address of PC1 3001::2.

```
C:\>ping6 3001::2 –s 2002:0200:0002:1::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
```

#PC2 can ping the address 3001::2 of PC1.

## 11.3.5. **Configure Basic Functions of ISATAP Tunnel**

### Network Requirements

- Device and PC2 communicate with each other via the IPv4 network.
- PC2 is the ISATAP host, setting up the ISATAP tunnel with Device.
- PC1 in IPv6 Network and PC2 in IP Network perform the IPv6 communication via the ISATAP tunnel.

### Network Topology

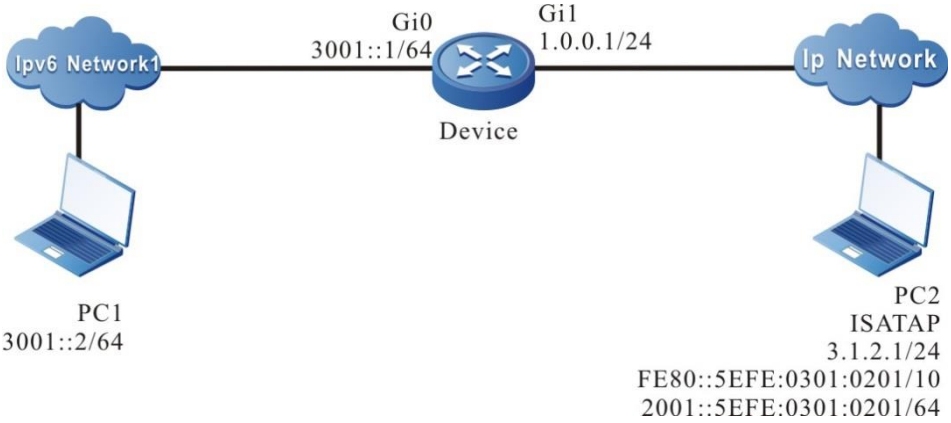



Figure 11-5 Networking for Configuring the basic functions of the ISATAP tunnel

### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** On Device, configure the ISATAP tunnel.

#On Device, configure the ISATAP tunnel (tunnel1), the source address is 1.0.0.1, and the IPv6 address is 2001::5efe:100:1.

```
Device#configure terminal
```

```
Device(config)#interface tunnel 1
Device(config-if-tunnel1)#tunnel mode ipv6ip isatap
Device(config-if-tunnel1)#tunnel source 1.0.0.1
Device(config-if-tunnel1)#ipv6 address 2001::5efe:100:1/64
Device(config-if-tunnel1)#exit
```

#Query the IPv6 route table of Device.

```
Device#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management



L   ::1/128 [0/0]
    via ::, 6d:05:16:46, lo0
C   2001::/64 [0/0]
    via ::, 00:07:56, tunnel1
L   2001::5efe:100:1/128 [0/0]
    via ::, 00:07:54, lo0
C   3001::/64 [0/0]
    via ::, 02:42:37, gigabitethernet0
L   3001::1/128 [0/0]
    via ::, 02:42:36, lo0
```

#Query the ISATAP tunnel information of Device.

```
Device#show iptl kernel
IP tunnel kernel information:
IP Tunnel Interface 1 (0x1000212):
   Tunnel mode is ipv6ip isatap
   Tunnel state is up
   Destination address is Unknown family
   Source address is 1.0.0.1
   Source interface is vlan1 (0x4000213)
   Time To Live (TTL) is 255
   Type Of Service (TOS) is not set
   VRF is global (0x0)
   Source interface VRF is global (0x0)
   Internal flags is 0x30000
```

```
total(1)
```

**Step 3:** On Device, disable the RA response suppression function of Tunnel1.

#On Device, disable the RA response suppression function of Tunnel1.

```
.Device(config)#interface tunnel 1
Device(config-if-tunnel1)#no ipv6 nd suppress-ra response
Device(config-if-tunnel1)#exit
```

**Step 4:** Configure the ISATAP host PC2.

#On the host, the configuration of the ISATAP tunnel varies with the operation system. This text takes the Windows XP operation system as an example to describe. On PC2, install the IPv6 protocol.

```
C:\>ipv6 install
```

#Usually, after Windows XP is installed with the IPv6 protocol successfully, IPv6 interface 2 is the ISATAP interface. On PC2, query the information of IPv6 interface 2.

```
C:\>ipv6 if 2

Interface 2: Automatic Tunneling Pseudo-Interface

 Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}

 does not use Neighbor Discovery

 does not use Router Discovery

 routing preference 1

 EUI-64 embedded IPv4 address: 0.0.0.0

 router link-layer address: 0.0.0.0

  preferred link-local fe80::5efe:3.1.2.1, life infinite

 link MTU 1280 (true link MTU 65515)

 current hop limit 128

 reachable time 42500ms (base 30000ms)
```

retransmission interval 1000ms

DAD transmits 0

default site prefix length 48

The IPv6 interface 2 of PC2 automatically generates the ISATAP format link-local address fe80::5efe:3.1.2.1, and do not get the address prefix.

#On the IPv6 interface 2 of PC2, configure the destination address of the ISATAP tunnel as 1.0.0.1.

C:\>ipv6 rlu 2 1.0.0.1

#On PC2, query the information of IPv6 interface 2.

C:\>ipv6 if 2

Interface 2: Automatic Tunneling Pseudo-Interface

Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}

does not use Neighbor Discovery

uses Router Discovery

routing preference 1

EUI-64 embedded IPv4 address: 3.1.2.1

router link-layer address: 1.0.0.1

preferred global 2001::5efe:3.1.2.1, life 29d23h59m46s/6d23h59m46s (public)

preferred link-local fe80::5efe:3.1.2.1, life infinite

link MTU 1500 (true link MTU 65515)

current hop limit 255

reachable time 42500ms (base 30000ms)

retransmission interval 1000ms

```
DAD transmits 0

default site prefix length 48
```

The IPv6 interface 2 of PC2 gets the address prefix 2001::/64, and generates the global unicast address 2001::5efe:3.1.2.1.

**Step 5:** Check the result.

#On PC1, ping the address of PC2 IPv6 interface 2 2001::5efe:3.1.2.1.

```
C:\>ping 2001::5efe:3.1.2.1

Pinging 2001::5efe:3.1.2.1 with 32 bytes of data:

Reply from 2001::5efe:3.1.2.1: time<1ms
Reply from 2001::5efe:3.1.2.1: time<1ms
Reply from 2001::5efe:3.1.2.1 :time<1ms
Reply from 2001::5efe:3.1.2.1: time<1ms

Ping statistics for 2001::5efe:3.1.2.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#PC1 can ping the address of PC2 IPv6 interface2 2001::5efe:3.1.2.1.

# 12. IPV6 TUNNEL

## 12.1. Overview

IPv6 tunnel (Generic Packet Tunneling in IPv6) is one of tunnel technologies. The starting and ending points of the tunnel need to be manually configured. It is a virtual point-to-point connection. It provides a transmission channel for encapsulated packets. The two ends of the tunnel encapsulate and de-capsulate the packet respectively. The IPv6 tunnel can encapsulate IPv4 and IPv6 packets, so that these encapsulated packets can be transmitted in another IPv6 network.

- IPv6 tunnel encapsulation

When the IPv4 or IPv6 packets are sent through the IPv6 tunnel, an IPv6 packet header is added to the header, the Next Header field in the IPv6 packet header is set to 4 or 41, the source address in the IPv6 packet header is set to the source address of the tunnel, and the destination address in the IPv6 packet header is set as the destination address of the tunnel.

- The structure of the IPv6 tunnel packet

| IPv6 Header | Original Header | Original Packet Payload |
|---|---|---|

Original Packet Payload: The payload of the packet before entering the tunnel, which serves as the valid payload of the tunnel packet.

Original Header: The header of the packet before entering the tunnel, such as IPv4 or IPv6 header

IPv6 Header: The encapsulated outer IPv6 packet header, which is the transmission tool of the original packet across the IPv6 network.

- The forwarding of the IPv6 tunnel packet

After the packet is encapsulated at the beginning of the IPv6 tunnel, select the route according to the encapsulated destination address, and then, send the packet from the corresponding network interface. The intermediate device forwards it as an ordinary IPv6 packet until the packet reaches the end of the tunnel.

- The encapsulation/de-capsulation of the IPv6 tunnel packet

The de-capsulation process and the encapsulation process are opposite. The tunnel end first analyzes the IPv6 header after receiving the packet. If the destination address is its own address, check the Next Header field of the IPv6 header. If the Next Header field is 4 or 41, hand over the packet to the IPv6 tunnel for processing. After the tunnel removes the IPv6 header of the packet, select the route according to the packet type and destination address of the packet after de-capsulation, and perform the subsequent processing according to the result of the route selection.

## 12.2. IPv6 Tunnel Function Configuration

Table 12-1 IPv6 tunnel function configuration list

| Configuration task | |
|---|---|
| Configure the IPv6 tunnel | Configure the IPv6 tunnel |

### 12.2.1. Configure an IPv6 Tunnel

#### Configuration Conditions

Before configuring the IPv6 tunnel, complete the following tasks:

- Configure the IPv6 address of the physical interface, making the neighboring nodes reachable at the network layer.
- Create one tunnel interface, and configure the basic parameters (refer to the configuration manual of the tunnel interface).
- Configure any unicast routing protocol, making the route at the two ends of the tunnel reachable.

#### Configure an IPv6 Tunnel

Table 12-2 Configure an IPv6 tunnel

| Step | | Command | Description |
|---|---|---|---|
| Enter the global configuration mode | | **configure terminal** | - |
| Enter the tunnel configuration mode | | **interface tunnel** *tunnel-number* | - |
| Configure the tunnel interface address | Configure the IPv4 unicast address | **ip address** *ip-address* { *mask* \| *mask-length* } | Either<br>By default, do not configure the address on the tunnel interface. |
| | Configure IPv6 global unicast address or anycast address or the local address of the site or auto get address | **ipv6 address {** *ipv6-address/prefix-length* [ **anycast** \| **eui-64** ] \| **autoconfig** } | |

| Step | | Command | Description |
|---|---|---|---|
| | Configure the local address of the IPv6 link | **ipv6 address** *ipv6-address* **link-local** | Optional<br>By default, after the interface enables IPv6, automatically generate the local address of the link. |
| Configure the tunnel interface mode as the IPv6 tunnel | | **tunnel mode ipv6** | Mandatory<br>By default, the tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface of the tunnel interface | | **tunnel source** { *ipv6-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the tunnel interface. |
| Configure the destination address or host name of the tunnel interface | | **tunnel destination** { *ipv6-address* \| *hostname* } | Mandatory<br>By default, do not configure the destination address or host name on the tunnel interface. |

**Note:**

- The source address and destination address must be configured at both ends of the tunnel, and the addresses of the two ends are mutually the source address and destination address.
- If adopting the interface mode when configuring the tunnel source, the source address of the tunnel is the IPv6 address of the source interface.
- The two ends of the tunnel should be configured as the same tunnel mode. Otherwise, transmitting via the tunnel fails.
- On one device, you cannot configure multiple tunnels whose tunnel mode, source address, and destination address are all the same.

### 12.2.2. IPv6 Tunnel Monitoring and Maintaining

Table 12-3 IPv6 tunnel monitoring and maintaining

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***]** | Display the configuration information of all tunnels or the specified tunnel |

## 12.3. Typical Configuration Example of IPv6 Tunnel

### 12.3.1. Configure Basic Functions of IPv6 Tunnel

#### Network Requirements

- IP Network1 and IP Network2 are the private IP network of Device1 and Device3 respectively.
- IPv6 Network1 and IPv6 Network2 communicate via the IPv6 over IPv4 manual tunnel between Device1 and Device3.
- IP Network1 and IP Network2 communicate via the IPv6 tunnel between Device1 and Device3.
- IPv6 Network1 and IPv6 Network2 communicate via the IPv6 tunnel between Device1 and Device3.

#### Network Topology

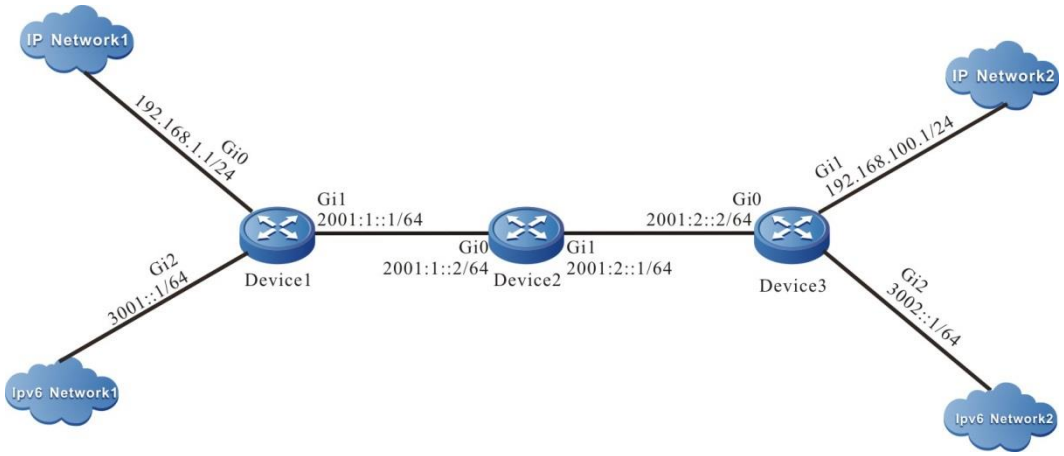



Figure 12-1 Networking for Configuring the basic functions of the IPv6 tunnel

#### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure OSPFv3, making Device, Device2, and Device3 communicate with each other.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#ipv6 router ospf 100
Device1(config-ospf6)#router-id 1.1.75.1
```

```
Device1(config-ospf6)#exit
Device1(config)#interface gigabitethernet1
Device1(config-if-gigabitethernet1)#ipv6 router ospf tag 100 area 0
Device1(config-if-gigabitethernet1)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#ipv6 router ospf 100
Device2(config-ospf6)#router-id 1.2.75.1
Device2(config-ospf6)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#ipv6 router ospf tag 100 area 0
Device2(config-if-gigabitethernet0)#exit
Device2(config)#interface gigabitethernet1
Device2(config-if-gigabitethernet1)#ipv6 router ospf tag 100 area 0
Device2(config-if-gigabitethernet1)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#ipv6 router ospf 100
Device3(config-ospf6)#router-id 1.1.73.1
Device3(config-ospf6)#exit
Device3(config)#interface gigabitethernet0
Device3(config-if-gigabitethernet0)#ipv6 router ospf tag 100 area 0
Device3(config-if-gigabitethernet0)#exit
```

#Query the IPv6 route table of Device3.

```
Device3#show ipv6 route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management


L   ::1/128 [0/0]
    via ::, 6w0d:23:09:31, lo0
O   2001:1::/64 [110/2]
    via fe80::508b:fff:fee4:ff6, 00:08:37, gigabitethernet0
C   2001:2::/64 [0/0]
    via ::, 00:15:51, gigabitethernet0
L   2001:2::2/128 [0/0]
    via ::, 00:15:50, lo0
```

```
C   3002::/64 [0/0]
    via ::, 00:15:06, gigabitethernet2
L   3002::1/128 [0/0]
    via ::, 00:15:04, lo0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the querying process is omitted.

**Step 3:** Configure the IPv6 tunnel.

#On Device1, configure the IPv6 tunnel (tunnel1), the source address is 2001:1::1, destination address is 2001:2::2, IP address is 10.0.0.1, and IPv6 address is 10::1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode ipv6
Device1(config-if-tunnel1)#tunnel source 2001:1::1
Device1(config-if-tunnel1)#tunnel destination 2001:2::2
Device1(config-if-tunnel1)#ip address 10.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#ipv6 address 10::1/64
Device1(config-if-tunnel1)#exit
```

#On Device3, configure the IPv6 tunnel (tunnel1), the source address is 2001:2::2, destination address is 2001:1::1, IP address is 10.0.0.2, and IPv6 address is 10::2.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode ipv6
Device3(config-if-tunnel1)#tunnel source 2001:2::2
Device3(config-if-tunnel1)#tunnel destination 2001:1::1
Device3(config-if-tunnel1)#ip address 10.0.0.2 255.255.255.0
Device3(config-if-tunnel1)#ipv6 address 10::2/64
Device3(config-if-tunnel1)#exit
```

#Query the IPv6 tunnel information of Device3.

```
Device3#show tunnel 1

Tunnel 1:
  Tunnel mode is ipv6
    Gre checksum validation is disabled
    Gre key is not set
  Source ipv6 address is 2001:2::2 (Source ipv6 address is up on source interface gigabitethernet0)
  Destination ipv6 address is 2001:1::1
  Tunnel state is up
```

Encapsulation vrf is global(0x0)

TTL(time-to-live) is 255

TOS(type of service) is not set

total(1)

**Note:**

- The querying method of Device1 is the same as that of Device3, so the querying process is omitted.
- When the tunnel is not in the same network segment, it is necessary to configure the static route to the peer tunnel on the devices at both ends of the tunnel, and the output interface is the tunnel interface.

**Step 4:** Configure the static route.

#On Device1, configure the static route to IP Network2 with the egress interface tunnel1.

Device1(config)#ip route 192.168.100.0 255.255.255.0 tunnel1

#On Device1, configure the static route to IPv6 Network2 with the egress interface tunnel1.

Device1(config)#ipv6 route 3002::/64 tunnel1

#On Device3, configure the static route to IP Network1 with the egress interface tunnel1.

Device3(config)#ip route 192.168.1.0 255.255.255.0 tunnel1

#On Device3, configure the static route to IPv6 Network1 with the egress interface tunnel1.

Device3(config)# ipv6 route 3001::/64 tunnel1

#Query the route table of Device3.

Device3#show ip route

Codes: C - connected, S - static, R - RIP, O - OSPF, OE-OSPF External, M - Management

D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C 10.0.0.0/24 is directly connected, 00:17:12, tunnel1

S 192.168.1.0/24 [1/100000] is directly connected, 00:00:10, tunnel1

C 192.168.100.0/24 is directly connected, 00:00:28, gigabitethernet1

#Query the IPv6 route table of Device3.

Device3#show ipv6 route

Codes: C - Connected, L - Local, S - static, R - RIP, B - BGP, i-ISIS

U - Per-user Static route

O - OSPF, OE-OSPF External, M - Management

```
L   ::1/128 [0/0]
    via ::, 6w0d:23:50:28, lo0
C   10::/64 [0/0]
    via ::, 00:12:23, tunnel1
L   10::2/128 [0/0]
    via ::, 00:12:22, lo0
O   2001:1::/64 [110/2]
    via fe80::508b:fff:fee4:ff6, 00:49:34, gigabitethernet0
C   2001:2::/64 [0/0]
    via ::, 00:56:48, gigabitethernet0
L   2001:2::2/128 [0/0]
    via ::, 00:56:46, lo0
S   3001::/64 [1/100000]
    via ::, 00:00:14, tunnel1
C   3002::/64 [0/0]
    via ::, 00:56:02, gigabitethernet2
L   3002::1/128 [0/0]
    via ::, 00:56:01, lo0
```

**Note:**

- The querying method of Device1 is the same as that of Device3, so the querying process is omitted.

# 13. DVPN TUNNEL

## 13.1. Overview of DVPN Tunnel

DVPN is the abbreviation of Dymatic Vitual Private Network (DVPN). It is a non-public technology to solve the complex configuration of traditional tunnel technology, which is inconvenient to respond to the changes of network topology and branch address. DVPN uses the idea of centralized control, and a central node completes the online, authentication and public-private network address management of branch nodes. With the help of the tunnel established between nodes, the private network routing information is transmitted, and then the whole network interworking is realized.

In the network of DVPN tunnel, devices have the following roles, among which Hub and Spoke are called Client in the network:

- Hub——The gateway device of the enterprise headquarters
- Spoke——The gateway device of the enterprise branch
- Server——The server is responsible for receiving, managing and maintaining the information of DVPN nodes and responding to queries from clients (only in UDP tunnels).

The DVPN tunnels are divided to the MGRE tunnel and UDP tunnel.

### MGRE tunnel

MGRE (Multipoint Generic Routing Encapsulation) is a dynamic VPN Technology based on NHRP (NextHop resolution protocol). MGRE is a point-to-multipoint GRE tunnel technology. Packet encapsulation and decapsulation are the same as GRE tunnel. Please refer to the relevant chapter of GRE tunnel in configuration manual for detailed introduction. Unlike GRE tunnel, only the tunnel source address needs to be specified, but the destination address of tunnel is not required. MGRE tunnel relies on the NHRP protocol, which can dynamically learn the addresses of other nodes. Hub maintains the mapping table entries of the tunnel address and public network address. Spoke will register its own address information with Hub, and Hub and multiple spooks will establish permanent tunnel. When Spoke needs to communicate with other spooks, Spoke will obtain the addresses of other Spokes from Hub, so as to establish the tunnel between Spoke and Spoke.

### UDP tunnel

UDP tunnel is a tunnel technology which uses UDP protocol as encapsulation protocol and transmits other transport protocols in the network through IP Transport Protocol. The UDP tunnel is a point to multipoint tunnel technology. It is the dynamic VPN technology to solve the problem of complex configuration, inconvenient response to network topology changes and branch address changes. At the same time, the UDP tunnel technology provides identity authentication, control packet encryption protection and data packet encryption protection, and realizes the safe transmission of VPN internal data on the public network. The network environment of UDP tunnel adopts the client/server model. All the clinet node information (hub and spoke) is registered on the server, and the client's own information is saved on the server. Therefore, the server is responsible for collecting, managing and maintaining the client's information, and the client can obtain other client's information through the server. Thus, an independent tunnel is established between clients for communication. In the network, the spoke and hub will register with the server, and the server will establish an information table of all clients for management and maintenance. Once a client successfully registers with the server, the server will send all hub information to the spoke, and the spoke will send a tunnel establishment request to the hub to establish a permanent tunnel, Then, the spoke will obtain

the address of the spoke to be communicated from the server, so as to establish a tunnel between the spoke and the spoke.

## 13.2. DVPN Tunnel Function Configuration

Table 13-1 DVPN tunnel function configuration list

| Configuration Task | |
|---|---|
| Configure the MGRE tunnel | Configure the MGRE tunnel |
| Configure the UDP tunnel | Configure the UDP tunnel |

### 13.2.1. Configure MGRE Tunnel

#### Configuration Conditions

Before configuring the MGRE tunnel, it is necessary to complete the following tasks:

- Configure the IP address of the interface, so that the network layer of the adjacent nodes can reach.
- Create a tunnel interface and configure basic parameters (refer to tunnel interface configuration manual).
- Configure any unicast routing protocol or static route to make both ends of the tunnel reachable.

#### Configure MGRE Tunnel

Table 13-2 Configure the MGRE tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the Tunnel interface configuration mode | **interface tunnel** *tunnel-number* | - |
| Configure the Tunnel interface address | **ip address** *ip-address* { *mask* \| *mask-length* } | Mandatory<br>By default, the address is not configured on the Tunnel interface. |

| Step | Command | Description |
|---|---|---|
| Configure the Tunnel interface mode as MGRE | **tunnel mode dvpn mgre** | Mandatory<br>By default, the Tunnel interface mode is GRE over IPv4. |
| Configure the source address or interface name of the Tunnel interface | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the Tunnel interface. |
| Configure the checksum function of the Tunnel interface | **tunnel checksum** | Optional<br>By default, do not configure the checksum function on the Tunnel interface.<br>It is not required to configure the checksum function at both sides of the tunnel at the same time. |
| Configure the keyword of the Tunnel interface | **tunnel key** *key-number* | Optional<br>By default, do not configure keywords on the tunnel interface.<br>The keywords at both ends of the tunnel must be set consistently. Otherwise, the transmission through the tunnel will fail. |
| Configure the IPSEC protection function of the Tunnel interface | **crypto-profile** *profile-name* | Optional<br>By default, do not configure the IPSEC protection function on the Tunnel interface. |

| Step | Command | Description |
|---|---|---|
| Configure the pre-share key of the Tunnel interface | **dvpn authentication key** {0 *string* \| 7 *cipher-string*} | Optional<br>By default, no preshared key is configured on the tunnel interface<br>The pre shared key of the spoke and the hub must be set the same. Otherwise, the spoke cannot be registered successfully |
| Configure the aging time of the MGRE entry | **dvpn nhrp-entry holdtime** *seconds* | Optional<br>By default, do not configure the aging time of the MGRE entry on the Tunnel interface. |
| Configure the interval of sending the registration request packet | **dvpn registration interval** *seconds* | Optional<br>By default, do not configure the interval of sending the registration request packet on the Tunnel interface. |
| Configure the interval of re-transmitting the client request packet | **dvpn retry interval** *time-interval* | Optional<br>By default, do not configure the interval of re-transmitting the client request packet on the Tunnel interface. |
| Configure the slient time of the client connection timeout | **dvpn dumb** *time-interval* | Optional<br>By default, do not configure the slient time of the client connection timeout on the Tunnel interface. |

| Step | Command | Description |
|---|---|---|
| Configure the Hub-Entry address mapping entry | **dvpn nhrp-entry** *protocol-address* { *public-address* } [**register**] | Mandatory<br>By default, do not configure the Hub-Entry address mapping entry on the Tunnel interface. |
| Turn on the switch of printing the security log information of hub or spoke registration failure, save the log and view it through show logging security-data | **tunnel logging security-data dvpn-mgre { register-fail \| register-ok }** | Optional<br>By default, do not enable the printing command. |

**Note:**

- The source address must be configured at both ends of the tunnel, and there is no need to configure the destination address of the tunnel.
- When configuring the tunnel source, if the interface mode is adopted, the source address of the tunnel is the primary address of the source interface.
- At most two mGRE tunnels can be configured on the same device, and the tunnel sources cannot be the same.
- At most two Hub-Entries (with register in the command line of the configuration table item) are configured on the spoke side.
- Up to 2048 spokes are supported. If the number of spokes on Hub exceeds the upper limit, the registration of spokes will fail.

### 13.2.2. **Configure UDP Tunnel**

#### Configuration Conditions

Before configuring the UDP tunnel, it is necessary to complete the following tasks:

- Configure the IP address of the interface, so that the network layer of the adjacent nodes can reach.
- Create a tunnel interface and configure basic parameters (refer to Tunnel interface configuration manual).
- Configure any unicast routing protocol or static route to make both ends of the tunnel reachable.

### Configure UDP Tunnel

Table 13-3 Configure the UDP tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the Tunnel interface configuration mode | **interface tunnel** *tunnel-number* | - |
| Configure the Tunnel interface address | **ip address** *ip-address* { *mask* \| *mask-length* } | Mandatory<br>By default, do not configure the address on the Tunnel interface. |
| Configure the Tunnel interface mode as UDP | **tunnel mode dvpn udp** | Mandatory<br>By default, the mode of the Tunnel interface is GRE over IPv4. |
| Configure the source address or interface name of the Tunnel interface | **tunnel source** { *ip-address* \| *interface-name* } | Mandatory<br>By default, do not configure the source address or interface name on the Tunnel interface. |
| Configure the client identity of the Tunnel interface | **dvpn client** { **Hub** \| **Spoke** } | Mandatory<br>By default, do not configure the client identity on the Tunnel interface. |
| Configure the IP address of the DVPN server on the Tunnel interface | **dvpn** { **primary** \| **secondary**} **server** {*public-address* } | Mandatory<br>By default, do not configure the IP address of the DVPN server on the Tunnel interface. |

| Step | Command | Description |
|---|---|---|
| Configure the idle timeout of the Spoke-Spoke DVPN tunnel on the Tunnel interface | **dvpn Spoke-Spoke idle** *time-interval* | Optional<br>By default, do not configure the idle timeout of the Spoke-Spoke DVPN tunnel on the Tunnel interface. |
| Configur the IPSEC protection function of the Tunnel interface | **crypto-profile** *profile-name* | Optional<br>By default, do not configure the IPSEC protection function on the Tunnel interface. |
| Configure the pre-share key of the Tunnel interface | **dvpn authentication key** {0 *string* \| 7 *cipher-string*} | Mandatory<br>By default, do not configure the pre-share key on the Tunnel interface.<br>The preshare key of the client and Server should be consistent. Otherwise, the Cclient cannot be registered successfully. |
| Configure the interval of re-transmitting the client request packet | **dvpn retry interval** *time-interval* | Optional<br>By default, do not configure the interval of re-transmitting the client request packet on the Tunnel interface. |
| Configure the silent time of the client connection timeout | **dvpn dumb** *time-interval* | Optional<br>By default, do not configure the silent time of the client connection timeout on the Tunnel interface. |

Table 13-4 Configure DVPN Server

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the DVPN server | **dvpn server enable** | Enter the DvpnServer view |
| Configure the authentication key word of the DVPN server | **authentication-key** { **0** *string* \| **7** *cipher-string* } | Mandatory<br>By default, the DVPN server is not configured with the authentication key word.<br>The key words of the Client and Server should be consistent. Otherwise, client cannot be registered successfully. |
| Configure the encryption algorithm of the DVPN server | **encryption-algorithm** {**3des-cbc** \| **aes-cbc-128** \| **aes-cbc-192** \| **aes-cbc-256** \| **des-cbc** \| **SM1** \| **SM4** } | Optional<br>By default, all configurable encryption algorithms are supported on the primary and standby servers, and the priority order is DES-CBC, 3DES-CBC, AES-CBC-128, AES-CBC-192, AES-CBC-256, SM1, and SM4. |
| Configure the authentication algorithm of the DVPN server | **authentication-algorithm** { **SM3** \| **MD5** \| **sha-1** \| **sha-256** \| **HMAC-MD5** \| **HMAC-SHA1** } | Mandatory<br>By default, all configurable authentication algorithms are supported on the DVPN primary and standby servers. The priority order is MD5, SHA-1, SHA-256, SM3, HMAC-MD5, and HMAC-SHA1. |

| Step | Command | Description |
|---|---|---|
| Configure the interval of sending the keepalive packet between Server and Client | **dvpn keepalive interval** *seconds* | Mandatory<br>By default, do not configure the source address or interface name on the Tunnel interface. |

**Note:**

- The source address must be configured at both ends of the tunnel, and there is no need to configure the destination address of the tunnel.
- When configuring the tunnel source, if the interface mode is adopted, the source address of the tunnel is the primary address of the source interface.
- At most two UDP tunnels can be configured on the same device.
- The pre-share keys should be configured on the primary and standby servers, and the pre shared keys configured on the client and server must be the same. Otherwise, the registration cannot be successful, and the configuration algorithms supported on the primary and standby servers are the same.
- Each client can only be configured with two servers (one primary and one standby). When the primary server is offline, the client automatically switches to the standby server for communication. If the primary server is not configured, the standby server cannot be configured.
- Up to 64 spokes are supported and controlled by DVPN server. If the number of Spokes on the server exceeds the upper limit, the registration of spokes will fail. Support up to 2 servers, primary server and standby server.

### 13.2.3. **DVPN Tunnel Monitoring and Maintaining**

Table 13-5 MGRE tunnel monitoring and maintaining

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***] [slot** *slot-num***]** | Display the tunnel information |
| **show dvpn mgre entry { tunnel** *tunnel-id* **}[slot** *slot-num***]** | Display all static and dynamic address mapping entries in the specified tunnel |
| **clear { tunnel** *tunnel-id* **} dvpn mgre entry [***tunnel-addr* **\| all]** | Clear the peer entry of the tunnel interface |

Table 13-6 UDP tunnel monitoring and maintaining

| Command | Description |
|---|---|
| **show tunnel [***tunnel-id***] [slot** *slot-num***]** | Display the tunnel information |
| **show dvpn udp client fsm** | Display the status machine information of all DVPN tunnels in the UDP mode |
| **show dvpn udp entry { tunnel** *tunnel-id* **}[slot** *slot-num***]** | Display all IPv4 DVPN tunnel information of the specified tunnel in the UDP mode |
| **show dvpn udp server address-map** | Display the IPv4 private network address mapping information of the DVPN client registered to the DVPN server |
| **show dvpn udp server fsm**[*protocol-address*] | Display the client information of the specified private network address |

**Note:**

- *slot-num* indicates displaying the tunnel status information on the specified board card.

## 13.3. DVPN Tunnel Typical Configuration Example

### 13.3.1. Configure MGRE Basic Functions

#### Network Requirements

- The IP networks connected with Device1, Device2 and Device3 are the corresponding private IP networks.
- IP network1 and IP network2 communicate through the DVPN MGRE tunnel between Device1 and Device2.
- IP network1 and IP network3 communicate through the DVPN MGRE tunnel between Device1 and Device3.
- IP network2 and IP network3 communicate through the DVPN MGRE tunnel between Device2 and Device3.

## Network Topology

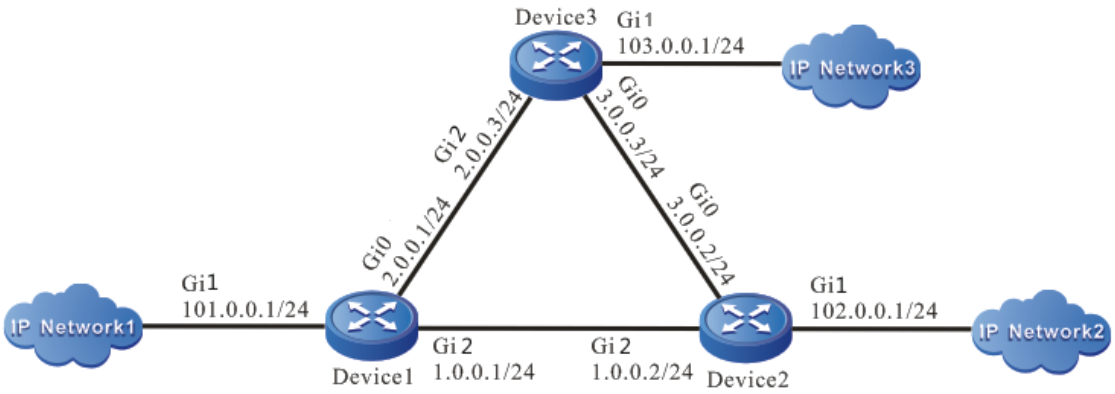



Figure 13-1Networking of configuring DVPN MGRE basic functions

## Configuration Steps

**Step 1:** Configure the IP address of the physical interface. (omitted)

#Configure Device1. Configure loopback port as the source address of the tunnel.

```
Device1#configure terminal
Device1(config)#interface loopback0
Device1(config-if-loopback0)#ip address 192.168.1.1 255.255.255.255
Device1(config-if-loopback0)#exit
```

#Configure Device2. Configure the loopback port as the source address of the tunnel.

```
Device2#configure terminal
Device2(config)#interface loopback0
Device2(config-if-loopback0)#ip address 192.168.1.2 255.255.255.255
Device2(config-if-loopback0)#exit
```

#Configure Device3. Configure the loopback port as the source address of the tunnel.

```
Device3#configure terminal
Device3(config)#interface loopback0
Device3(config-if-loopback0)#ip address 192.168.1.3 255.255.255.255
Device3(config-if-loopback0)#exit
```

**Step 2:** Configure OSPF, making the route between Device1, Device2 and Device3 reachable.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device1(config-ospf)# network 192.168.1.1 0.0.0.0 area 0
```

```
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 3.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 192.168.1.2 0.0.0.0 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#network 3.0.0.0 0.0.0.255 area 0
Device3(config-ospf)# network 192.168.1.3 0.0.0.0 area 0
Device3(config-ospf)#exit
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

O   1.0.0.0/24 [110/2] via 2.0.0.1, 05:21:40, gigabitethernet2
                [110/2] via 3.0.0.2, 05:28:42, gigabitethernet0
C   2.0.0.0/24 is directly connected, 06:07:27, gigabitethernet2
L   2.0.0.3/32 is directly connected, 06:07:27, gigabitethernet2
C   3.0.0.0/24 is directly connected, 18:52:48, gigabitethernet0
L   3.0.0.3/32 is directly connected, 18:52:48, gigabitethernet0
C   103.0.0.0/24 is directly connected, 18:52:48, gigabitethernet1
L   103.0.0.3/32 is directly connected, 18:52:48, gigabitethernet1
LC   192.168.1.3/32 is directly connected, 05:13:45, loopback0
O   192.168.1.1/32 [110/2] via 2.0.0.1, 05:28:42, gigabitethernet2
O   192.168.1.2/32 [110/2] via 3.0.0.2, 05:21:40, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the querying process is omitted.

**Step 3:** Configure the MGRE mode of the DVPN tunnel.

#On Device1, configure the DVPN tunnel (tunnel1) as the MGRE mode, source address is loopback0, tunnel address is 188.0.0.1, and register to Device3.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode dvpn mgre
Device1(config-if-tunnel1)#tunnel source loopback0
Device1(config-if-tunnel1)#dvpn nhrp-entry 188.0.0.3 192.168.1.3 register
Device1(config-if-tunnel1)#ip address 188.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#exit
```

#On Device2, configure the DVPN tunnel (tunnel1) as the MGRE mode, source address is loopback0, tunnel address is 188.0.0.2, and register to Device3.

```
Device2(config)#interface tunnel 1
Device2(config-if-tunnel1)#tunnel mode dvpn mgre
Device2(config-if-tunnel1)#tunnel source loopback0
Device2(config-if-tunnel1)#dvpn nhrp-entry 188.0.0.3 192.168.1.3 register
Device2(config-if-tunnel1)#ip address 188.0.0.2 255.255.255.0
Device2(config-if-tunnel1)#exit
```

#On Device3, configure the DVPN tunnel (tunnel1) as the MGRE mode, source address is loopback0, and tunnel address is 188.0.0.3.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode dvpn mgre
Device3(config-if-tunnel1)#tunnel source loopback0
Device3(config-if-tunnel1)#ip address 188.0.0.3 255.255.255.0
Device3(config-if-tunnel1)#exit
```

**Note:**

- Device1 and Device2 acts as spoke to initiate registration to Device3 as hub.
- A DVPN MGRE tunnel is established between devices. The public network and private network of the DVPN tunnel need to be routed through OSPF.

**Step 4:** Configure OSPF, making the public network and private network routes of the DVPN tunn reachable.

#Configure Device1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#ip ospf network point-to-multipoint
Device1(config-if-tunnel1)#exit
Device1(config)#router ospf 101
Device1(config-ospf)#network 101.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2(config)#interface tunnel 1
Device2(config-if-tunnel1)#ip ospf network point-to-multipoint
Device2(config-if-tunnel1)#exit
Device2(config)#router ospf 101
Device2(config-ospf)#network 102.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#ip ospf network point-to-multipoint
        Device3(config-if-tunnel1)#exit
Device3(config)#router ospf 101
Device3(config-ospf)#network 103.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
    U - Per-user Static route
    O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

O   1.0.0.0/24 [110/2] via 2.0.0.1, 05:21:40, gigabitethernet2
          [110/2] via 3.0.0.2, 05:28:42, gigabitethernet0
C   2.0.0.0/24 is directly connected, 06:07:27, gigabitethernet2
L   2.0.0.3/32 is directly connected, 06:07:27, gigabitethernet2
C   3.0.0.0/24 is directly connected, 18:52:48, gigabitethernet0
L   3.0.0.3/32 is directly connected, 18:52:48, gigabitethernet0
O   101.0.0.0/24 [110/1001] via 188.0.0.1, 07:04:55, tunnel1
O   102.0.0.0/24 [110/1001] via 188.0.0.2, 07:04:55, tunnel1
C   188.0.0.0/24 is directly connected, 07:01:21, tunnel1
O   188.0.0.1/32 [110/1000] via 188.0.0.1, 07:00:13, tunnel1
L   188.0.0.3/32 is directly connected, 07:01:21, tunnel1
O   188.0.0.2/32 [110/1000] via 188.0.0.2, 06:57:33, tunnel1
C   103.0.0.0/24 is directly connected, 18:52:48, gigabitethernet1
L   103.0.0.3/32 is directly connected, 18:52:48, gigabitethernet1
```

```
LC  192.168.1.3/32 is directly connected, 05:13:45, loopback0
O   192.168.1.1/32 [110/2] via 2.0.0.1, 05:28:42, gigabitethernet2
O   192.168.1.2/32 [110/2] via 3.0.0.2, 05:21:40, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the process is omitted.

**Step 5:** Check the result.

#On Device1, view the static mapping entry.

```
Device1#show dvpn mgre entry tunnel 1
Private address     Public address     Type        Created time      Expire time
188.0.0.3           192.168.1.3        static      06:30:39          --
```

#On Device2, view the static mapping entry.

```
Device2#show dvpn mgre entry tunnel 1
Private address     Public address     Type        Created time      Expire time
188.0.0.3           192.168.1.3        static      06:29:28          --
```

#On Device3, view the static mapping entry.

```
Device3#show dvpn mgre entry tunnel 1
Private address     Public address     Type        Created time      Expire time
188.0.0.1           192.168.1.1        hub-dynamic 06:30:50          01:59:11
188.0.0.2           192.168.1.2        hub-dynamic 06:33:24          01:56:37
```

#The packet forwarding between IP networks is normal, and the packet is protected by mGRE mode of DVPN tunnel.

## 13.3.2. Configure UDP Basic Functions

### Network Requirements

- The IP networks connected with Device1, Device2 and Device3 are the corresponding private IP networks.
- IP network1 and IP network2 communicate through the DVPN UDP tunnel between Device1 and Device2.
- IP network1 and IP network3 communicate through the DVPN UDP tunnel between Device1 and Device3.
- IP network2 and IP network3 communicate through the DVPN UDP tunnel between Device2 and Device3.

## Network Topology

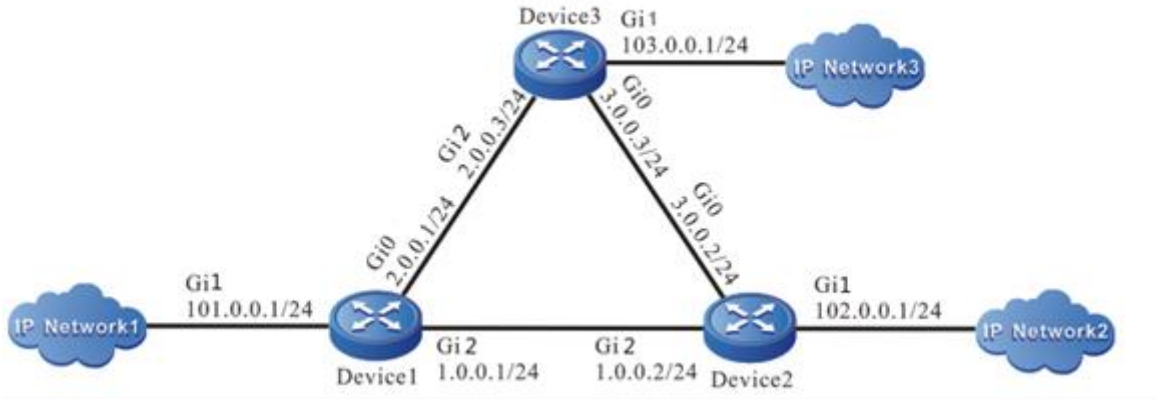



Figure 13-2 Networking of configuring DVPN UDP basic functions

## Configuration Steps

**Step 1:** Configure the IP address of the physical interface. (omitted)

#Configure Device1. Configure loopback port as the source address of the tunnel.

```
Device1#configure terminal
Device1(config)#interface loopback0
Device1(config-if-loopback0)#ip address 192.168.1.1 255.255.255.255
Device1(config-if-loopback0)#exit
```

#Configure Device2. Configure loopback port as the source address of the tunnel.

```
Device2#configure terminal
Device2(config)# interface loopback0
Device2(config-if-loopback0)#ip address 192.168.1.2 255.255.255.255
Device2(config-if-loopback0)#exit
```

#Configure Device3. Configure loopback port as the source address of the tunnel.

```
Device3#configure terminal
Device3(config)#interface loopback0
Device3(config-if-loopback0)#ip address 192.168.1.3 255.255.255.255
Device3(config-if-loopback0)#exit
```

**Step 2:** Configure OSPF, making the route between Device1, Device2, and Device3 reachable.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#router ospf 100
Device1(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#network 192.168.1.1 0.0.0.0 area 0
```

```
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#router ospf 100
Device2(config-ospf)#network 1.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 3.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 192.168.1.2 0.0.0.0 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#router ospf 100
Device3(config-ospf)#network 2.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#network 3.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#network 192.168.1.3 0.0.0.0 area 0
Device3(config-ospf)#exit
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

O   1.0.0.0/24 [110/2] via 2.0.0.1, 05:21:40, gigabitethernet2
                [110/2] via 3.0.0.2, 05:28:42, gigabitethernet0
C   2.0.0.0/24 is directly connected, 06:07:27, gigabitethernet2
L   2.0.0.3/32 is directly connected, 06:07:27, gigabitethernet2
C   3.0.0.0/24 is directly connected, 18:52:48, gigabitethernet0
L   3.0.0.3/32 is directly connected, 18:52:48, gigabitethernet0
C   103.0.0.0/24 is directly connected, 18:52:48, gigabitethernet1
L   103.0.0.3/32 is directly connected, 18:52:48, gigabitethernet1
LC   192.168.1.3/32 is directly connected, 05:13:45, loopback0
O   192.168.1.1/32 [110/2] via 2.0.0.1, 05:28:42, gigabitethernet2
O   192.168.1.2/32 [110/2] via 3.0.0.2, 05:21:40, gigabitethernet0
```

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the process is omitted.

**Step 3:** On Device3, configure UDP Server.

```
Device3(config)#dvpn server enable
Device3(config-dvpnserver)#authentication-key 0 a
Device3(config-dvpnserver)#exit
```

**Step 4:** Configure the UDP mode of the DVPN tunnel.

#On Device1, configure the DVPN tunnel (tunnel1) as the UDP mode, source address is loopback0, tunnel address is 188.0.0.1, and Device1 acts as spoke to register to Server.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#tunnel mode dvpn udp
Device1(config-if-tunnel1)#tunnel source loopback0
Device1(config-if-tunnel1)#dvpn primary server 192.168.1.3
Device1(config-if-tunnel1)#ip address 188.0.0.1 255.255.255.0
Device1(config-if-tunnel1)#dvpn authentication key 0 a
Device1(config-if-tunnel1)#dvpn client spoke
Device1(config-if-tunnel1)#exit
```

#On Device2, configure the DVPN tunnel (tunnel1) as the UDP mode, source address is loopback0, tunnel address is 188.0.0.2, and Device2 acts as spoke to register to Server.

```
Device2(config)#interface tunnel 1
Device2(config-if-tunnel1)#tunnel mode dvpn udp
Device2(config-if-tunnel1)#tunnel source loopback0
Device2(config-if-tunnel1)#dvpn primary server 192.168.1.3
Device2(config-if-tunnel1)#ip address 188.0.0.2 255.255.255.0
Device2(config-if-tunnel1)#dvpn authentication key 0 a
Device2(config-if-tunnel1)#dvpn client spoke
Device2(config-if-tunnel1)#exit
```

#On Device3, configure the DVPN tunnel (tunnel1) as the UDP mode, source address is loopback0, tunnel address is 188.0.0.3, and Device3 acts as hub to register to Server.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#tunnel mode dvpn udp
Device3(config-if-tunnel1)#tunnel source loopback0
Device3(config-if-tunnel1)#dvpn primary server 192.168.1.3
Device3(config-if-tunnel1)#ip address 188.0.0.3 255.255.255.0
Device3(config-if-tunnel1)#dvpn authentication key 0 a
Device3(config-if-tunnel1)#dvpn client hub
Device3(config-if-tunnel1)#exit
```

**Note:**

- Device3 acts as Server, Device1 and Device2 act as spoke to initiate registration to Server, and Device3 initiates registration to itself as hub.

- The DVPN UDP tunnel between devices is established, and the public network and private network of the DVPN tunnel need to be routed through OSPF.

**Step 5:** Configure OSPF, making the public network and private network routes of the DVPN tunnel of all devices reachable.

#Configure Device1.

```
Device1(config)#interface tunnel 1
Device1(config-if-tunnel1)#ip ospf network point-to-multipoint
Device1(config-if-tunnel1)#exit
Device1(config)#router ospf 101
Device1(config-ospf)#network 101.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device1(config-ospf)#exit
```

#Configure Device2.

```
Device2(config)#interface tunnel 1
Device2(config-if-tunnel1)#ip ospf network point-to-multipoint
Device2(config-if-tunnel1)#exit
Device2(config)#router ospf 102
Device2(config-ospf)#network 102.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device2(config-ospf)#exit
```

#Configure Device3.

```
Device3(config)#interface tunnel 1
Device3(config-if-tunnel1)#ip ospf network point-to-multipoint
Device3(config-if-tunnel1)#exit
Device3(config)#router ospf 103
Device3(config-ospf)#network 103.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#network 188.0.0.0 0.0.0.255 area 0
Device3(config-ospf)#exit
```

#View the route table of Device3.

```
Device3#show ip route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

O   1.0.0.0/24 [110/2] via 2.0.0.1, 05:21:40, gigabitethernet2
              [110/2] via 3.0.0.2, 05:28:42, gigabitethernet0
```

C 2.0.0.0/24 is directly connected, 06:07:27, gigabitethernet2
L 2.0.0.3/32 is directly connected, 06:07:27, gigabitethernet2
C 3.0.0.0/24 is directly connected, 18:52:48, gigabitethernet0
L 3.0.0.3/32 is directly connected, 18:52:48, gigabitethernet0
O 101.0.0.0/24 [110/1001] via 188.0.0.1, 07:04:55, tunnel1
O 102.0.0.0/24 [110/1001] via 188.0.0.2, 07:04:55, tunnel1
C 188.0.0.0/24 is directly connected, 07:01:21, tunnel1
O 188.0.0.1/32 [110/1000] via 188.0.0.1, 07:00:13, tunnel1
L 188.0.0.3/32 is directly connected, 07:01:21, tunnel1
O 188.0.0.2/32 [110/1000] via 188.0.0.2, 06:57:33, tunnel1
C 103.0.0.0/24 is directly connected, 18:52:48, gigabitethernet1
L 103.0.0.3/32 is directly connected, 18:52:48, gigabitethernet1
LC 192.168.1.3/32 is directly connected, 05:13:45, loopback0
O 192.168.1.1/32 [110/2] via 2.0.0.1, 05:28:42, gigabitethernet2
O 192.168.1.2/32 [110/2] via 3.0.0.2, 05:21:40, gigabitethernet0

**Note:**

- The querying methods of Device1 and Device2 are the same as that of Device3, so the process is omitted.

**Step6:** Check the result.

#On Device3, view the server status, hub registration status and tunnel setup.

```
Device3#show dvpn udp server fsm
Server status        : Enabled
Registered spoke number: 2
Registered hub number  : 1

Client type              : Spoke
Current state             : Online
Hold time                : 00:34:24
Client port              : 60001
Client area id            : 0
Client request id          : 3
Client private address       : 188.0.0.1
Client public  address       : 192.168.1.1
Client DH group           : 8192 bits DH group
Encryption-algorithm        : AES-CBC-256
Authentication-algorithm     : SHA-1
```

Data encryption-algorithm : AES-CBC-256
Data authentication-algorithm : SHA-1

Client type : Hub
Current state : Online
Hold time : 00:35:38
Client port : 60001
Client area id : 0
Client request id : 3
Client private address : 188.0.0.3
Client public address : 192.168.1.3
Client DH group : 8192 bits DH group
Encryption-algorithm : AES-CBC-256
Authentication-algorithm : SHA-1
Data encryption-algorithm : AES-CBC-256
Data authentication-algorithm : SHA-1

Client type : Spoke
Current state : Online
Hold time : 00:34:51
Client port : 60001
Client area id : 0
Client request id : 3
Client private address : 188.0.0.2
Client public address : 192.168.1.2
Client DH group : 6144 bits DH group
Encryption-algorithm : AES-CBC-256
Authentication-algorithm : SHA-1
Data encryption-algorithm : AES-CBC-256
Data authentication-algorithm : SHA-1

Device3#show dvpn udp client fsm
Primary server : 192.168.1.3
Interface tunnel : 1
Current state : Online
Hold time : 00:37:14
Key update expire time : 23:22:46

```
Keepalive interval time      : 300 seconds
DH group value               : 8192 bits DH group
Encryption-algorithm         : AES-CBC-256
Authentication-algorithm     : SHA-1
Data encryption-algorithm    : AES-CBC-256
Data authentication-algorithm : SHA-1

Device3#show dvpn udp entry tunnel 1
Private address  Public address  Port  Type  State    Expire time  Hold time
188.0.0.1        192.168.1.1     60001 H-S   Success  --           00:37:10
188.0.0.2        192.168.1.2     60001 H-S   Success  --           00:36:43
```

#On Device1, view the registration status and tunnel setup information.

```
Device1#show dvpn udp client fsm
Primary server               : 192.168.1.3
Interface tunnel             : 1
Current state                : Online
Hold time                    : 00:09:13
Key update expire time       : 23:50:47
Keepalive interval time      : 300 seconds
DH group value               : 6144 bits DH group
Encryption-algorithm         : AES-CBC-256
Authentication-algorithm     : SHA-1
Data encryption-algorithm    : AES-CBC-256
Data authentication-algorithm : SHA-1

Device1#show dvpn udp entry tunnel 1
Private address  Public address  Port  Type  State    Expire time  Hold time
188.0.0.3        192.168.1.3     60001 S-H   Success  --           00:09:17
```

#The packets between IP Networks are forwarded normally, and the packets are protected by the UDP mode of the DVPN tunnel.

**Note:**

- The querying method of Device1 is the same as that of Device2, so the process is omitted.

# 14. DHCP

## 14.1. Overview

It is hard to manage a large network. For example, in a network in which IP addresses are manually allocated, IP address conflicts are common. The only way of solving the problem is to dynamically allocate IP addresses to the hosts. The Dynamic Host Configuration Protocol (DHCP) allocates IP address to requesting hosts from an IP address pool. DHCP also provides other information, such as gateway IP and DNS server address. DHCP reduces the workload of the administrator in recording and tracking manually allocated IP addresses.

DHCP is a protocol that is based on UDP broadcast. The process for a DHCP client to obtain an IP address and other configuration information contains four phases.

DISCOVER phase. When the DHCP client accesses the network for the first time, it sends a DHCP DISCOVER packet with the source address 0.0.0.0 and destination address 255.255.255.255 to the network.

OFFER phase. After the DHCP server receives the DHCP DISCOVER broadcast packet sent by the client, it selects an IP address from the corresponding IP address pool according to the policy, and sends the IP address and other parameters to the client in a DHCP OFFER packet.

REQUEST phase. If the DHCP client receives response messages from multiple DHCP servers on the network, it selects one DHCP OFFER (usually the one that arrives first). Then it sends a DHCP REQUEST packet to the network, telling the entire DHCP server the IP address of which server it will accept.

ACK phase. After the DHCP server receives the DHCP REQUEST packet from the DHCP server, it sends a DHCP ACK message containing the provided IP address and other configuration to the DHCP client, telling the DHCP client that the DHCP client can use the provided IP address.

The IP address that the DHCP server allocates to the DHCP client has a lease. After the lease expires, the server will take back the allocated IP address. When the lease term of the IP address of the DHCP client has passed half time, the DHCP client sends a DHCP REQUEST packet to the DHCP server requesting to update its IP address lease. If the DHCP server allows the DHCP client to use its IP address, the DHCP server responds with a DHCP ACK packet, requesting the DHCP client to update the lease. If the DHCP server does not allow the DHCP client to continue to use the IP address, the DHCP server responds with a DHCP NAK packet.

During dynamic IP address acquisition, request packets are sent in broadcast mode; therefore, DHCP is applied only when the DHCP client and server are in the same subnet. If multiple subnets exist in a network and the hosts of the subnets need to obtain configuration information such as IP address through the DHCP server, the hosts of the subnets communicate with the DHCP server through a DHCP relay to obtain IP addresses and other configuration information.

## 14.2. DHCP Function Configuration

Table 14-1 DHCP function list

| Configuration Tasks | |
|---|---|
| Configure a DHCP address pool. | Create a DHCP address pool. |
| | Bind an IP address to a DHCP host. |
| | Configure an IP address range. |
| | Configure a domain suffix. |
| | Configure the address of the NETBIOS server. |
| | Configure a DNS server address. |
| | Configure the default route. |
| | Configure the lease of an IP address. |
| | Configure VRF properties. |
| | Configure user-defined options. |
| | Configure the address pool of the specified manufacturer |
| Configure other parameters of a DHCP server. | Configure the DHCP server |
| | Configure the range of reserved IP addresses. |
| | Configure DHCP ping detection parameters. |
| | Configure DHCP data log function |

| Configuration Tasks | |
|---|---|
| Configure the functions of a DHCP client. | Configure a DHCP client. |
| | Configure the manufacturer ID |
| | Configure the DHCP routing distance |
| | Configure the Option60 |
| | Configure the DHCP client not to request the default route |
| Configure the DHCP relay function | Configure the DHCP relay |
| | Configure the Option 82 function |
| | Configure the source address of the interface DHCP relay packet |
| | Configure the address of the DHCP server |

### 14.2.1. Configure a DHCP Address Pool

#### Configuration Condition

None

#### Create a DHCP Address Pool

A DHCP server needs to select and allocate IP addresses and other parameters from a DHCP address pool. Therefore, a DHCP address pool must be created for the DHCP server first.

Table 14-2 Creating a DHCP address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |

| Step | Command | Description |
|---|---|---|
| Create a DHCP address pool and enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | Mandatory.<br>By default, no DHCP address pool has been created by the system. |

**Note:**

- Address pools fall into three types: Network, and Range, which can be configured respectively through the network, and range commands.

### Configure an IP Address Range

On a DHCP server, each DHCP address pool must be configured with an IP address range to allocate IP addresses to DHCP clients.

Table 14-3 Configuring an IP address range

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure the IP address range of the Network type. | **network** *ip-address* [*network-mask* **|** *mask-len*] | Optional.<br>By default, an IP address range is not configured for an address pool. |
| Configure an IP address range of the Range type. | **range** *low-ip-addres high-ip-address* [*network-mask* | *mask-len* ] | Optional.<br>By default, an IP address range is not configured for an address pool. |

**Note:**

- After an IP address range is configured for an address pool by using the **network** or **range** command, if you run the **network** or **range** command again, the new IP address range configuration overwrites the existing configuration.
- Modify the network type of the address pool to the range type (or change the range type of address pool to the network type). If there is an intersection between the newly configured address range and the old configured address range, the command line will prompt the user whether to perform the operation. If so, all address configurations

(static binding, vendor sub pool) and dynamic leases under the address pool will be deleted; If the actual effective range of the newly configured address includes that of the old configuration, the address pool will retain all the address configurations (static binding, vendor sub pool) under the address pool, but will delete the IP range and dynamic lease configured by the vendor sub pool.

### Configure a DNS Server Address

On a DHCP server, you can configure the DNS server address respectively for each DHCP address pool. When a DHCP server allocates an IP address for a DHCP client, it also sends the DNS server address to the client.

When the DHCP client starts dynamic domain name resolution, it queries the DNS server.

Table 14-4 Configuring a DNS server address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure a DNS server address. | **dns-server** { *ip-address*&<1-8> \| **autoconfig** } | Mandatory.<br>By default, the DNS server address is not configured. |

### Configure the Default Route

On a DHCP server, you can specify the address of a gateway corresponding to clients for each DHCP address pool. When the server allocates an IP address to a client, it also sends the gateway address to the client.

When a DHCP client accesses a server or host that is not in the network segment, its data is forwarded through the gateway.

Table 14-5 Configuring the default route

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure the default route. | **default-router** *ip-address*&<1-8> | Mandatory.<br>By default, no default route is configured. |

### Configure the Lease of an IP Address

The IP address that the DHCP server allocates to the DHCP client has a lease. After the lease expires, the server will take back the allocated IP address. If the DHCP client wants to continue to use the IP address, it must have the IP address lease updated.

On the DHCP server, you can configure an IP address lease for each DHCP address pool.

Table 14-6 Configuring the lease of an IP address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure an IP address lease. | **lease** *days* [ *hours* [ *minutes* ] ] | Mandatory.<br>By default, the value of *days* is 1, the value of *hours* is 6, and the value of *minutes* is 0. |

### Configure IP and MAC Address Binding

The command is used to configure IP and MAC address binding. When the client with specified MAC address requests the DHCP server to assign IP address, the DHCP server will assign its bound IP address. As long as the MAC address of the client remains unchanged (such as changing network card), the IP address obtained by the client from the server is the same every time.

Table 14-7 Configure IP and MAC address binding

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure IP and MAC address binding | **bind** { *ip-address mac-address* \| **automatic**} | Mandatory<br>By default, do not configure the IP, MAC address binding. |

### Configure User-Defined Options

For some options, RFC does not give specifications; therefore, you can define these options according to the actual requirement.

Table 14-8 Configuring user-defined options

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configuring user-defined options. | **option** *option-code* { **ascii** *ascii-string* \| **hex** *hex-string* \| **ip** *ip-address*&<1-8> } | Mandatory.<br>By default, user-defined options are not configured. |

### Configure the Address Pool of Specified Manufacturer

When the client requests for the IP address, it may carry the option60, indicating the manufacturer ID. The customer can specify different IP address segments for different manufacturers.

Table 14-9 Configure the address pool of the DHCP manufacturer

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCP configuration mode. | **ip dhcp pool** *pool-name* **[ vrf** *vrf-name* **]** | - |
| Configure the address pool of the manufacturer and enter the DHCP manufacturer address pool configuration mode | **vendor-class-identifier** *vendor_id* | By default, do not configure the manufacturer address pool. |
| Configure the manufacturer address pool range | **ip range** *low-ip-address high-ip-address* | By default, do not configure the range. |
| Configure the option 43 content returned for the specified manufacturer | **option 43** { **ascii** *ascii-string* \| **hex** *hex-string* \| **ip** *ip-address*&<1-8> } | By default, do not configure. |

## 14.2.2. Configure Other Parameters of a DHCP Server

### Configuration Condition

None

### Configure DHCP Server

After the interface works in the DHCP server mode, when the interface receives the DHCP request packet from the DHCP client, the DHCP server will assign the IP address and other network parameters to the client.

Table 14-10 Configure the DHCP server

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |

| Step | Command | Description |
|---|---|---|
| Configure the DHCP server function | **ip dhcp server** | Mandatory<br>By default, do not configure the DHCP server function. |

### Configure the Range of Reserved IP Addresses

In a DHCP address pool, some IP addresses are reserved for some special devices, and some IP addresses conflict with the IP addresses of other hosts in the network. Therefore, the IP addresses cannot be dynamically allocated.

Table 14-11 Configuring the range of reserved IP addresses

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the range of reserved IP addresses. | **ip dhcp excluded-address** *low-ip-address* [ *high-ip-address* ] [ **vrf** *vrf-name* ] | Mandatory.<br>By default, the range of reserved IP addresses is not configured.<br>The IP addresses in the reserved IP address range will not be allocated. |

### Configure DHCP Ping Detection Parameters

To prevent an IP address conflict, before dynamically allocating an IP address to a DHCP client, a DHCP server must detect the IP address. The detection operation is performed through the ping operation. The DHCP server determines whether an IP address conflict exists by checking whether an ICMP response packet is received within the specified time.

Table 14-12 Configuring DHCP ping detection parameters

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure DHCP ping detection parameters. | **ip dhcp ping** { **packets** *packet-num* \| **timeout** *milliseconds* } | Mandatory.<br>By default, the number of ping packets is 1, and the timeout time is 500ms. |

### Configure DHCP Data Log Function

After the data log function of the DHCP server is enabled, the address pool allocation of the DHCP server will be recorded in the data log.

Table 14-13 Configure the DHCP data log function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the data log function switch of the DHCP server | **ip dhcp logging security-data** | Mandatory’<br>By defaut, do not enable the data log function. |

## 14.2.3. Configure the Functions of a DHCP Client

### Configuration Condition

None

### Configure a DHCP Client

A DHCP client interface obtains an IP address and other parameters through DHCP.

Table 14-14 Configuring a DHCP Client

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |

| Step | Command | Description |
|---|---|---|
| Configure the DHCP client to obtain an IP address. | **ip address dhcp** [ **request-ip-address** *ip-address* \| *network-mask*] | Mandatory.<br>By default, the DHCP client is not configured to obtain an IP address. |

### Configure DHCP Routing Distance

In the IP routing table, each protocol has a management distance to control the routing, that is, routing distance. The routing distance is used to decide the routing for the same network segment of different protocols. The routing with a short distance has a high priority.

Table 14-15 Configure the DHCP routing distance

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the DHCP routing distance | **ip dhcp route-distance** *distance* | Mandatory<br>By default, the DHCP routing distance is 254. |

### Configure Option 60

The content of DHCP Option 60 is the manufacturer ID. When the DHCP client requests, it can carry option 60. The server can make the policy for distributing the IP address according to the option.

Table 14-16 Configure the DHCP Option 60 function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Configure the option 60 function | **ip dhcp vendor-class-identifier {disable \| content** *hex-string}* | By default, the DHCP client carries the manufacturer ID, and uses the default manufacturer ID carried by the system. |

### Configure the DHCP Client Not to Request Default Route

When the DHCP client requests the IP address, request the default route by default. The user can specify the DHCP client not to request the default route, but configure the route by itself.

Table 14-17 Configure the DHCP client not to request the default route

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the DHCP client not to request the default route | **ip dhcp router-option disable** | Mandatory<br>By default, the DHCP client requests the default route. |

## 14.2.4. Configure the Function of a DHCP Relay

### Configuration Condition

None

### Configure a DHCP Relay

If multiple subnets exist in a network and the hosts of the subnets need to obtain configuration information such as IP address through the DHCP server, the hosts of the subnets communicate with the DHCP server through a DHCP relay to obtain IP addresses and other configuration information. If the interface works in the DHCP relay mode, when the interface receives the DHCP packet from the DHCP client, it will relay the packet to the configured DHCP server, and the DHCP server will assign the IP address.

Table 14-18 Configuring a DHCP relay

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode. | **interface** *interface-name* | - |
| Configure the DHCP relay function. | **ip dhcp relay** | Mandatory.<br>By default, the DHCP relay function is not configured. |

### Configure the Option82 Function

The option 82 option, called the relay information option, records the location information of the DHCP client. If the DHCP relay is configured to enable the Option 82 function, and after receiving the request packet sent by the DHCP client to the DHCP server, and the request packet does not have option 82 option, add Option 82 to the request packet and forward it to the DHCP server; If the DHCP relay is configured to support Option 82 function and Option 82 has been carried in the request packet, the next processing will be carried out according to the action configured by the **ip dhcp relay information strategy** command, and then the packet will be forwarded to the server; If the DHCP relay receives the DHCP reply packet with the option 82, the option 82 option will be deleted and forwarded to the DHCP client.

Table 14-19 Configuring the Option82 function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable DHCP relay to support Option 82 | **ip dhcp relay information option** | Mandatory<br>By default, do not enable DHCP relay to support Option 82. |
| Configure the processing policy when the DHCP relay receives the request packet with option 82 from the client | **ip dhcp relay information strategy { drop \| keep \| replace }** | Optional<br>The packet with Option 82 uses the replace action. |

| Step | Command | Description |
|---|---|---|
| Configure Option 82 function | **ip dhcp relay information** { **option \| remote-id** { **ascii** *ascii-string* \| **hex** *hex-string* }\| **circuit-id** { **ascii** *ascii-string* \| **hex** *hex-string* } } | Mandatory'<br>By default, do not configure the Option 82 function. |

### Configure the Source Address of the DHCP Relay Packet as the Relay Address

When the DHCP relay sends the packet, the default used source address of the packet is the auto selected address of the system. In some special environments, the DHCP server cannot communicate the address, so the user can configure the DHCP relay to fill in the source address of the packet as the relay address.

Table 14-20 Configure the source address of the DHCP relay packet as the relay address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the source address of the DHCP relay packet | **ip dhcp relay source-address relay-address** | Mandatory<br>By default, the source address of the DHCP relay packet is the out interface address of the route to the DHCP server |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the source address of the DHCP relay packet | **ip dhcp relay source-address** *ip-address* | Mandatory<br>By default, the source address of the DHCP relay packet is the out interface address of the route to the DHCP server. |

**Note:**

- The source address configured by the command **ip dhcp relay source-address** *ip-address* is the interface address of the local device. Meanwhile, the interface address should be in the same vrf as the relay interface. Otherwise, the relay packet cannot be sent.

- If the command **ip dhcp relay source-address** *ip-address* is configured in the interface mode, and the command **ip dhcp relay source-address relay-address** is configured in the global mode, the priority of the former is higher than that of the latter. The DHCP relay will use the configured *ip-address* to fill in the source address of the packet sent by the DHCP relay to the DHCP server.

### Configure DHCP Server Address

When the interface receives the DHCP packet from the DHCP client, it will relay the packet to the configured DHCP server, and the DHCP server will assign the IP address.

Table 14-21 Configure the DHCP server address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the DHCP server address | **ip dhcp relay server -address** *ip-address* | Mandatory<br>By default, do not configure the DHCP server address. |

## 14.2.5. DHCP Monitoring and Maintaining

Table 6-22 DHCP monitoring and maintaining

| Command | Description |
|---|---|
| **clear ip dhcp pool** *pool-name* { **lease [** *ip-address* **]** \| **conflict [***ip-address***]** } | Clear the dynamic lease information in the address pool or address information with address conflict |
| **clear ip dhcp server interface [***interface-name* **] statistics** | Clear the key statistics of the packet interaction between DHCP server and client or relay |
| **clear ip dhcp relay statistics** | Clear the statistics on DHCP relay device |

| Command | Description |
|---|---|
| **show ip dhcp server interface** *interface-name* **[statistics]** | Display the address pool information associated with the specified interface, or display the key information statistics when the DHCP server interacts with the client or relay |
| **show ip dhcp pool** *pool-name* **{ summary \| ping_list \| offer_list \| excluded_list \| conflict_list \| lease \| binding }** | Display the summary information of the specified address pool, the address information being pinged or the sent OFFER packet, the address information of the REQUEST packet waiting for the DHCP client to respond, display the excluded address information under address pool or display the address information of the address conflict under address pool or display the dynamic lease information under address pool or static binding information under the address pool |
| **show ip dhcp pool** *pool-name* **specific { ip-address** *ip-address* **\| mac-address** *mac-address* **}** | Display the information about the IP address or MAC address specified under the address pool |
| **show show ip dhcp relay [interface** *interface-name* **]** | Display the packet statistics on DHCP relay devices. |

## 14.3. DHCP Typical Configuration Example

### 14.3.1. Configure a DHCP Server to Statically Allocate IP Addresses

**Network Requirements**

- Device2 acts as a DHCP server to allocate IP addresses, gateway IP addresses, and DNS server IP addresses in a static manner.
- The DHCP server allocates an IP address to PC in MAC binding mode.

## Network Topology

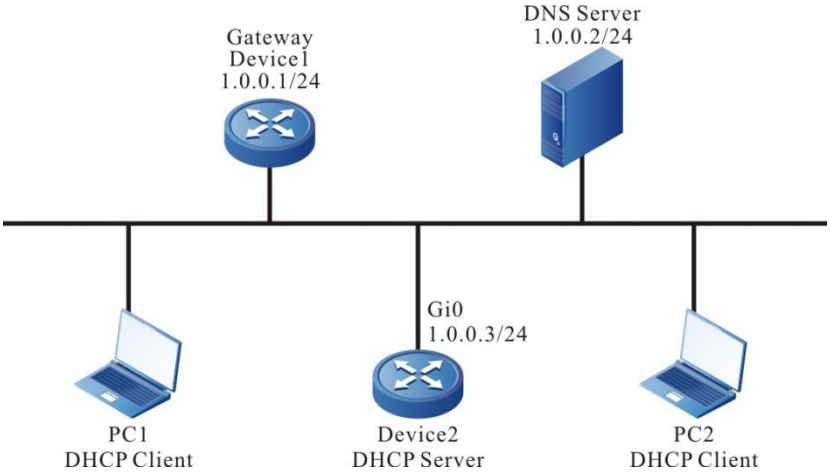



Figure 14-1 Configuring a DHCP server to statically allocate IP addresses

## Configuration Steps

**Step 1:** Configure IP addresses for interface of Device2. (Omitted)

```
Device2#configure terminal
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#ip address 1.0.0.3 255.255.255.0
Device2(config-if-gigabitethernet0)#exit
```

**Step 2:** Configure statically bound address pools and parameters.

#Configure the address pool mac-binding, and allocate an IP address to PC1 in static MAC binding mode.

```
Device2(config)#ip dhcp pool mac-binding
Device2(dhcp-config)#host 1.0.0.11 255.255.255.0
Device2(dhcp-config)#hardware-address 00e0.00c1.013d
Device2(dhcp-config)#default-router 1.0.0.1
Device2(dhcp-config)#dns-server 1.0.0.2
Device2(dhcp-config)#exit
```

#Configure the address pool client-id-binding, and allocate an IP address to PC2 in static client ID binding mode.

```
Device2(config)#ip dhcp pool client-id-binding
Device2(dhcp-config)#host 1.0.0.12 255.255.255.0
Device2(dhcp-config)#client-identifier 0100.e04c.113c.f2
Device2(dhcp-config)#default-router 1.0.0.1
Device2(dhcp-config)#dns-server 1.0.0.2
Device2(dhcp-config)#exit
```

**Step 3:** Check the result.

#On Device2, use the command **show ip dhcp server interface gigabitethernet0** to view the associated address pool of the interface.

```
Device2(config)#exit
Device2#show ip dhcp server interface gigabitethernet0
DHCP server status information:
DHCP server is enabled on interface: gigabitethernet0
Vrf : global
DHCP server pool information:
Available directly-connected pool:
Interface IP      Pool name          Pool Range          Pool utilization
-----------   ------------------  ------------------  ----------------
 1.0.0.3/24       mac-binding         1.0.0.4 – 1.0.0.254        0.00%
```

#On Device2, use the command **show ip dhcp pool mac-binding binding** to view the binding IP address assigned for PC.

```
Device2#show ip dhcp pool mac-binding binding
IP Address     MAC Address       Vendor Id    Type     Time Left(s)
------------ ----------------- ------------ --------  -------------
1.0.0.11       00e0.00c1.013d       Global     Binding       NA
```

#On Device2, use the command **show ip dhcp pool mac-binding lease** to view the address assigned for the PC.

```
Device#show ip dhcp pool danymic-pool2 lease
IP Address       MAC Address       Vendor Id       Type    Time Left(s)
------------ ------------------ ------------------ -------- -------------
1.0.0.11       00e0.00c1.013d         Global         Lease    107980
```

Check that the IP address, gateway IP address and DNS server address obtained on PC are correct.

## 14.3.2. Configure a DHCP Server to Dynamically Allocate IP Addresses

### Network Requirements

- Two Ethernet interfaces of Device, gigabitethernet0 and gigabitethernet1, are respectively configured with IP addresses in the 1.0.0.3/24 and 2.0.0.3/24 network segments.
- The DHCP server Device dynamically allocates IP addresses in the 1.0.0.0/24 and 2.0.0.0/24 network segments to the two clients in the directly-connected physical network.
- The IP addresses in network segment 1.0.0.0/24 have a one-day lease, the gateway address is 1.0.0.3, and the DNS server address is 2.0.0.4. The IP addresses in network segment 2.0.0.0/24 have a three-day lease the gateway address is 2.0.0.3, and the DNS server address is 2.0.0.4.
- The first 10 IP addresses in network segments 1.0.0.0/24 are reserved and cannot be allocated.

## Network Topology

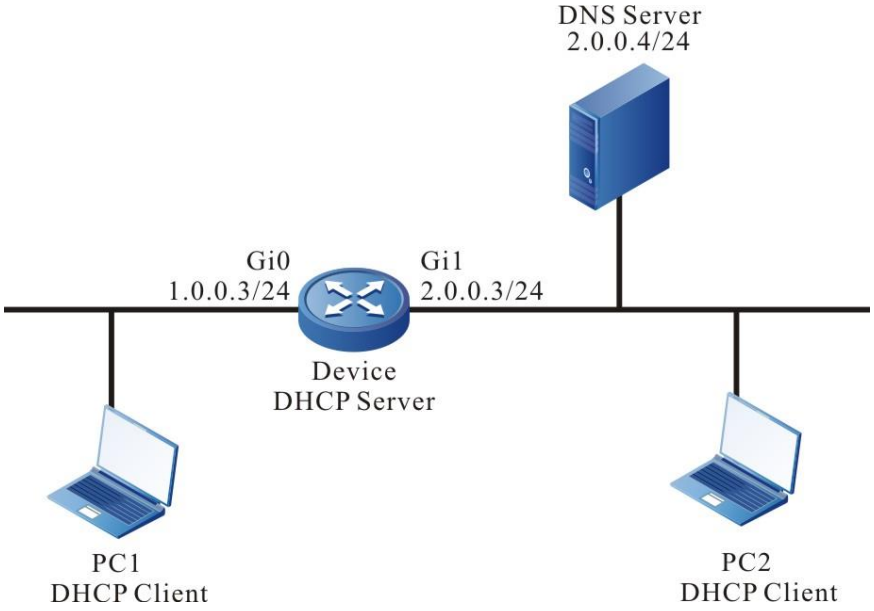



Figure 14-2 Networking for configuring the DHCP server to dynamically allocate IP addresses

## Configuration Steps

**Step 1:** Configure the IP address of the device interface and make the interface work in the DHCP server mode.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip address 1.0.0.3 255.255.255.0
Device(config-if-gigabitethernet0)#ip dhcp server
Device(config-if-gigabitethernet0)#exit
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip address 2.0.0.3 255.255.255.0
Device(config-if-gigabitethernet1)#ip dhcp server
Device(config-if-gigabitethernet1)#exit
```

**Step 2:** On the DHCP server Device, configure two dynamic address pools and their parameters.

#Configure the first 10 IP addresses in the two address pools to be reserved.

```
Device(config)#ip dhcp excluded-address 1.0.0.1 1.0.0.10
Device(config)#ip dhcp excluded-address 2.0.0.1 2.0.0.10
```

#Configure address pool dynamic-pool1 and its parameters (including address range, gateway, DNS, address lease).

```
Device(config)#ip dhcp pool dynamic-pool1
Device(dhcp-config)#network 1.0.0.0 255.255.255.0
Device(dhcp-config)#default-router 1.0.0.3
Device(dhcp-config)#dns-server 2.0.0.4
Device(dhcp-config)#lease 1 0 0
```

```
Device(dhcp-config)#exit
```

#Configure address pool dynamic-pool2 and its parameters (including address range, gateway, DNS address, address lease).

```
Device(config)#ip dhcp pool dynamic-pool2
Device(dhcp-config)#network 2.0.0.0 255.255.255.0
Device(dhcp-config)#default-router 2.0.0.3
Device(dhcp-config)#dns-server 2.0.0.4
Device(dhcp-config)#lease 3 0 0
Device(dhcp-config)#exit
```

**Step 3:** Check the result.

#View the information of the associated address pool of the server on Device.

```
Device(config)#exit
Device#show ip dhcp server interface g0
DHCP server status information:
DHCP server is enabled on interface: gigabitethernet0
Vrf : global
DHCP server pool information:
Available directly-connected pool:
Interface IP      Pool name         Pool Range        Pool utilization
-----------   ------------------  ------------------  ----------------
1.0.0.3/24      dynamic-pool1     1.0.0.0 – 1.0.0.255       0.00%
Device#show ip dhcp server interface g1
DHCP server status information:
DHCP server is enabled on interface: gigabitethernet1
Vrf : global
DHCP server pool information:
Available directly-connected pool:
Interface IP      Pool name         Pool Range        Pool utilization
-----------   ------------------  ------------------  ----------------
1.0.0.3/24      dynamic-pool2     2.0.0.0 – 2.0.0.255       0.00%
```

#View the IP address information assigned for the client on Device.

```
Device#show ip dhcp pool danymic-pool1 lease
IP Address      MAC Address       Vendor Id      Type   Time Left(s)
-----------  -----------------  ------------------ -------- -------------
1.0.0.11      0001.7a6a.0268       Global       Lease    86390
Device#show ip dhcp pool danymic-pool2 lease
IP Address      MAC Address       Vendor Id      Type   Time Left(s)
```

```
------------ ----------------- ------------------ -------- ------------
2.0.0.11     0001.7a6a.0269    Global     Lease    259194
```

#View the statistics information of the configured IP address pool on Device.

```
Device#show ip dhcp pool dynamic-pool1 summary
Pool: dynamic-pool1
Pool Configuration  : 1.0.0.0  255.255.255.0
Pool Range          : 1.0.0.0  1.0.0.255
Pool Utilization    : 0.39%
 VRF                : global
DNS Server          : 2.0.0.4
 Default Router     : 1.0.0.3
Lease Time          : 1 Days 0 Hours 0 Minutes
 Free Addresses     : 243
Static Bind         : 0
 Lease Count        : 1
PingList            : 0
OfferList           : 0
 ConflictList       : 0
 ExcludeList        : 12

Device#show ip dhcp pool dynamic-pool2 summary
Pool: dynamic-pool2
 Pool Configuration  : 2.0.0.0  255.255.255.0
 Pool Range          : 2.0.0.0  2.0.0.255
Pool Utilization    : 0.39%
 VRF                : global
DNS Server          : 2.0.0.4
 Default Router     : 2.0.0.3
 Lease Time         : 3 Days 0 Hours 0 Minutes
Free Addresses      : 243
 Static Bind        : 0
Lease Count         : 1
PingList            : 0
OfferList           : 0
 ConflictList       : 0
ExcludeList         : 12
```

On the DHCP client, view that the IP address is got correctly.

**Note:**

- The IP addresses in the address pool must be within the network segment range of the interface that provides the service.

## 14.3.3. Configure a DHCP Relay

### Network Requirements

- Device1 is a DHCP server, and Device2 interface enables DHCP relay function.
- The DHCP server provides services for clients of 1.0.0.0/24 network segment, and retains the top 10 IP addresses.
- The DHCP client obtains the IP address through the DHCP relay.

### Network Topology

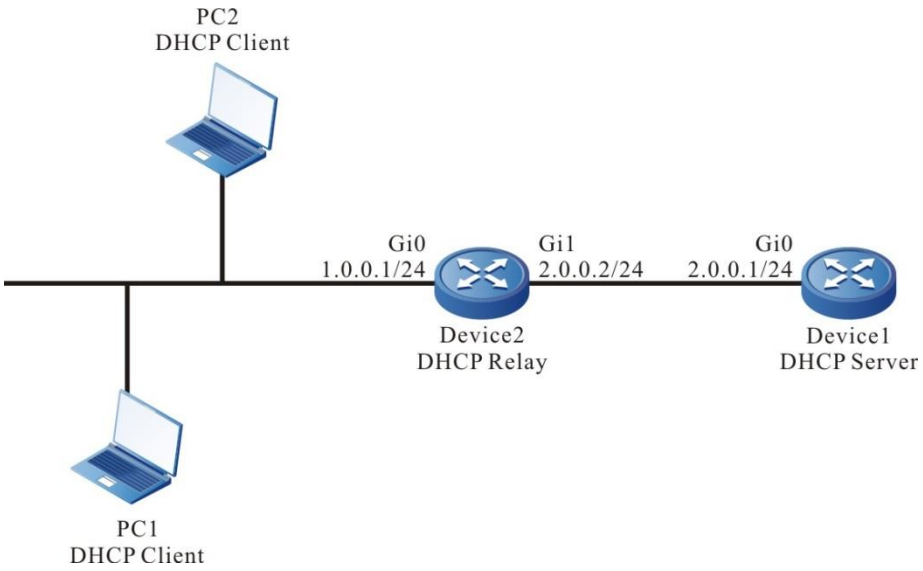



Figure 14-3 Networking for configuring a DHCP relay

### Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the IP address pool of Device1 and the reserved IP address, and work in the DHCP server mode.

#Configure the DHCP server.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#ip dhcp server
Device1(config-if-gigabitethernet0)#exit
```

#Configure IP addresses which are from 1.0.0.1 to 1.0.0.10 not to be allocated.

```
Device1#configure terminal
Device1(config)#ip dhcp excluded-address 1.0.0.1 1.0.0.10
```

#Configure IP address pool dynamic-pool for Device1.

```
Device1(config)#ip dhcp pool dynamic-pool
Device1(dhcp-config)#network 1.0.0.0 255.255.255.0
```

```
Device1(dhcp-config)#default-router 1.0.0.1
Device1(dhcp-config)#lease 1 0 0
Device1(dhcp-config)#exit
```

#Configure a static route to network segment 1.0.0.0/24.

```
Device1(config)#ip route 1.0.0.0 255.255.255.0 2.0.0.2
```

**Step 3:** On the interface gigabitethernet0 of Device2, configure the DHCP server address as 2.0.0.1, and make the interface work in the relay mode.

```
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#ip dhcp relay
Device2(config-if-gigabitethernet0)#ip dhcp relay server-addres 2.0.0.1
Device2(config-if-gigabitethernet0)#exit
```

**Step 4:** Check the result.

#On Device1, query the IP addresses that have been allocated.

```
Device1#show ip dhcp pool dynamic-pool lease
IP Address      MAC Address      Vendor Id      Type   Time Left(s)
------------   --------------   --------------- --------   -----------
1.0.0.11       0001.7a6a.0268       Global       Lease      86387
```

Use the **show ip dhcp pool dynamic-pool lease** command to query the IP addresses that have been allocated to clients. The result shows that a client has obtained the IP address 1.0.0.11.

## 14.3.4. Configure the DHCP Relay to Support Option82

### Network Requirements

- On the DHCP relay device, Option82 is enabled.
- For Option82 sub-option Remote ID, 0102030405 is specified.
- The DHCP relay Device2 adds Option82 in a request packet and forwards the request to DHCP server. The DHCP server then allocates IP addresses in the 1.0.0.0/24 network segment to the client.

## Network Topology

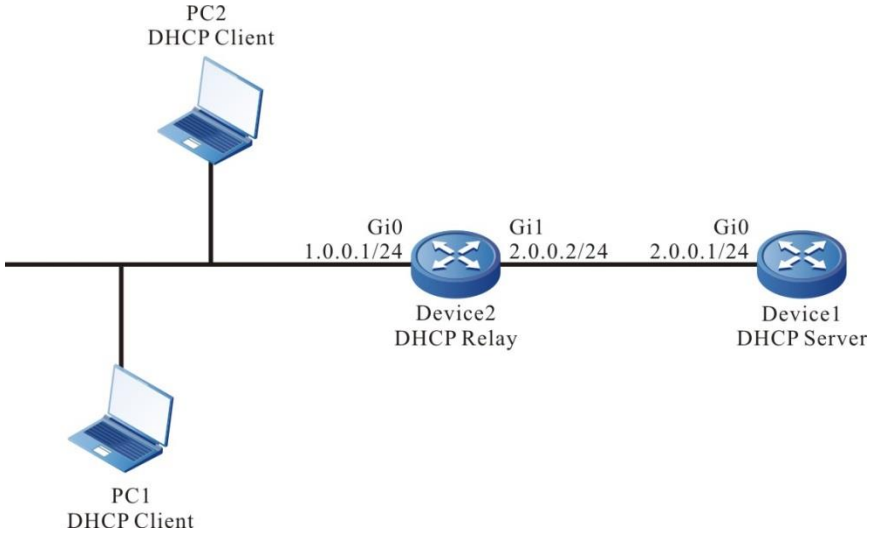



Figure 14-4 Networking for configuring the DHCP relay to support Option82

## Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the DHCP server.

```
Device1# configure terminal
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#ip dhcp server
Device1(config-if-gigabitethernet0)#exit
Device1(config)#ip dhcp pool dynamic-pool
Device1(dhcp-config)#network 1.0.0.0 255.255.255.0
Device1(dhcp-config)#default-router 1.0.0.1
Device1(dhcp-config)#exit
```

#Configure the static route to the segment 1.0.0.0/24.

```
Device1(config)#ip route 1.0.0.0 255.255.255.0 2.0.0.2
```

**Step 3:** Configure the DHCP relay Device2 and Option82 parameters.

#Set the IP address of the relay server to 2.0.0.1.

```
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#ip dhcp relay
Device2(config-if-gigabitethernet0)#ip dhcp relay server-address 2.0.0.1
```

#Enable Option82, and set sub-option Remote-ID to 0102030405.

```
Device2(config)#ip dhcp relay information option
Device2(config)#ip dhcp relay information remote-id hex 0102030405
```

**Step 4** Check the result.

#Query the information about the IP address assigned for the client on Device1.

```
Device1#show ip dhcp pool danymic-pool1 lease
IP Address      MAC Address       Vendor Id        Type   Time Left(s)
--------------- ----------------- ---------------- ------ ------------
1.0.0.2         0001.7a6a.0268    Global           Lease  107992
```

Check the IP address of the 1.0.0.0/24 network segment obtained by the network card on the DHCP client.

By capturing packets on the DHCP server, you can verify that the filling value of option 82 remote-id in the discover packet received by the server is 0102030405

**Note:**

- After Option82 is enabled, it sub-option Circuit ID is filled with the receiving interface index and the system ID of the relay device.

# 15. DHCPV6

## 15.1. Overview

It is hard to manage a large network. For example, in a network in which IPv6 addresses are manually allocated, IPv6 address conflicts are common. The only way of solving the problem is to dynamically allocate IPv6 addresses to the hosts. The Dynamic Host Configuration Protocol (DHCPv6) allocates IPv6 address to requesting hosts from an address pool. DHCPv6 also provides other information, such as DNS server address. DHCPv6 reduces the workload of the administrator in recording and tracking manually allocated IPv6 addresses.

DHCPv6 is a protocol that is based on UDP broadcast. The process for a DHCPv6 client to obtain an IPv6 address and other configuration information contains four phases:

SOLICT phase. When the DHCPv6 client logs into the network for the first time, it sends a DHCP SOLICT packet, whose source address is the linklocal address of the client and destination address is ff02::1:2.

ADVERTISE phase. After the DHCPv6 server receives the DHCP SOLICT broadcast packet sent by the client, it selects an IPv6 address from the corresponding IP address pool according to the policy, and sends the IP address and other parameters to the client in a DHCP ADVERTISE packet.

REQUEST phase. If the DHCP client receives response messages from multiple DHCP servers on the network, it selects one DHCP OFFER (usually the one that arrives first). Then it sends a DHCP REQUEST packet to the network, telling all DHCP servers the IP address of which server it will accept.

REPLY phase. After the DHCPv6 server receives the DHCPv6 REQUEST packet from the DHCPv6 client, it sends a DHCP ACK message containing the provided IPv6 address and other configuration to the DHCPv6 client, telling the DHCPv6 client that the DHCPv6 client can use the provided IPv6 address.

The IPv6 address that the DHCPv6 server allocates to the DHCPv6 client has a lease. After the lease expires, the DHCPv6 server will take back the allocated IPv6 address. When the lease term of the IPv6 address of the DHCPv6 client has passed half time, the DHCPv6 client sends a DHCP ENEW packet to the DHCPv6 server requesting to update its IPv6 address lease. If the DHCPv6 client can continue to use the IPv6 address, the DHCPv6 server responds with a DHCP REPLY packet, requesting the DHCPv6 client to update the lease. If the DHCPv6 DHCP client cannot to continue to use the IPv6 address, the DHCPv6 server does not respond.

During dynamic IPv6 address acquisition, request packets are sent in broadcast mode; therefore, DHCPv6 is applied only when the DHCPv6 client and server are in the same subnet. If multiple subnets exist in a network and the hosts of the subnets need to obtain configuration information such as IPv6 address through the DHCPv6 server, the hosts of the subnets communicate with the DHCPv6 server through a DHCPv6 relay to obtain IPv6 addresses and other configuration information.

## 15.2. DHCPv6 Function Configuration

Table 15-1 DHCPv6 Function List

| Configuration Tasks | |
|---|---|
| Configure a DHCPv6 address pool | Create a DHCPv6 address pool and specify the VRF attributes |
| | Configure an IPv6 address range |
| | Configure a DNS server address |
| | Configure the lease of an IPv6 address |
| | Configure IPv6 to bind with DUID and IAID |
| Configure other parameters of a DHCPv6 server | Configure the DHCPv6 server |
| | Configure the range of reserved IPv6 addresses |
| | Configure DHCPv6 ping detection parameters. |
| | Configure the data log function of the DHCPv6 server |
| Configure the functions of a DHCPv6 client | Configure a DHCPv6 client |
| | Configure the DHCPv6 Option 16 function |
| Configure the DHCPv6 relay function | Configure a DHCPv6 relay. |
| | Configure the source address of the DHCPv6 relay packet |
| | Configure the DHCPv6 server address |
| | Configure the DHCPv6 interface-id option |

## 15.2.1. Configure a DHCPv6 Address Pool

### Configuration Condition

None

### Create a DHCPv6 Address Pool

A DHCPv6 server needs to select and allocate IPv6 addresses and other parameters from a DHCPv6 address pool. Therefore, a DHCPv6 address pool must be created for the DHCPv6 server.

Table 15-2 Creating a DHCPv6 Address Pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Create one DHCPv6 address pool and enter the DHCPv6 configuration mode | **ipv6 dhcp pool** *pool-name*[ **vrf** *vrf-name* ] | Mandatory<br>By default, the system does not create the DHCPv6 address pool. |

**Note:**

- Address pools fall into two types: Network and Range. The two types of address pools can be configured respectively through the network and range commands.

### Configure an IPv6 Address Range

On a DHCPv6 server, each DHCPv6 address pool must be configured with an IPv6 address range to allocate IPv6 addresses to DHCPv6 clients.

Table 15-3 Configuring an IPv6 Address Range

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCPv6 configuration mode | **ipv6 dhcp pool** *pool-name*[ **vrf** *vrf-name* ] | - |
| Configure an IPv6 address range for an address pool of the Network type. | **network** *ipv6-address*/*prefix-length* | Optional.<br>By default, an IPv6 address range is not configured for an address pool. |

| Step | Command | Description |
|---|---|---|
| Configure an IPv6 address range for an address pool of the Range type. | **range** *low-ipv6-address high-ipv6-address prefix-length* | Optional.<br>By default, an IPv6 address range is not configured for an address pool. |

**Note:**

- Modify the type of the address pool from network to range (or from range to network). If the new address range intersects with the old address range, the command line will prompt the user whether to perform the operation. If yes, it will delete the address configuration (static binding) and dynamic lease related to the address pool; if the actual effective range of the new address includes the actual effective range of the old address, the address pool reserves the relevant address configuration (static binding) under the address pool. But dynamic leases are deleted.

### Configure a DNS Server Address

On a DHCPv6 server, you can configure the DNS server address respectively for each DHCPv6 address pool. When a DHCPv6 server allocates an IPv6 address for a DHCPv6 client, it also sends the DNS server address to the client.

When the DHCPv6 client starts dynamic domain name resolution, it queries the DNS server.

Table 15-4 Configuring a DNS Server Address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCPv6 configuration mode | **ipv6 dhcp pool** *pool-name***[ vrf** *vrf-name* **]** | - |
| Configure a DNS server address | **dns-server** { *ipv6-address*&<1-8> \| **autoconfig** } | Mandatory.<br>By default, the DNS server address is not configured. |

### Configure the Lease of an IPv6 Address

The IPv6 address that the DHCPv6 server allocates to the DHCPv6 client has a lease. After the lease expires, the server will take back the allocated IPv6 address. If the DHCPv6 client wants to continue to use the IPv6 address, it must have the IPv6 address lease updated.

On the DHCPv6 server, you can configure an IPv6 address lease for each DHCPv6 address pool.

Table 15-5 Configuring the Lease of an IPv6 Address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCPv6 configuration mode | **ipv6 dhcp pool** *pool-name*[ **vrf** *vrf-name* ] | - |
| Configure the lease of the IPv6 address | **lease preferred-lifetime** *preferred-lifetime* **valid-lifetime** *valid-lifetime* | Mandatory<br>By default, the preferred-lifetime is 604800 (sevent days), and valid-lifetime is 2592000s (30 days). |

**Configure IPv6 to Bind with DUID, IAID**

Configure IPv6 to bind with client DUID and IAID. When specifying the client of DUID and IAID to request for allocating the IPv6 address to the DHCPv6 server, the DHCPv6 server will allocate the IPv6 address it binds to. As long as the DIID and IAID of the client remain unchanged, the IPv6 address acquired by the client from the server is the same every time.

Table 15-6 Configure Ipv6 to bind with DUID, IAID

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the DHCPv6 configuration mode | **ipv6 dhcp pool** *pool-name*[ **vrf** *vrf-name* ] | - |
| Configure Ipv6 to bind with DUID, IAID | **bind** *ipv6-address* **duid** *duid* [ **iaid** *iaid* ] | Mandatory<br>By default, do not configure IPv6 to bind with DUID, IAID. |

**Note:**

- The command is valid only for the Range and Network address pools.
- When configuring the static binding of the same duid and iaid, the address pool permits binding five IPv6 addresses.
- When the configured static binding only specifies duid, not specifying iaid, the address pool only permits binding one IPv6 address.

## 15.2.2. Configure Other Parameters of a DHCPv6 Server

### Configuration Condition

None

### Configure DHCPv6 Server

After configuring the interface to work in the DHCPv6 server mode, the DHCPv6 server will distribute the IPv6 address and other network parameters to the client when the interface receives the DHCPv6 request packet from the DHCPv6 client.

Table 15-7 Configure the DHCPv6 server

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the DHCPv6 server | **ipv6 dhcp server** | Mandatory<br>By default do not configure the DHCPv6 server. |

### Configure the Range of Reserved IPv6 Addresses

In a DHCPv6 address pool, some IPv6 addresses are reserved for some special devices, and some IPv6 addresses conflict with the IPv6 addresses of other hosts in the network. Therefore, the IPv6 addresses cannot be dynamically allocated.

Table 15-8 Configure the Range of Reserved IPv6 Addresses

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the range of reserved IPv6 addresses. | **ipv6 dhcp excluded-address** *low-ipv6-address* [ *high-ipv6-address* ] [**vrf** *vrf-name*] | Mandatory.<br>By default, the range of reserved IPv6 addresses is not configured.<br>The IPv6 addresses in the reserved IP address range will not be allocated. |

### Configure DHCPv6 Ping Detection Parameters

To prevent an IPv6 address conflict, before dynamically allocating an IPv6 address to a DHCPv6 client, a DHCPv6 server must detect the IPv6 address. The detection operation is performed through the ping operation. The DHCPv6 server determines whether an IPv6 address conflict exists by checking whether an ICMP echo response packet is received within the specified time.

Table 15-9 Configuring DHCPv6 Ping Detection Parameters

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure DHCPv6 ping detection parameters | **ipv6 dhcp ping** { **packets** *packet-num* \| **timeout** *milliseconds* } | Mandatory.<br>By default, the number of ping packets is 1, and the timeout time is 50 ms. |

### Configure the Data Log Function of the DHCPv6 Server

After enabling the data log function of the DHCPv6 server, the distribution of the address pool on the DHCPv6 server is recorded in the data log.

Table 15-10 Configuring the Data Log Function of the DHCPv6 Server

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable the data log function of the DHCPv6 server | **ipv6 dhcp logging security-data** | Mandatory<br>By default, do not enable the data log function. |

## 15.2.3. Configure the Functions of a DHCPv6 Client

### Configuration Condition

None

### Configure a DHCPv6 Client

The interface of the DHCPv6 client obtains an IPv6 address and other parameters through DHCP.

Table 15-11 Configuring a DHCPv6 Client

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the DHCPv6 client to obtain an IPv6 address | **ipv6 dhcp client address [ rapid-commit ]** | Mandatory.<br>By default, the DHCPv6 client is not configured to obtain an IPv6 address. |
| Configure the DHCPv6 client to get the IPv6 prefix | **ipv6 dhcp client pd** *pool-name* **[ rapid-commit ]** | Mandatory<br>By default, do not configure the DHCPv6 client to get the IPv6 prefix. |

### Configure DHCPv6 Option 16 Function

The content of DHCPv6 option 16 is the manufacturer ID. During the request process of the DHCPv6 client, option 16 can be carried. The server can customize IPv6 address allocation policy according to this option.

Table 15-12 Configure DHCPv6 Option 16 function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the Option 16 function in global mode | **ipv6 dhcp client vendor-class-identifier { enable \| enterprise-number** *enterprise-number* **\| content** *hex-string* **}** | Mandatory<br>By default, the DHCPv6 client does not carry manufacturer ID. |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the Option 16 function in the interface mode | **ipv6 dhcp client vendor-class-identifier { enable \| enterprise-number** *enterprise-number* **\| content** *hex-string* **}** | Mandatory<br>By default, the DHCPv6 client does not carry the manufacturer ID. |

**Note:**

- When the option16 content and enterprise-number are configured simultaneously in global mode or interface mode, content takes precedence.
- When DHCPv6 client option 16 is configured in both global and interface, the configuration in interface takes precedence.

## 15.2.4. Configure the Function of a DHCPv6 Relay

### Configuration Condition

None

### Configure a DHCPv6 Relay

If multiple subnets exist in a network and the hosts of the subnets need to obtain configuration information such as IPv6 address through the DHCPv6 server, the hosts of the subnets communicate with the DHCPv6 server through a DHCPv6 relay to obtain IPv6 addresses and other configuration information. If an interface is configured to work in DHCPv6 relay mode, after the interface receives DHCPv6 packets from a DHCPv6 client, it relays the packet to the specified DHCPv6 server. The DHCPv6 server then allocates an IP address.

Table 15-13 Configuring a DHCPv6 Relay

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the DHCPv6 relay function | **ipv6 dhcp relay** | Mandatory<br>By default, do not configure the DHCPv6 relay function. |

### Configure Source Address of DHCPv6 Relay

DHCPv6 relays the DHCPv6 client to the source address of the server packet. By default, use the address of the egress interface of the route to the DHCPv6 server. In some special environment, the DHCPv6 server cannot communicate with the address. Therefore, users can configure the source address of the DHCPv6 relay packet to the DHCPv6 server and the LinkAddr field in the packet through the **ipv6 dhcp relay source-address** command.

Table 15-14 Configure the source address of the DHCPv6 relay packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the source address of the DHCPv6 relay | **ipv6 dhcp relay source-address** *ipv6-address* | Mandatory<br>By default, do not configure the source address of the DHCPv6 relay packet. |

### Configure the Address of the DHCPv6 Server

When the interface receives the DHCPv6 packet sent by the DHCPv6 client, relay the packet to the configured DHCPv6 server, which distributes the IPv6 address.

Table 15-15 Configure the address of the DHCPv6 server

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the address of the DHCPv6 server | **ipv6 dhcp relay server -address** *ipv6-address* | Mandatory<br>By default, do not configure the address of the DHCPv6 server. |

### Configure DHCPv6 interface-id Option

The command is used to configure the interface-id option supported by DHCPv6 relay.

Table 15-16 Configure DHCPv6 interface-id option

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure DHCPv6 interface-id option | **ipv6 dhcp relay interface –id [ interface \| user-define** *defined-string***]** | Mandatory<br>By default, do not configure DHCPv6 interface-id option. |

### Configure DHCPv6 remote-id Option

This command is used to configure the filling mode of the remote ID option supported by DHCPv6 relay.

Table 15-17 Configure DHCPv6 server address

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the DHCPv6 remote-id option | **ipv6 dhcp relay remote–id user-define** *defined-string* | Mandatory<br>By default, do not configure the filling mode of remote-id option. |

## 15.2.5. DHCPv6 Monitoring and Maintaining

Table 15-18 DHCPv6 Monitoring and Maintaining

| Command | Description |
|---|---|
| **clear ipv6 dhcp pool** *pool-name* { **lease** \| **conflict** [*ipv6-address*] } | Clear the dynamic lease information or conflict address information in the address pool |
| **clear ipv6 dhcp server interface** [*interface-name* ] **statistics** | Clear the key information statistics when the DHCPv6 server interacts packets with the client or relay |
| **clear ipv6 dhcp relay statistics** | Clear the statistics information on the DHCPv6 relay device |
| **show ipv6 dhcp server interface** *interface-name* **[statistics]** | Display the associated address pool information in the specified interface or display the key information statistics when the DHCPv6 server in the specified interface interacts packets with the client or relay |

| Command | Description |
|---|---|
| **show ipv6 dhcp pool** *pool-name* **{ summary \| ping_list \| offer_list \| excluded_list \| conflict_list \| lease \| binding }** | Display the summary information of the specified address pool or the address information of the ping check or the information about the address that has sent OFFER packet and is waiting for DHCPv6 client to reply the REQUEST packet or display the exclude address information in the address pool or display the conflict address information in the address pool or display the dynamic lease information in the address pool or display the static binding information in the address pool. |
| **show ipv6 dhcp pool** *pool-name* **specific { ipv6-address** *ipv6-address* **\| duid** *duid* **}** | Display the specified ipv6 address or client DUID information in the address pool |
| **show ipv6 dhcp relay [interface** *interface-name* **]** | Display the packet statistics information on the DHCPv6 relay device |

## 15.3. DHCPv6 Typical Configuration Example

### 15.3.1. Configure a DHCPv6 Server to Statically Allocate IPv6 Addresses

#### Network Requirements

- Device2 acts as a DHCPv6 server to allocate IPv6 addresses and DNS server IPv6 addresses in a static manner.
- The DHCPv6 server allocates an IP address to PC1 in DUID binding mode, and allocates an IPv6 address to PC2 in the DUID+IAID binding mode.

## Network Topology

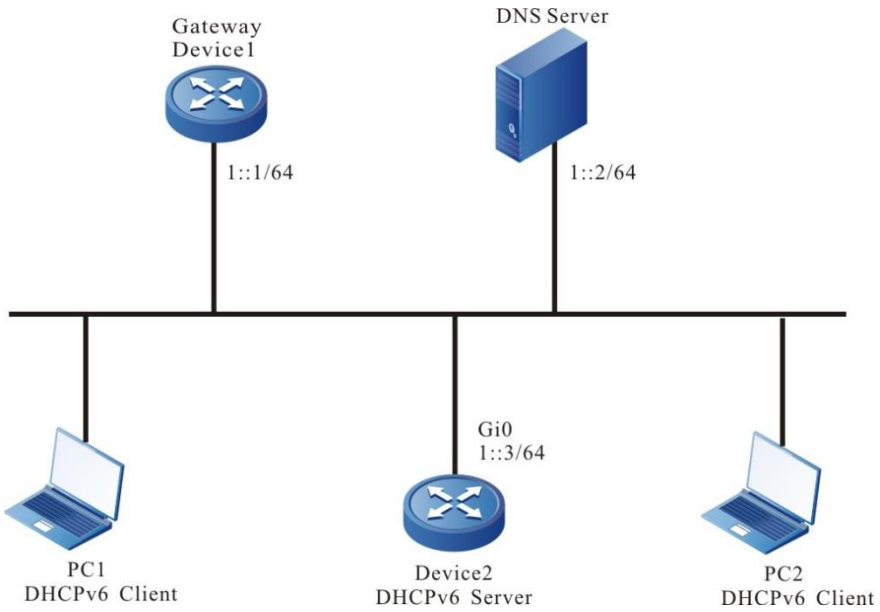



Figure 15-1 Configuring a DHCPv6 Server to Statically Allocate IPv6 Addresses

## Configuration Steps

**Step 1:** Configure the IPv6 address of the Device2 interface and the DHCPv6 server.

```
Device2#configure terminal
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#ipv6 address 1::3/64
Device2(config-if-gigabitethernet0)#ipv6 dhcp server
Device2(config-if-gigabitethernet0)#exit
```

**Step 2:** Configure the static binding address pool and parameters.

#Configure the address pool binding, and adopt the DUID binding mode to distribute IPv6 address for PC1. Adopt the static DUID+IAID binding mode to distribute the IPv6 address for PC2.

```
Device2(config)#ipv6 dhcp pool binding
Device2(dhcp6-config)#bind 1::11 duid 0002000016133030303137616366356466 34
Device2(dhcp6-config)#bind 1::12 duid 000200001613636364383166313037616239 iaid 00010071
Device2(dhcp6-config)#dns-server 1::2
Device2(dhcp6-config)#exit
```

**Step 3:** Check the result.

#Check the association of the server interface and address.

```
Device2#show ipv6 dhcp server interface gigabitethernet 0
DHCPv6 server status information:
DHCP server is enabled on interface: gigabitethernet 0
```

```
Vrf : global

DHCPv6 server pool information:
Available directly-connected pool:
  Interface IP: 1::1/64
    Pool name: binding
    Range:
      min: 101::
      max: 101::ffff:ffff:ffff:ffff
    utilization: 0.00%
```

#Check the static binding of the server.

```
Device2#show ipv6 dhcp pool binding binding
IPv6 Address          Duid                                 Iaid     Type    Time Left(s)
------------ ------------------------------------     -------- ------- ------
------
1::11      00020000161330303031376163663564663 4    00000000  Binding     NA
1::12      00020000161363636438316631303761623 9    00010071  Binding     NA
```

#On Device2, query the IPv6 addresses distributed for PC1 and PC2 via the **show ipv6 dhcp pool binding lease** command.

```
Device2#show ipv6 dhcp pool mac-binding lease
IPv6 Address          Duid                                 Iaid     Type     Time Left(s) -----------
- ------------------------------------     -------- -------  ------------
1::11      00020000161330303031376163663564663 4     00000000  Lease     2591974
1::12      00020000161363636438316631303761623 9     00010071  Lease     2591974
```

On PC1 and PC2, check whether the got IPv6 addresses, and the IPv6 address of the DNS server are correct.

## 15.3.2. Configure a DHCPv6 Server to Dynamically Allocate IPv6 Addresses

### Network Requirements

- Two Ethernet interfaces of Device, gigabitethernet0 and gigabitethernet 1, are respectively configured with IPv6 addresses 1::3/64 and 2::3/64.
- The DHCPv6 server Device dynamically allocates IPv6 addresses 1::/64 and 2::/64 to the two clients in the directly-connected physical network.
- The addresses in network segment 1::/64 have a one-day lease, the DNS server address is 2::4. The addresses in network segment 2::/64 have a three-day lease the gateway address is 2::3, the DNS server address is 2::4.
- The first 10 IPv6 addresses in network segments 1::/64 and 2::/64 are reserved and cannot be allocated.

## Network Topology

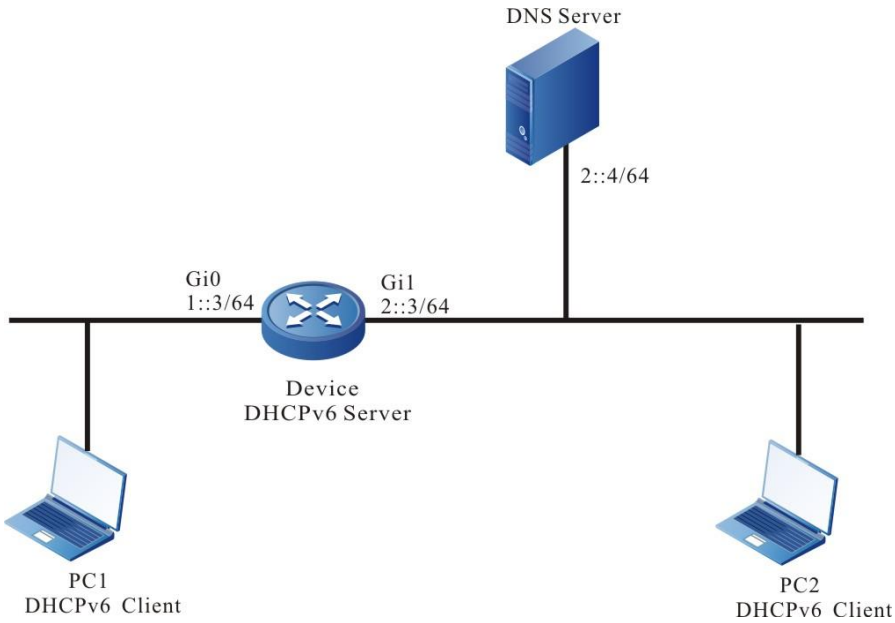



Figure 15-2 Configuring a DHCPv6 Server to dynamically Allocate IPv6 Addresses

## Configuration Steps

**Step 1:** Configure the IPv6 address of the interface (omitted).

**Step 2:** On the DHCPv6 server Device1, configure two dynamic address pools and their parameters.

#Configure the DHCPv6 server.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet 0)#ipv6 dhcp server
Device(config-if-gigabitethernet 0)#exit
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet 1)#ipv6 dhcp server
Device(config-if-gigabitethernet 1)#exit
```

#Configure the first 10 IP addresses in the two address pools to be reserved.

```
Device(config)#ipv6 dhcp excluded-address 1::0 1::9
Device(config)#ipv6 dhcp excluded-address 2::0 2::9
```

#Configure address pool dynamic-pool1 and its parameters (including address range, DNS address, address lease).

```
Device(config)#ipv6 dhcp pool dynamic-pool1
Device(dhcp6-config)#network 1::/64
Device(dhcp6-config)#dns-server 2::4
Device(dhcp6-config)#lease preferred-lifetime 86300 valid-lifetime 86400
Device(dhcp6-config)#exit
```

#Configure address pool dynamic-pool2 and its parameters (including address range, DNS address, address lease).

```
Device(config)#ip DHCPv6 pool dynamic-pool2
Device(dhcp6-config)#network 2::/64
Device(dhcp6-config)#dns-server 2::4
Device(dhcp6-config)#lease preferred-lifetime 259100 valid-lifetime 259200
Device(dhcp6-config)#exit
```

**Step 3:** Check the result.

#On Device, query the IPv6 addresses that are allocated to clients.

```
Device#show ipv6 dhcp pool dynamic-pool1 lease
IPv6 Address          Duid                    Iaid    Type     Time Left(s)
------------    ---------------------------    --------  -------   ---------
---
1::a       00020000161330303031376163663564663 4  00000000  Lease     86390
Device2#show ipv6 dhcp pool dynamic-pool2 lease
IPv6 Address       Duid                 Iaid       Type     Time Left(s)
------------  -------------------------    --------   -------   ---------
---
2::a     00020000161330303031376163663564663 4 00000000  Lease     2591974
```

On the DHCPv6 clients, query whether the IPv6 addresses have been obtained properly.

**Caution:**

- The IPv6 addresses in the address pool must be within the network segment range of the interface that provides the service.

## 15.3.3. Configure DHCPv6 Relay

### Network Requirements

- Device1 is the DHCPv6 server, and the interface of Device2 enables the DHCPv6 relay function.
- The DHCPv6 server provides the service for the client of the segment 1::/64, and the first ten IPv6 addresses are reserved.
- The DHCPv6 client gets the IPv6 address via DHCPv6 relay.

## Network Topology

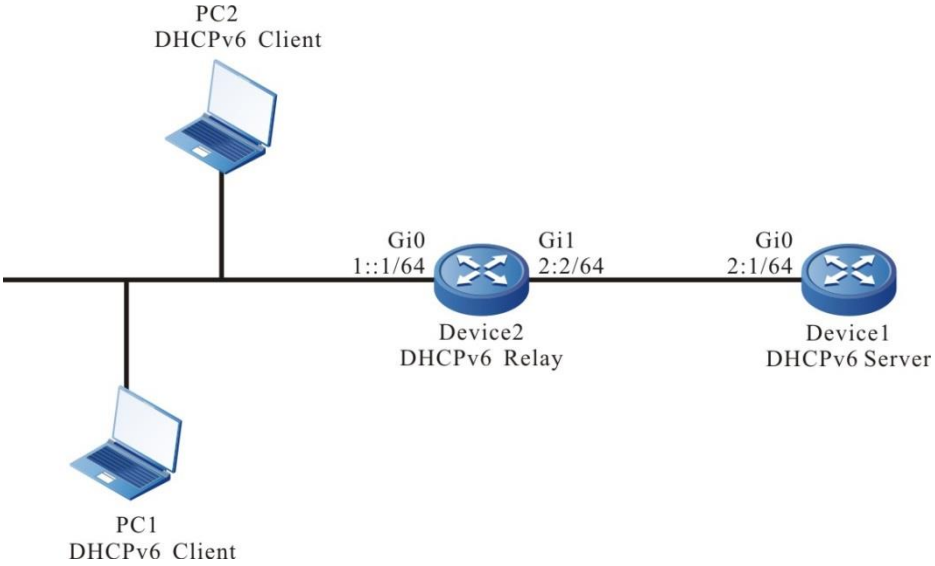



Figure 15-3 Networking for Configuring a DHCPv6 Relay

## Configuration Steps

**Step 1:** Configure the IPv6 address of the interface (omitted).

**Step 2:** Configure an IPv6 address pool for Device 1, and configure the reserved IPv6 addresses.

#Configure IPv6 addresses which are from 1::0 to 1::9 not to be allocated.

```
Device1(config)#ipv6 dhcp excluded-address 1::0 1::9
```

#Configure IPv6 address pool dynamic-pool for Device1.

```
Device1(config)#ipv6 dhcp pool dynamic-pool
Device1(dhcp6-config)#network 1::/64
Device1(dhcp6-config)#lease preferred-lifetime 300 valid-lifetime 600
Device1(dhcp6-config)#exit
```

#Configure a static route to network segment 1::/64.

```
Device1(config)#ipv6 route 1::0/64 2::2
```

**Step 3:** On the interface gigabitethernet0 of Device2, enable the DHCPv6 relay and configure the address of the DHCPv6 server 2::1.

```
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#ipv6 dhcp relay
Device2(config-if-gigabitethernet0)#ipv6 dhcp relay server-address 2::1
Device2(config-if-gigabitethernet0)#exit
```

**Step 4:** Check the result.

#On Device1, query the IPv6 addresses that have been allocated.

```
Device1#show ipv6 dhcp pool dynamic-pool lease
IPv6 Address              Duid                                 Iaid     Type    Time Left(s)
------------ ------------------------------------ -------- ------- ------
------
1::0         000200001613303030313761636635646634      00000000  Lease     574
```

Use the **show ipv6 dhcp pool dynamic-pool lease** command to query the IPv6 addresses that have been allocated to clients. The result shows that a client has obtained the IPv6 address 1::0.

# 16. DNS

## 16.1. Overview

Domain Name System (DNS) is a distributed database that maps domain names and IP addresses. It provides conversion between domain names and IP addresses. With the use of DNS, when users access the Internet, they can use easy-to-memory and meaningful domain names. Then the domain name server in the network resolves the domain names into correct IP addresses. DNS is categorized into static DNS and dynamic DNS.

Static domain name resolution is conducted through a static DNS table. In the static DNS table, domain names and IP addresses are mapped, and some frequently used domain names are added. When a client requests for the IP address of a domain name, the DNS server first searches static DNS table for the corresponding IP address. This improves the efficiency of domain name resolution.

Dynamic domain name resolution is implemented by querying the DNS. A DNS client sends a domain name resolution request to a DNS server. After the DNS server receives the domain name resolution request, it first determines whether the requested domain name is located in its authorized management sub-domain. If yes, it searches the database for the required IP address and then sends the query result to the client. If the domain name is not in the authorized management sub-domain, the DNS server starts a recursive resolution with other DNS server, and then it sends the resolution result to the client. Alternatively, it specifies the address of the next DNS server in the response packet to the DNS client. Then, the client sends another domain name resolution request to the domain name server. This is so called iterative resolution mode.

## 16.2. DNS Function Configuration

Table 16-1 DNS function list

| **Configuration Tasks** | |
|---|---|
| Configure the DNS cache specification | Configure the maximum specification of the static cache |
| | Configure the maximum specification of the dynamic cache |
| Configure the DNS client function. | Configure static domain name resolution. |
| | Configure dynamic domain name resolution. |
| Configure the DNS proxy function | Configure the DNS proxy |

| Configuration Tasks | |
|---|---|
| Configure the DNS detection function | Configure the domain name list |
| | Detect the domain name resolution |
| Configure the DNS transparent proxy function | Configure DNS front transparent proxy |
| | Configure the DNS link transparent proxy |
| Configure the DNS64 function | Configure DNS64 |

## 16.2.1. Configure DNS Cache Specification

### Configuration Condition

None

### Configure DNS Specification

Modify the maximum specification supported by DNS. If the current specification is M, the current number is n; the configured specification is N; There are the following scenarios:

1. Static specification, if N > M or n < N < M, the configuration takes effect immediately; If N < n, the configuration failed
2. Dynamic specifications, if N > M or n < N < M, the configuration takes effect immediately; If N < n, the configuration takes effect, waiting for dynamic number aging.

Table 16-2 Configure the list of privileged mode authentication methods

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure the maximum specification supported by static DNS cache | **dns static max-count** *number* | Optional<br>By default, the static cache supports 64 at most. |

| Step | Command | Description |
|---|---|---|
| Configure the maximum specification supported by dynamic DNS cache | **dns dynamic max-count** *number* | Optional<br>By default, the dynamic cache supports 10K at most. |

## 16.2.2. Configure the DNS Client Function

### Configuration Condition

None

### Configure Static Domain Name Resolution

In configuring static domain name resolution, you can configure a domain names to map an IPv4 address.

Table 16-3 Configuring static domain name resolution

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure a domain name to map an IPv4 address. | **ip host** [ **vrf** *vrf-name* ] *domain-name ipv4-address* | Mandatory.<br>By default, no domain name and its corresponding IPv4 address is configured. |
| Configure the corresponding IPv6 address of the static domain name | **IPv6 host** [ **vrf** *vrf-name* ] *domain-name IPv6-address* | Mandatory<br>By default, do not configure the domain name or the corresponding IPv6 address. |

### Configure Dynamic Domain Name Resolution

In configuring dynamic domain name resolution, you need to configure the IP address of a domain name server. Then, domain resolution requests can be sent to the proper domain server for resolution.

Users can pre-configure a domain suffix. Then, when the users use a domain name, they can input only part fields of the domain name, and the system automatically adds pre-configured domain suffix for resolution.

Table 16-4 Configuring dynamic domain name resolution

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Configure a domain suffix. | **ip domain-name** [ **vrf** *vrf-name* ] *domain-name* | Mandatory.<br>By default, no domain suffix is configured. |
| Configure a DNS server address. | **ip name-server** [ **vrf** *vrf-name* ] *ip-address* | Mandatory.<br>By default, the DNS server address is not configured. |

### 16.2.3. **Configure DNS Proxy Function**

#### Configuration Condition

None

#### Configure DNS Proxy

The DNS proxy does not have the DNS resolution function. It just forwards the resolution request from the DNS client to the DNS server, and forwards the resolution result from the DNS server to the DNS client.

Table 16-5 Configure the DNS proxy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode. | **configure terminal** | - |
| Enable the DNS proxy function | **dns proxy enable** [ **vrf** *vrf-name* ] | Mandatory<br>By default, do not enable the DNS proxy function. |
| Configure the DNS proxy cache | **dns proxy cache** [ **vrf** *vrf-name* ] | Optional<br>By default, do not configure the DNS proxy cache function. |

| Step | Command | Description |
|---|---|---|
| Configure the life time of the DNS proxy cache entry | **dns proxy ttl** *ttl-value* [ **vrf** *vrf-name* ] | Optional<br>By default, the life time of the DNS proxy cache entry is 1800s. |

## 16.2.4. Configure the DNS Detection Function

### Configuration Condition

None

### Configure the Domain List

You can add some common domain names to the domain name list by configuring the domain name list. When necessary, directly specify the index of the domain name list.

Table 16-6 Configure the domain name list

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the domain name list and enter the domain name list configuration mode | **dns domain-list** *list-name* | Mandatory<br>By default, do not configure the domain name list. |
| Configure the domain name | **domain** *domain-name* | Mandatory<br>By default, the domain name is not configured in the domain name list. |

### Detect the Domain Name Resolution

You can detect whether the DNS server can resolve the specified domain name correctly by detecting the domain name resolution.

Table 16-7 Detect the domain name resolution

| Step | Command | Description |
|---|---|---|
| Detect the domain name resolution | **dns-query** [ **vrf** *vrf-name* ] *ip-address* [ **name** *domain-name* **\| name-list** *index* ] [ **count** *packet-num* ] [ **timeout** *time* ] | Mandatory<br>By default, do not detect the domain name resolution. |

## 16.2.5. **Configure DNS Transparent Proxy Function**

DNS transparent proxy function can modify the destination address of some DNS request packets to the DNS server address of other ISPs (such as the DNS server address of ISP2). DNS requests are forwarded to different ISPs, and the resolved server address belongs to different ISPs, so the Internet traffic will be forwarded through different ISP links. In this way, one link will not be congested while other links will be idle, and all link resources will be fully utilized.

### Configuration Condition

None

### Configure DNS Transparent Proxy

Tabke 16-8 Configure the DNS transparent proxy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the DNS transparent proxy | **dns transparent-proxy enable** | Mandatory<br>By default, do not enable the DNS transparent proxy. |
| When the configured transparent proxy rules of front domain name fails, continue to perform the function of link transparent proxy | **dns transparent-proxy control enable** | Optional<br>By default, when the pre transparent proxy fails, do not continue to perform the function of link transparent proxy. |

### Configure DNS Pre Transprent Proxy

After the domain name under the specified domain name list (refer to the configuration of the domain name list in the DNS detection function node) does not act as the DNS transparent proxy, even if the DNS server set by the client needs to act as the DNS transparent proxy, the router will not process the DNS request packet for accessing the domain name, but directly route and forward it.

If the preferred DNS address is specified for the domain name that does not act as DNS proxy (**server preferred preferred-dns-address**), the device will request the IP corresponding to all the domain names in the domain name list from the preferred DNS server in advance and cache them. The DNS requests corresponding to these domain names will first check the cache entries. If the cache entries exist, they will be sent to the DNS client through DNS response according to the contents of the cache entries; if the cache entries do not exist, forward the DNS request to the DNS server set by the client.

If both the preferred DNS server address and the alternate DNS address are specified (server preferred preferred-dns-address alternate alternate-dns-address), the device will first query the preferred DNS server for the IP addresses corresponding to all domain names in the domain name list. When the preferred DNS server does not respond, query the standby DNS server. The DNS request processing flow corresponding to these domain names is consistent with that when only the preferred DNS server is used.

Note that domain name cache table entries only support 'A' record cache.

Table 16-9 Configure the DNS pre transparent proxy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the DNS transparent proxy function of the domain name in the specified domain name list | **dns transparent-proxy exclude domain-list** *list-name* **[ server primary** *preferred-dns-address* **[ secondary** *alternate-dns-address* **] ]** | Mandatory<br>By default, do not perform the DNS transparent proxy for the domain name. |

### Configure DNS Link Tranparent Proxy

When the IP addresses of the preferred DNS server and the standby DNS server are bound at the same time, the DNS transparent proxy function will use the IP address of the preferred DNS server to replace the destination address of the DNS request packet, and then continue to forward the packet. When the state of the preferred DNS server is down, the DNS transparent proxy function will replace the destination address of the DNS request packet with the IP address of the standby DNS server.

Table 16-10 Configure the DNS link transparent proxy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the DNS server address of the link transparent proxy | **dns transparent-proxy server** *server-address* | Mandatory<br>By default, do not perform the link transparent proxy for any DNS server. |
| Configure the corresponding proxy DNS server address of the bound egress interface | **dns server bind interface** *interface-name* **primary** *primary-dns-address* **[ secondary** *secondary-dns-address* **]** | Mandatory<br>By default, do not configure the corresponding proxy DNS server address of the bound egress interface. |

## 16.2.6. **Configure DNS64 Function**

Synthesize the A record (IPv4 address) in DNS query information into the AAAA record (IPv6 address), and return the synthesized AAAA record to IPv6 users.

### Configuration Condition

None

### Configure DNS64

Table 16-11Configure DNS64

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the dns64 function | **dns64 enable [vrf** *vrf-name*] | Mandatory<br>By default, do not enable the dns64 function. |

| Step | Command | Description |
|---|---|---|
| Enable the dns cache function | **dns proxy cache** [ **vrf** *vrf-name* ] | Mandatory<br>By default, do not enable the dns cache function. |
| Configure the dns64 prefix | **dns64 prefix** [ **vrf** *vrf-name* ] *IPv6-prefix* | Optional<br>By default, the dns64 prefix is "64:FF9B::/96". |

### DNS Monitoring and Maintaining

Table 16-12 DNS monitoring and maintaining

| Command | Description |
|---|---|
| **enable** | Privileged mode |
| **debug dns {all \| config \| event \| forwarding \| mpos \| packet \| timer}** | Open the DNS debug information |
| **show dns domain-list** [*list-name*] | Display the domain name list |
| **show dns proxy client** | Displays the DNS client information on the DNS proxy that did not receive the resolution result |
| **show dns proxy config** | Display the DNS proxy configuration information |
| **show hosts** | Display the domain name resolution table entries |
| **show name-server [vrf** *vrf-name*] | Display the DNS server information |

## 16.3. DNS Typical Configuration Example

### 16.3.1. Configure Static Domain Name Resolution

#### Network Requirements

- Device and PC are interconnected, and the route is reachable.

- The host name of PC is host.xxyyzz.com, and the IP address is 1.0.0.2/24.
- On Device, access the host host.xxyyzz.com through static domain name resolution.

### Network Topology

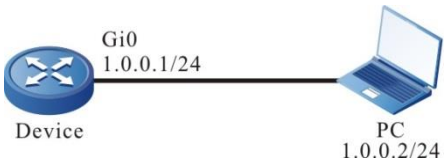



Figure 16-1 Networking for Configure Static Domain Name Resolution

### Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure a static domain name.

#On Device, configure the host name host.xxyyzz.com to correspond to IP address 1.0.0.2.

```
Device#configure terminal
Device(config)#ip host host.xxyyzz.com 1.0.0.2
```

**Step 3:** Check the result.

#On Device, ping host host.xxyyzz.com. Device obtains the IP address 1.0.0.2 that corresponds to the host name through local domain name resolution.

```
Device#ping host.xxyyzz.com
%Bad IPv6 address or unknown hostname!
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/6/16 ms.
```

**Note:**

- In pinging a host name, the IPv6 address corresponding to the host name is first resolved, and then the IPv4 address.

## 16.3.2. Configure Dynamic Domain Name Resolution

### Network Requirements

- The IP address of the DNS server is 1.0.0.3/24, the IP address of Device is 1.0.0.1/24, and the IP address of PC is 1.0.0.2/24.
- The DNS server, Device, and PC are interconnected through a LAN, and the route is reachable. On the DNS server, the DNS record of host.xxyyzz.com and 1.0.0.2 exists.
- Device access PC through dynamic resolution of host.xxyyzz.com through the DNS server.

### Network Topology

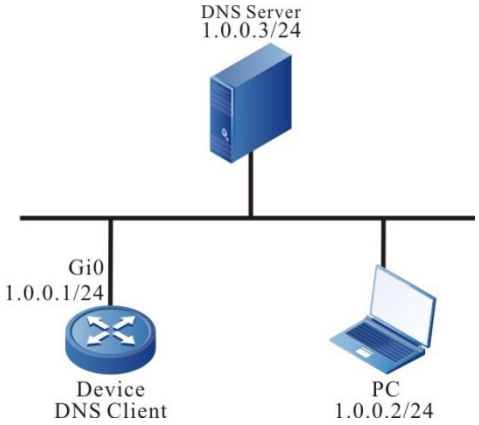



Figure 16-2 Networking for configure dynamic domain name resolution

### Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the DNS server.(Omitted)

**Step 3:** Configure the DNS client.

#Specify a DNS server for the client, and the IP address is 1.0.0.3.

```
Device#configure terminal
Device(config)#ip name-server 1.0.0.3
```

**Step 4:** Check the result.

#On Device, ping host host.xxyyzz.com. Device obtains the IP address 1.0.0.2 that corresponds to the host name through the DNS server.

```
Device#ping host.xxyyzz.com
%Bad IPv6 address or unknown hostname!

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/6/16 ms.
```

## 16.3.3. Configure DNS Proxy

### Network Requirements

- The IP address of DNS server is 1.0.0.2/24, and the IP address of PC is 100.0.0.2/24.
- Set the DNS records of host.xxyyzz.com and 100.0.0.1 on the DNS server.
- The DNS server setting of PC is 100.0.0.1.
- Device enables DNS proxy function.

## Network Topology

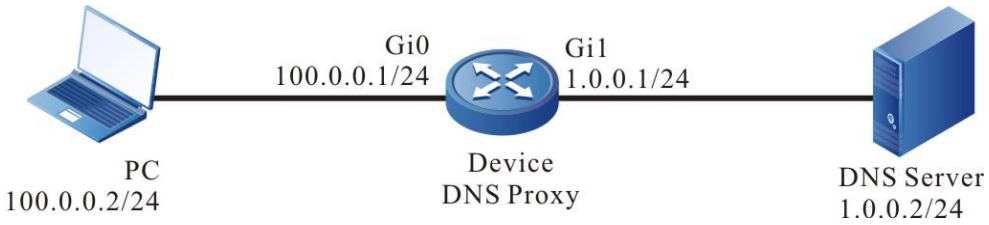



Figure 16-3 Networking of configuring the DNS proxy

## Configuration Steps

**Step 1:** Configure IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the DNS server.(Omitted)

**Step 3:** Configure the DNS server address of PC (omitted).

**Step 4:** Configure the DNS proxy.

#Specify the DNS server for Devie and the IP address is 1.0.0.2.

```
Device#configure terminal
Device(config)#ip name-server 1.0.0.2
```

#Enable the DNS proxy and cache function for Device.

```
Device#configure terminal
Device(config)#dns proxy enable
Device(config)#dns proxy cache
```

**Step 5:** Check the result.

#On Device, view whether the DNS proxy function is enabled.

```
Device#show dns proxy config
VRF name    Proxy    Cache    TTL
--------    -----    -----    -----
global      On       On       1800(s)
```

#You can ping the host name host.xxyyzz.com on PC. Check the DNS proxy cache on the device. There is a dynamic DNS cache with the address of 100.0.0.1 corresponding to the domain name host.xxyyzz.com.

```
Device #show hosts
Static: 0  Dynamic: 1

     hostname            inet address/inet6 address       type   remain(s)   vrf
--------------------------  ------------------------------------  ------ ----
-----   ---
Host.xxyyzz.com          100.0.0.1                    dynamic  3595        global
```

**Note:**

- After enabling the DNS proxy function, listen to the packet of port 53. Forward the DNS request packet to the DNS server. At the same time, it also forwards DNS response packet to the client.
- When the DNS cache is enabled, it can cache the domain IP obtained from the DNS server. When the cache is not enabled, only the DNS packet can be transmitted transparently, but cannot cache the domain IP.

## 16.3.4. Configure DNS Transparent Proxy

### Network Requirements

- The IP address of DNS server is 2.0.0.2/24, that of server is 2.0.0.3/24, and that of PC is 1.0.0.1/24.
- DNS server, PC and server are interconnected through device, and the route is reachable bi-directionally. The DNS records of host.xxyyzz.com and 2.0.0.3 are set on DNS server.
- PC gateway is set to 1.0.0.2, DNS address is set to 2.0.0.1.
- Device enables DNS transparent proxy function.
- PC dynamically resolves host.xxyyzz.com to access the server through the transparent proxy function of Device.

### Network Topology

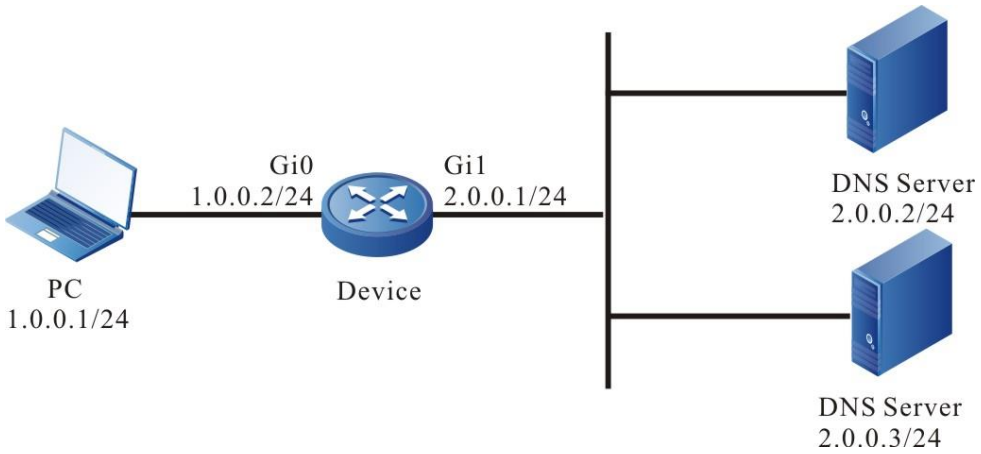



Figure 16-4 Networking of configuring DNS transparent proxy

### Configuration Steps

**Step 1:** Configure the IP address and default gateway of the interface (omitted).

**Step 2:** Configure the DNS server (omitted).

**Step 3:** Configure the DNS server address of PC (omitted).

**Step 4:** Configure the DNS transparent proxy.

#Enable the DNS transparent proxy function.

```
Device#configure terminal
```

```
Device(config)#dns transparent-proxy enable
```

#Configure the DNS domain name list.

```
Device(config)#dns domain-list 1
Device(config-domain-list)#domain host.xxyyzz.com
```

#Configure the DNS transparent proxy rules.

```
Device(config)#dns transparent-proxy exclude domain-list 1 server primary 2.0.0.2
```

**Step 5:** Check the result.

#View the transparent proxy cache entries of Device.

```
Device#show dns transparent-proxy exclude
Domain Name                Type      Ip-Address     Ttl
----------------           -----     ---------      ---
host.xxyyzz.com            dynamic   2.0.0.3        180
```

#View the transparent proxy cache entries on Device.

```
Device#show dns transparent-proxy exclude
Domain Name                                Ip-Address
----------------                           ----------
host.xxyyzz.com                            2.0.0.3
```

#On the PC, ping the host name host.xxyyzz.com, and the ping can be connected.

```
C:\>ping host.xxyyzz.com

Pinging host.xxyyzz.com [2.0.0.3] with 32 bytes of data:
Reply from 2.0.0.3: bytes=32 time<1ms TTL=255
Reply from 2.0.0.3: bytes=32 time<1ms TTL=255
Reply from 2.0.0.3: bytes=32 time<1ms TTL=255
Reply from 2.0.0.3: bytes=32 time<1ms TTL=255

Ping statistics for 2.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

**Note:**

- After the transparent proxy function is enabled, identify the packet with port number 53. After configuring the transparent proxy rule, the device searches the domain name according to the domain name rule, and DNS constructs the response packet to the client according to the search result of the domain name.
- When DNS transparent proxy, DNS link transparent proxy and DNS proxy functions are enabled at the same time, and dns transparent-proxy control enable is enabled, the

priority order of the three functions is DNS transparent proxy > DNS link transparent proxy > DNS proxy.

## 16.3.5. Configure DNS Link Transparent Proxy

### Network Requirements

- The IP address of DNS Server1 is 2.0.0.2/24, the IP address of DNS server2 is 3.0.0.2/24, the IP address of PC1 is 1.0.0.1/24, and the IP address of PC2 is 1.0.0.2/24.
- The DNS server and PC are connected by Device, and the route is reachable bi-directionally. The DNS records of host.xyyzz.com and 2.0.0.2 are set on DNS Server1, and the DNS records of host.xyyzz.com and 3.0.0.2 are set on DNS server2.
- Both DNS server and PC use Device as gateway router, and the DNS address of PC1 is set to 4.0.0.1, and that of PC2 is set to 5.0.0.1.
- Device enables the DNS link transparent proxy function and sets the route to 4.0.0.0/24 and 5.0.0.0/24 network segments.
- PC1 and PC2 dynamically analyze host.xyzz.com through the link proxy function of Device.

### Network Topology

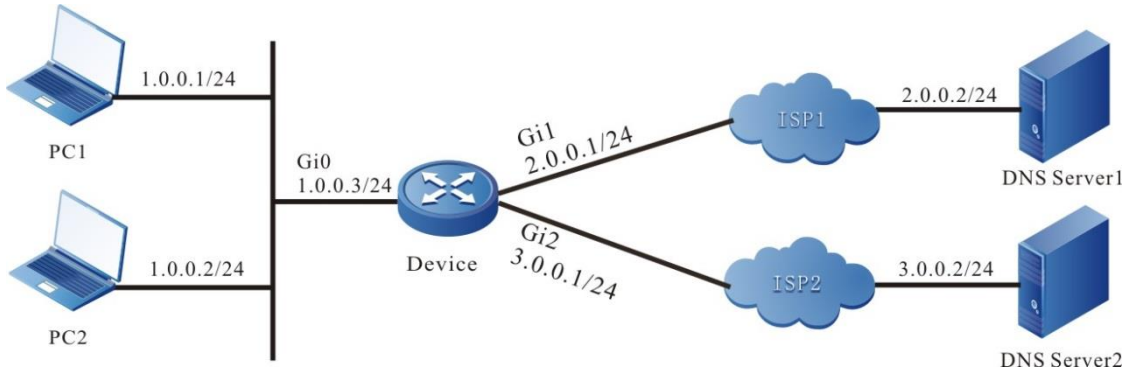



Figure 16-5 Networking of configuring DNS link transparent proxy

### Configuration Steps

**Step 1:** Configure the IP address and default gateway of the interface (omitted).

**Step 2:** Configure the DNS server (omitted).

**Step 3:** Configure the DNS server address of PC1 and PC2 (omitted).

**Step 4:** Configure the DNS link transparent proxy.

#Enable the DNS transparent proxy function.

```
Device#configure terminal
Device(config)#dns transparent-proxy enable
```

#Configure the monitoring address, out interface and proxy rules of the DNS transparent proxy.

```
Device(config)#dns transparent-proxy server 4.0.0.1
Device(config)#dns transparent-proxy server 5.0.0.1
```

```
Device(config)#dns server bind interface gigabitethernet1 primary 2.0.0.2
Device(config)#dns server bind interface gigabitethernet2 primary 3.0.0.2
```

# Configure the route.

```
Device(config)#ip route 4.0.0.0 255.255.255.0 gigabitethernet1
Device(config)#ip route 5.0.0.0 255.255.255.0 gigabitethernet2
```

**Step 5:** Check the result.

#On PC1, ping the host name host.xxyyzz.com, and the ping can be connected.

```
C:\>ping host.xxyyzz.com

Pinging host.xxyyzz.com [2.0.0.2] with 32 bytes of data:
Reply from 2.0.0.2: bytes=32 time<1ms TTL=255
Reply from 2.0.0.2: bytes=32 time<1ms TTL=255
Reply from 2.0.0.2: bytes=32 time<1ms TTL=255
Reply from 2.0.0.2: bytes=32 time<1ms TTL=255

Ping statistics for 2.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On PC2, ping the host name host.xxyyzz.com, and the ping can be connected.

```
C:\>ping host.xxyyzz.com

Pinging host.xxyyzz.com [3.0.0.2] with 32 bytes of data:
Reply from 3.0.0.2: bytes=32 time<1ms TTL=255
Reply from 3.0.0.2: bytes=32 time<1ms TTL=255
Reply from 3.0.0.2: bytes=32 time<1ms TTL=255
Reply from 3.0.0.2: bytes=32 time<1ms TTL=255

Ping statistics for 3.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

**Note:**

- After the device service is identified, DNS queries the route of the forwarding packet and records the information of the request packet according to the query result and the link transparent proxy, modifies the destination address of the packet and sends it to

the server according to the link transparent proxy rules bound to the out interface of the route; for the response packet, the device is identified as a DNS packet, and the DNS constructs the response packet to the client according to the recorded request packet information.

## 16.3.6. Configure NAT64-Based DNS64

### Network Requirements

- The IP address of the DNS server is 2.0.0.2/24, that of server is 2.0.0.3/24, and that of PC is 1:: 1/96.
- DNS server, Server and PC are interconnected through Device, and Device is used as gateway router.
- The DNS address of the PC is set to 1::2, and the DNS records of host.xxyyzz.com and 2.0.0.3 are set on the DNS server.
- Device enables the DNS64 function and NAT64 function.
- PC dynamically resolves host.xxyyzz.com to access server through the DNS64 function of Device.

### Network Topology

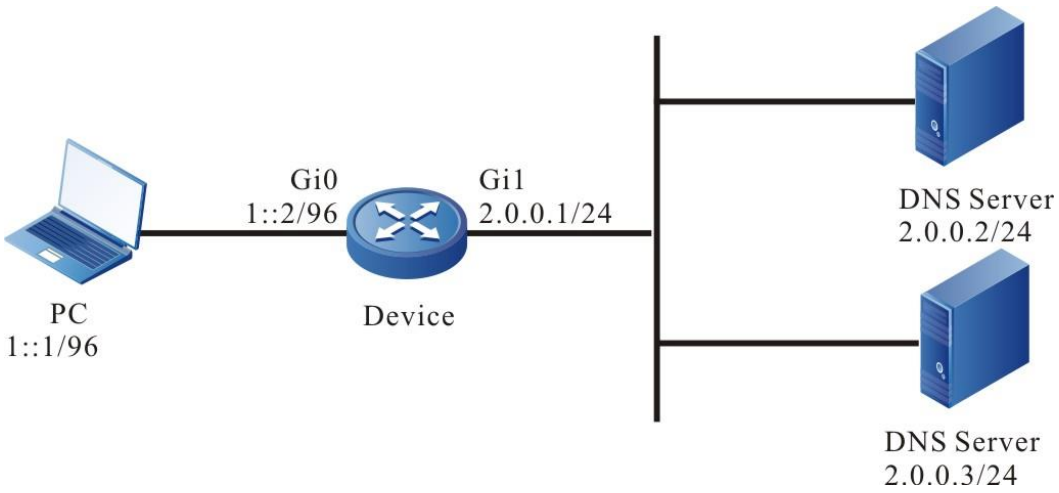



Figure 16-6 Networking of configuring DNS64

### Configuration Steps

**Step 1:** Configure the IP address and default gateway of the interface (omitted).

**Step 2:** Configure the DNS server (omitted).

**Step 3:** Configure the DNS server address of PC (omitted).

**Step 4:** Enable the DNS proxy function.

```
Device#configure terminal
Device(config)#ip name-server 2.0.0.2
Device(config)#dns proxy enable
Device(config)#dns proxy cache
```

**Step 5:** Enable the DNS64 function.

```
Device(config)#dns64 enable
Device(config)#dns64 prefix 1:2:3::/96
```

**Step 6:** Enable the NAT64 function and configure the nat64 translation rules.

```
Device(config)#interface gigabitethernet0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#exit
Device (config)#IPv6 access-list extended 7001
Device (config-v6-list)#permit IPv6 any 1:2:3::/96
Device (config-v6-list)#exit
Device (config)# nat64 prefix 1:2:3::/96
Device (config)#nat64 v6v4 list 7001 interface gigabitethernet1 overload
```

**Step 7:** Check the result.

#On PC, ping the host name **host.xxyyzz.com**, and the ping can be connected.

```
C:\>ping host.xxyyzz.com

Pinging host.xxyyzz.com [1:2:3::200:3] with 32 bytes of data:
Reply from 1:2:3::200:3: time<1ms
Reply from 1:2:3::200:3: time<1ms
Reply from 1:2:3::200:3: time<1ms
Reply from 1:2:3::200:3: time<1ms

Ping statistics for 1:2:3::200:3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On Device, view the dynamic domain name information.

```
Device#show hosts
Static: 0  Dynamic: 1

        hostname              inet address/inet6 address       type   remain(s)   vrf
```

```
-------------------------  ----------------------------------  ------ ----
-----  ---
host.xxyyzz.com          2.0.0.3                    dynamic 3384       global
```

#On Device, view the NAT64 translation information.

```
Device# show nat64 translation
      Proto   IPv6 source:port                    IPv4 source:port        lifetime
      IPv6 destination:port                  IPv4 destination:port
      -----  --------------------                 --------------------    ---
-----
      ICMP    [1::1]:1                           2.0.0.1:10001          59
      [1:2:3::200:3]:1                    2.0.0. 3:10001

      Total:1
```

**Note:**

- After the NAT64 function is enabled, if the device receives the AAAA request, it will send the AAAA request to the DNS server. If the request times out or the domain name resolution fails, it will send the AAAA request to the server, and splice the IPv4 address in the response packet into the IPv6 address through the DNS64 prefix, and return it to the client.
- According to the application scenarios of DNS, DNS64 is divided into client DNS64 and proxy DNS64.

# 17. L2TP

## 17.1. Overview

VPDN (Virtual Private Dial-up Network) provides connection services for the remote users expecting to connect to the enterprise network through the ISP (Internet Service Provider). The VPDN allows independent protocol domain to enjoy the common visit infrastructure, such as the modem, visiting server, and the ISDN router. The enterprise can purchase or rent the remote access infrastructure from the ISP to enable its remote work team to visit the intranet.

The VPDN uses the Layer 2 Tunneling Protocol, such as Layer 2 Forwarding and Point-to-Point Tunneling Protocol, to enable the ISP or other visiting servers to create a virtual channel to connect the customers' remote sites or remote users of the with the main network of the enterprise.

The network visiting server of the ISP can be dialed by remote users. Frames without any links and transparent bytes are encapsulated using the preceding tunnel protocol, such as the L2TP and are forwarded through the corresponding channels. The main gateway of the customer (enterprise network) receives these L2TP frames, decapsulates the L2TP links, and processes these entered frames for the corresponding interfaces.

L2TP (Layer 2 Tunneling Protocol) is one of the VPDN technologies. The L2TP allows users to dial in the ISP and allows the dialed-in users to incorporate the connections into the enterprise network. This enables the enterprise network to allow the ISP to keep its point at the logical attribute for all the dial-in connections through all logical attribute, such as authenticating the network layer attribute.

The L2TP uses the following two VPDN peer devices to construct a L2TP tunnel: LAC (L2TP Access Concentrator) and LNS (L2TP Network Server). When the user dials in the LAC, the LAC initiates the L2TP tunnel connection with the LNS. The LNS locates at the edge of the enterprise network or behind the firewall of the internal enterprise network.

Before all the sessions between the LAC and LNS are established, the control connection message must be used to establish the tunnel. This tunnel can send the PPP grouping from the LAC to the LNS. Establishing the control connection includes confirming the peer device, L2TP version, sub frame, and channel carrier.

The L2TP uses three message exchanges to establish the control connection. The following figure shows the process of establishing a tunnel.

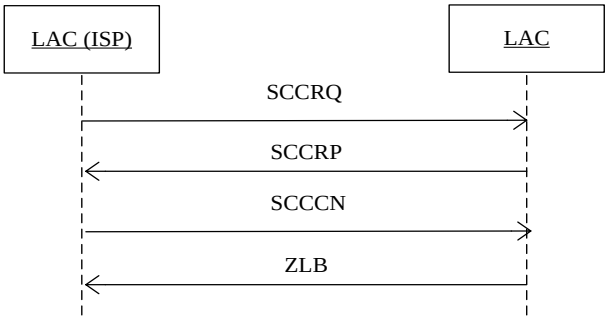



Figure 17-1 Establish the L2TP

1. The LAC sends the SCCRQ (Start Control Connection Request) to the LNS.
2. The LNS sends the SCCRP (Start Control Connection Reply) to the LAC.

3. The LAC sends the SCCCN (Start Control Connection Connected) to the LNS.
4. The LNS sends the ZLB (Zero-Length Body) to the LAC.

After the tunnel is established successfully, an independent L2TP session will be established. The following figure shows the process of creating a session:

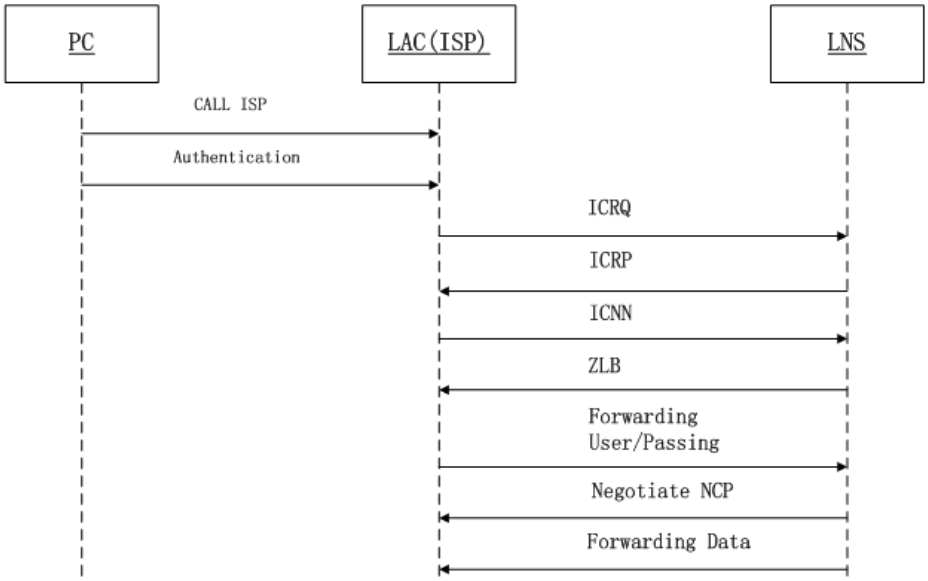



Figure 17-2 Establish the L2TP session

1. The host initiates the PPP connection with the LAC through the standard analog phone or the ISDN. The LAC receives the connection and establishes the data link layer.
2. The LAC checks part hosts, that is, checking whether the user is a VPDN client through the user name. If the user is not a VPDN user, check the user using the user name database provided by the local ISP for other connections. If the identified user name is a VPDN user, the L2TP session will be established subsequently.
3. The LAC initiates the L2TP session containing the ICRQ (Incoming Call Request) message.
4. The LNS receives the L2TP session containing the ICRP (Incoming Call Reply) message.
5. The LAC sends the L2TP session established by the ICCN (Incoming Call Connected).
6. The LNS replies to confirm the ZLB (Zero-Length Body) message.
7. The LAC forwards the user name and password to the LNS to complete the identification process.
8. The LAC forwards all the LCP (Link Control Protocol) negotiation options to the LNS. The LNS will establish a virtual access interface, which is copied from a virtual template. At this time, the user completes the identification of the LNS and all the other LCP negotiation options. If the user cannot perform the identification, the LNS will send a disconnection message to the LAC.
9. The data can be forwarded between the LAC and LNS through the L2TP tunnel.

## 17.2. L2TP Function Configuration

Table 17-1 The L2TP function list

| Configuration Task | |
|---|---|
| Configure the VPDN group | Enable the VPDN |
| | Configure the VPDN group of the LAC |
| | Configure the VPDN group of the LNS |
| Configure the VPDN authentication | Configure the VPDN authentication |
| Configure the VPDN group parameter | Configure the keep-alive time of the VPDN tunnel |
| | Configure the receive window size of the VPDN tunnel |
| Configure the spontaneous tunnel | Configure the L2TP spontaneous tunnel |

### 17.2.1. **Configure the VPDN group**

#### Configuration Condition

None

#### Enable the VPDN

To configure any VPDN technology, enable the VPDN at first. Then other VPDN commands can be used.

Table 17-2 Enable the VPDN

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the VPDN | **vpdn enable** | Mandatory<br>By default, the VPDN function is not enabled. |

### Configure VPDN Group of LAC

The VPDN group is a mechanism, organizing all the VPDN commands related to all the VPDN peer devices into an independent structure. This mechanism specifies the LAC request dial-in or the LNS dial-in. once the VPDN group is configured as a specific L2TP device, you cannot change the configuration.

If the VPDN group is configured as the request dial-in mode, the VPDN group is identified as the LAC L2TP device.

Table 17-3 Configure the VPDN group of the LAC

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the VPDN | **vpdn enable** | Mandatory<br>By default, the VPDN function is not enabled. |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not created. |
| Configure permitting the VPDN request dial-in and enter the VPDN request dial-in configuration mode | **request-dialin** | Mandatory<br>By default, the request dial-in is not configured for the VPDN group.<br>The VPDN group is identifies as the LAC. |

| Step | Command | Description |
|---|---|---|
| Configure the VPDN group application protocol type | **protocol l2tp** | Mandatory<br>By default, the application protocol is not configured for the VPDN group.<br>The application VPDN protocol is specified for the VPDN group. Currently, only the L2TP protocol is supported. |
| Configure the VPDN group domain name | **domain** *domain-name* | Mandatory<br>By default, the domain name is not configured for the VPDN group.<br>When the user dials in the LAC, the LAC needs to send user data to the specified LNS. That is, use the domain name to map the user to the LNS. |
| Return to the VPDN configuration mode | **exit** | - |
| Configure the IPv4 destination IP address of the L2TP tunnel | **initiate-to ip** *ip-address* | Mandatory for IPv4 tunnel<br>By default, the IP v4 destination address of the L2TP tunnel is not configured for the VPDN group. |
| Configure the IPV6 destination address of the L2TP tunnel | **initiate-to ipv6** *ipv6-address* | Mandatory for IPv6 tunnel<br>By default, the VPDN group does not configure the IPv6 destination address of the L2TP tunnel. |

| Step | Command | Description |
|---|---|---|
| Configure the LAC name | **local name** *host-name* | Optional<br>By default, the LAC name is not configured. |

**Configure the VPDN group of the LNS**

If the VPDN group is configured as the accept dial-in mode, the VPDN group is identified as the LNS L2TP device.

Table 17-4 Configure the VPDN group of the LNS

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enable the VPDN | **vpdn enable** | Mandatory<br>By default, the VPDN function is not enabled. |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not configured. |
| Configure permitting VPDN accept dial-in and enter the VPDN accept dial-in configuration mode | **accept-dialin** | Mandatory<br>By default, the accept dial-in is not configured for the VPDN group.<br>The VPDN group is identified as the LNS. |

| Step | Command | Description |
|---|---|---|
| Configure the VPDN group application protocol type | **protocol l2tp** | Mandatory<br>By default, the VPDN application protocol is not configured for the VPDN group.<br>The application VPDN protocol is specified in the VPDN group. Currently, only the L2TP protocol is supported. |
| Configure the VPDN group virtual template interface | **virtual-temple** *virtual-temple-number* | Mandatory<br>By default, the virtual template interface is not configured for the VPDN group.<br>The L2TP packet is the PPP packet with extra data header. If the data header is removed, the PPP packet must take effect. Therefore, an interface can parse the PPP data is required, that is, the virtual interface. |
| Return to the VPDN configuration mode | **exit** | - |
| Configure the LNS peer end name | **terminate-from hostname** *lac-hostname* | Optional<br>By default, the LNS peer device name is not configured for the VPDN group. |
| Configure the LNS name | **local name** *host-name* | Optional<br>By default, the LNS name is not configured. |

## 17.2.2. Configure VPDN Authentication

### Configuration Condition

Before configuring the VPDN authentication function, first complete the following task:

- The corresponding VPDN group is created.

### Configure VPDN Authentication

The L2TP uses a simple and configurable CHAP (Challenge Handshake Authentication Protocol) in the control connection establishment process. If the LAC or the LNS needs to be authenticated, then the AVP (Attribute Value Pairs) in the SCCRQ or SCCRP control packet contains the challenge message. Any party receives such SCCRQ or SCCRP must send the SCCRP or SCCCN control packet to response. If the response does not match the peer device expectation, the tunnel is not allowed to be established correctly.

Table 17-5 Configure the VPDN authentication

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not created. |
| Enable the VPDN authentication function | **l2tp tunnel authentication** | Mandatory<br>By default, the authentication function foe the VPDN is enabled. |
| Configure the VPDN identification password | **l2tp tunnel password** [**0**] *password* | Mandatory<br>By default, the authentication password for the VPDN is not configured. |

## 17.2.3. Configure VPDN Group Parameter

### Configuration Condition

Before configuring the VPDN authentication function, first complete the following task:

- The corresponding VPDN group is created.

### Configure Keep-alive Time of VPDN Tunnel

After the L2TP tunnel is established successfully, to detect whether the peer device can communicate normally, the L2TP device sends the keep-alive packet periodically to confirm the connectivity of the peer device.

Table 17-6 Configure the keep-alive time of the VPDN tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not created. |
| Configure the keep-alive time of the VPDN tunnel | **l2tp tunnel hello** *time-value* | Mandatory<br>By default, the VPDN tunnel keep-alive time is 60s.<br>After the L2TP tunnel is created, the packet is sent periodically to check whether the tunnel can communicate normally. |

### Configure VPDN Tunnel Connection Status Detection

The command is used to configure the packets passing the tunnel to detect the connectivity of the tunnel. Once the local tunnel can receive the data, it is considered that the tunnel can communicate normally and does not actively send the tunnel keepalive packet.

The **no l2tp tunnel predictive** command is used to prohibit the detection of tunnel connectivity through packets.

By default, it can detect the connectivity of the tunnel through packets.

Table 17-7 Configure VPDN tunnel connection status detection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not created. |

| Step | Command | Description |
|---|---|---|
| Configure VPDN tunnel connection status detection | **l2tp tunnel predictive** | Mandatory<br>By default, enable the connection status detection function of the VPDN tunnel. |

### Configure PPP LCP Negotiation Action in VPDN Tunnel Forced Mode

In forced mode, LAC will carry AVP and other information to LNS through ICCN. After analysis, LNS will compare the LCP information carried with the PPP configuration of LNS. If it does not match, it will renegotiate. If it matches, it will directly negotiate the authentication and IPCP stage.

lcp renegotiation always: LCP renegotiation always occurs.

lcp renegotiation on-mismatch: when LCP parameters do not match and are inconsistent, LCP renegotiation is performed.

The **no lcp renegotiation** command is used to cancel the LCP renegotiation function

By default, renegotiation is not performed.

Table 17-8 Configure the PPP LCP negotiation action in the VPDN tunnel forced mode

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not created. |
| Enable the PPP LCP forced re-negotiation in the VPDN tunnel forced mode | **lcp renegotiation always** | Optional<br>By default, PPP LCP forced renegotiation is not enabled in VPDN tunnel forced mode |
| Enable re-negotiation when PPP LCP does not match in the VPDN tunnel forced mode | **lcp renegotiation on-mismatch** | Optional<br>By default, do not enable re-negotiation when PPP LCP does not match in the VPDN tunnel forced mode |

### Configure Receive and Send Window Size of VPDN Tunnel

The control packets in the L2TP tunnel and session establishment of the L2TP use the queue mechanism to ensure the packet sequence. The user can specify the capacity of the maximum cached control packets in the VPDN group.

Table 17-9 Configure the receive and send window size of the VPDN tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the VPDN group and enter the VPDN configuration mode | **vpdn-group** *group-name* | Mandatory<br>By default, the VPDN group is not configured. |
| Configure the receive window size of the VPDN tunnel | **l2tp tunnel receive-window** *window-size* | Mandatory<br>By default, the VPDN tunnel receive window size is 60 packets. |
| Configure the sending window size of the VPDN tunnel | **l2tp tunnel send-window** *window-size* | Mandatory<br>By default, the sending window size of the VPDN tunnel is 60 packets. |

## 17.2.4. Configure Spontaneous Tunnel

The L2TP tunnel establishment is initiated by the PPP dialing client. Generally, the PPP dialing client is separated from the LAC. In the spontaneous L2TP tunnel mode, the PPP dialing client and the LAC are in the same device.

The spontaneous tunnel is enabled in the following three steps:

- Create the pseudo-wire template and specify the data encapsulation type as L2TPv2.
- Create the virtual PPP interface and configure the PPP dialing authentication user name and password.
- Associate the pseudo-wire template to trigger the tunnel establishment on the virtual PPP interface.

### Configuration Condition

None

### Configure the L2TP spontaneous tunnel

Table 17-10 Configure the L2TP spontaneous tunnel

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the pseudo-wire template and enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | Mandatory<br>By default, the pseudo-wire types are not created.<br>For the detailed commands, refer to the L2TPv3 command manual. |
| Configure the encapsulation data type | **encapsulation l2tpv2** | Mandatory<br>By default, the encapsulation type for the pseudo-wire template is L2TPv3.<br>The data type for the encapsulation pseudo-wire template is L2TP. |
| Associate the interface of the source IPv4 address | **ip local interface** *interface-name* | Mandatory for IPv4<br>By default, do not configure the source IPv4 source address of the pseudo wire template. |
| Associate the interface of the source IPv6 address | **ipv6 local interface** *interface-name ipv6-address* | Mandatory for IPv6<br>By default, do not configure the source IPv6 source address of the pseudo wire template. |
| Return to the global configuration mode | **exit** | - |
| Create the virtual PPP dialing interface | **interface virtual-ppp** *unit-number* | Mandatory<br>By default, the virtual PPP interface is not configured. |

| Step | Command | Description |
|---|---|---|
| Configure the encapsulation PPP type of the virtual PPP interface | **encapsulation ppp** | Mandatory<br>By default, the encapsulation type of the virtual PPP interface is PPP. |
| Configure the PPP authentication name and password | **ppp pap sent-username** *username@domain.com* **password 0** *password* | Mandatory<br>By default, the PPP authentication user name and password are not configured for the virtual PPP interface. |
| Configure the IP address parameter | **ip address** { *ip-address* \| **negotiated** } | Mandatory<br>By default, the IP address parameter is not configured for the virtual PPP interface. |
| Configure associating the pseudo wire template | **pseudowire** [*ipv4-address/ IPv6-address] vc-id* **pw-class** *pw-class-name* | Mandatory<br>By default, the virtual PPP interface is not configured to associate the pseudo-wire template. |

### 17.2.5. L2TP Monitoring and Maintaining

Table 17-11 The L2TP monitoring and maintaining

| Command | Description |
|---|---|
| **clear vpdn l2tp** [ **session** *session-id* \| **tunnel** *tunnel-id*] | Clear all sessions and tunnels and the specified tunnels and sessions or sessions corresponding to the user name |
| **show vpdn** [ **detail** [ **tunnel** *tunnel-id* **session** *session-id* ] ] | Display the information related to the VPDN tunnel and session |

## 17.3. L2TP Typical Example Configuration

### 17.3.1. Configure L2TP Forced Mode Combining PPPoE

#### Network Requirements

- Device1 acts as the PPPoE client, Device2 as the PPPoE server and LAC, and Device3 as the LNS.
- The routing between Device2 and Device3 is reachable.
- Device1 dials in Device2 through the PPPoE. The L2TP tunnel connection is triggered to be established between Device2 and Device3, realizing the communication between Device1 and Device3.

#### Network Topology

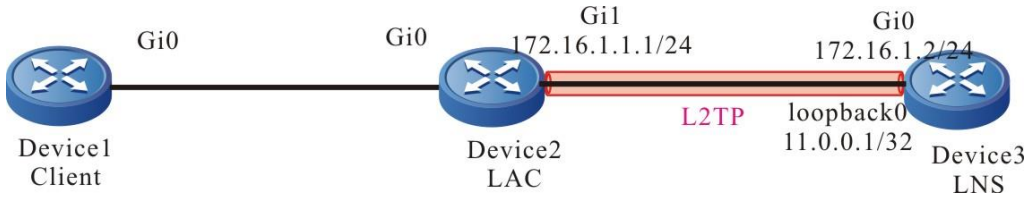



Figure 17-3 Networking of configuring the forced L2TP tunnel mode combining the PPPoE

#### Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure PPPoE client.

#Configure Device1.

Configure the dialing type to be triggered.

```
Device1#configure terminal
Device1(config)#dialer-list 1 protocol ip permit
```

Create the dialer0 interface and configure the dialing-related information.

```
Device1(config)#interface dialer0
Device1(config-if-dialer0)#ip address negotiated
```

```
Device1(config-if-dialer0)#ppp pap sent-username admin@cctv.com password 0 admin
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#dialer-group 1
Device1(config-if-dialer0)#exit
```

Add interface gigabitethernet0 to the dial pool 1.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#exit
```

Add a default routing with the outbound interface as dialer0.

```
Device1(config)#ip route 0.0.0.0 0.0.0.0 dialer0
```

**Step 3:** Configure Device2 as the PPPoE server and the LAC.

#Configure Device2.

Configure the PPPoE server. Create virtual-template0 and encapsulation the PPP protocol Enable the PAP authentication mode.

```
Device2#configure terminal
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#ppp authentication pap
Device2(config-if-virtual-template0)#exit
```

Enable the VPDN. Create the VPDN group PPPoE and configure it as the accept dial-in request mode, using the PPPoE protocol and virtual-template0.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group pppoe
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn-acc-in)#exit
Device2(config-vpdn)#exit
```

Enable the PPPoE server on the gigabitethernet0 interface.

```
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Configure the LAC. Create the VPDN group LAC and configure it as the dial-in request mode. Configure the L2TP protocol and configure the domain name as cctv.com. Specify the IP address of the LNS at the other end of the tunnel as. Configure the local device name as lac and configure the shared key of the LAC and LNS as admin.

```
Device2(config)#vpdn-group lac
Device2(config-vpdn)#request-dialin
Device2(config-vpdn-req-in)#protocol l2tp
Device2(config-vpdn-req-in)#domain cctv.com
Device2(config-vpdn-req-in)#exit
Device2(config-vpdn)#initiate-to ip 172.16.1.2
Device2(config-vpdn)#local name lac
Device2(config-vpdn)#l2tp tunnel password 0 admin
Device2(config-vpdn)#exit
```

**Step 4:** Configure the LNS.

#Configure Device3.

Configure the user name and password.

```
Device3#configure terminal
Device3(config)#local-user admin@cctv.com class network
Device3(config-user-network-admin@cctv.com)#service-type ppp
Device3(config-user-network-admin@cctv.com)#password 0 admin
Device3(config-user-network-admin@cctv.com)#exit
```

Create the local IP address pool, l2tp-pool ranging from 172.16.1.3 to 172.16.1.10.

```
Device3(config)#ip local pool l2tp-pool 172.16.1.3 172.16.1.10
```

Create virtual-template0 and encapsulation PPP protocol. Use the IP address of the interface Loopback0.the IP address is allocated to the PPPoE client from the IP address pool l2tp-pool.

```
Device3(config)#interface virtual-template 0
Device3(config-if-virtual-template0)#encapsulation ppp
Device3(config-if-virtual-template0)#peer default ip address pool l2tp-pool
Device3(config-if-virtual-template0)#ppp authentication pap
Device3(config-if-virtual-template0)#ip unnumbered loopback0
Device3(config-if-virtual-template0)#exit
```

Enable the VPDN. Create the VPDN group lns and configure it as the accept dial-in request mode, using the application L2TP protocol and virtual-template0. Only the L2TP connection request named lac is accepted. Configure the shared key of the LAC and LNS as admin.

```
Device3(config)#vpdn enable
Device3(config)#vpdn-group lns
Device3(config-vpdn)#accept-dialin
```

```
Device3(config-vpdn-acc-in)#protocol l2tp
Device3(config-vpdn-acc-in)#virtual-template 0
Device3(config-vpdn-acc-in)#exit
Device3(config-vpdn)#local name lns
Device3(config-vpdn)#terminate-from hostname lac
Device3(config-vpdn)#l2tp tunnel password 0 admin
Device3(config-vpdn)#exit
```

**Note:**

- The shared key of the LAC and LNS must be consistent. Otherwise, the L2TP tunnel cannot be established successfully.

**Step 5:** Check the result.

#View the PPPoE session information on Device1.

```
Device1#show pppoe session information


PPPoE Session Information:(Max Sessions=1024)
 UID  SID     RemMAC      LocMAC      O-Intf  state  C/S ActiveTime    Local-IP
Peer-IP
----- ----- -------------- -------------- ---------------- ----- --- ---------- -
--------------- ----------------
   5    5 0001.7ade.1819 0001.7adf.8d1b   gigabitethernet0  UP  C  00:00:40   172.16.1.4
11.0.0.1
There are 1 PPPoE sessions up
```

#View the dialer0 interface information on Device1.

```
Device1#show interface dialer0
dialer0:
   line protocol is up
   Flags: (0x1c008071) POINT-TO-POINT MULTICAST ARP RUNNING
   Type: PPP
   Internet address: 172.16.1.4/32
   Destination Internet address: 0.0.0.0
   Metric: 0, MTU: 1492, BW: 56 Kbps, DLY: 20000 usec, VRF: global
   Reliability 255/255, Txload 1/255, Rxload 1/255
   Last clearing of "show interface" counters never
   input peak rate 243 bits/sec,  0 hour 3 minutes 45 seconds ago
   output peak rate 288 bits/sec,  0 hour 3 minutes 45 seconds ago
   5 minutes input rate 0 bit/sec, 0 packet/sec
   5 minutes output rate 0 bit/sec, 0 packet/sec
```

16 packets received; 16 packets sent

0 multicast packets received

0 multicast packets sent

0 input errors; 0 output errors

0 collisions; 0 dropped

It can be observed that the dialer0 interface dialing succeeds and the negotiated IP address as 172.16.1.4.

#View the global routing table on Device1.

```
Device1#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

S   0.0.0.0/0 [1/178571] is directly connected, 00:00:39, dialer0
C   12.1.1.0/24 is directly connected, 00:14:51, gigabitethernet0
C   11.0.0.1/32 is directly connected, 00:00:39, dialer0
C   172.16.1.4/32 is directly connected, 00:00:39, dialer0
```

#View the L2TP information on Device2.

```
Device2#show vpdn detail
L2TP MaxTun 1000, MaxSes 1000:
tunnel free num: 999
 TUNNELS:
LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP       PORT
SES-CNT  STATE      REM-ADDR
15       45      lac             lns            lac              1701   1      ESTABLISHED
172.16.1.2
 session free num: 999
  SESSIONS:
LOCAL-ID REM-ID   TUN-ID   IF-NAME           SYSTEMID          IMSI/CALLING-
NUM       STATE
30       37       15       virtual-access0       --------          ----------------
ESTABLISHED

L2tp total Tunnel and Session Information. Tunnel 1  Session 1
```

It can be observed that theL2TP tunnel is established successfully between Device2 and Device3.

#View the L2TP information on Device3.

```
Device3#show vpdn detail

L2TP MaxTun 1000, MaxSes1000:
tunnel free num: 999
 TUNNELS:
LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP      PORT
SES-CNT  STATE      REM-ADDR
45       15      lns             lac            lns             1701   1      ESTABLISHED
172.16.1.1
 session free num: 999
 SESSIONS:
LOCAL-ID REM-ID  TUN-ID    IF-NAME          SYSTEMID          IMSI/CALLING-
NUM   STATE
37       30      45          virtual-access0      --------          ----------------
ESTABLISHED

L2TP total Tunnel and Session Information. Tunnel 1  Session 1
```

It can be observed that the L2TP tunnel is established between Device2 and Device3.

#View the global routing table on Device3.

```
Device3#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M -
Management
     D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C   172.16.1.0/24 is directly connected, 00:11:40, gigabitethernet0
C   11.0.0.1/32 is directly connected, 03:12:05, loopback0
C   172.16.1.4/32 is directly connected, 00:00:47, virtual-access0
```

#The interface IP address of virtual-template0 on Device3 can be pinged through on Device1.

```
Device1#ping 11.0.0.1

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 11.0.0.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

## 17.3.2. Configure L2TP Forced Mode Combining 4G Private Network

### Network Requirements

- Device1 acts as the 4G client, the LAC acts as the operator device, and Device2 acts as the LNS and connects to the ISP. Network-Center is the data center.
- Device1 dials in the operator through 4G. The LAC of the operator builds the L2TP connection with Device2. The communication between the 4G device and the data center is realized.

### Network Topology

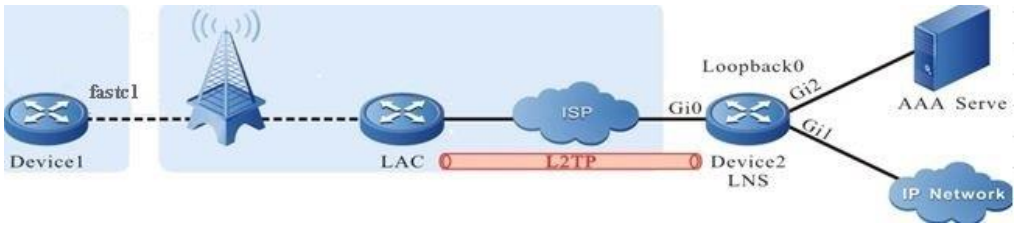



Figure 17-4 Networking of configuring the L2TP forced mode combining the 4G private network

| Device | Interface | IP Address | Device | Interface | IP Address |
|---|---|---|---|---|---|
| Device2 | Gi0 | 30.1.1.1/24 | Device2 | Loopback0 | 172.16.10.1/32 |
| | Gi2 | 130.255.142.29/24 | | | |

### Configuration Steps

**Step 1**: Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the 4G interface.

```
Device1#configure terminal
Device1(config)#interface fastcellular1/0
Device1(config)#dialer config apn cdmptx.sc
Device1(config-if-fastcellular1/0)#dialer config username test@jsyh.vpdn.sc password 0 admin
Device1(config-if-fastcellular1/0)#dialer config authtype pap
Device1(config-if-fastcellular1/0)#dialer config ipfamily ipv4
Device1(config-if-fastcellular1/0)#dialer mode auto
Device1(config-if-fastcellular1/0)#ip address dhcp
Device1(config-if-fastcellular1/0)#exit
```

**Note:**

- The APN name used on Device1 is provided by the operator.
- The 4G private network can be accessed through APN and domain name. The specific way is decided by the operator. The domain name is used for access in this case.

**Step 3:** Configure Device2 as AAA client.

#Configure Device2.

Configure the authentication and authorization list name as L2TP, and configure the address, authentication port, statistics port and radius server password of radius server （Refer to relevant chapter AAA of configuration manual).

```
Device2#configure terminal
Device2(config)#aaa server group radius l2tp
Device2(config-sg-radius-l2tp)# server 130.255.12.31 auth-port 1812 key l2tp
Device2(config-sg-radius-l2tp)# exit
Device2(config)#domain l2tp
Device2(config-isp-l2tpv2)# aaa authentication ppp radius-group l2tp
Device2(config-isp-l2tpv2)# aaa authorization ppp radius-group l2tp
Device2(config-isp-l2tpv2)# aaa accounting ppp start-stop radius-group l2tp
Device2(config-isp-l2tpv2)# exit
```

**Step 4:** Configure LNS.

#Configure Device2.

Configure the virtual template (virtual-template0), and use the AAA authentication and authorization list as l2tp.

```
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#peer default ip address 172.16.20.2
Device2(config-if-virtual-template0)#ppp authentication chap l2tp
Device2(config-if-virtual-template0)#ip unnumber loopback0
Device2(config-if-virtual-template0)#exit
```

Enable VPDN, create a VPDN group lns, and configure it to accept dial-in request mode, apply L2TP protocol, and borrow virtual template0. Configure the shared key of LAC and LNS as admin.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group lns
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol l2tp
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn-acc-in)#exit
Device2(config-vpdn)#local name lns
Device2(config-vpdn)#l2tp tunnel password 0 admin
Device2(config-vpdn)#exit
```

**Note:**

- If LAC does not need authentication, execute the **no l2tp tunnel authentication** command on LNS to disable authentication.

**Step 5:** Check the result.

#View the fastcellular 1/0 interface information of Device1.

```
Device1#show  interface fastcellular 1/0
Fastcellular1/0:
    line protocol is up
    Flags: (0xc208063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 172.16.10.6/30
    Broadcast address: 172.16.245.251
    Metric: 0, MTU: 1500, BW: 100000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 0001.7ab8.d858
    Last clearing of "show interface" counters never
    input peak rate 596 bits/sec,  1 hour 58 minutes 8 seconds ago
    output peak rate 715 bits/sec,  1 hour 58 minutes 8 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec
    5 minutes output rate 0 bit/sec, 0 packet/sec
    618 packets received; 1807 packets sent
    4 multicast packets received
    29 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
    Rate:  auto Duplex:  auto
    rxframes 618, rx bytes 52160, rx arps 21
    txframes 1807, tx bytes 308654, tx arps 25
    rx errors 0, tx errors 0
```

You can see that the 4G interface dialed successfully, and the negotiated IP address is 172.16.10.6.

#On Device2, view the L2TP information.

```
Device2#show vpdn detail
L2TP MaxTun 1000, MaxSes 1000:
 tunnel free num: 999
  TUNNELS:
```

```
LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP       PORT
SES-CNT  STATE      REM-ADDR
21       63      lns             GGSNCD01       lns              1701   1     ESTABLISHED
119.6.10.116
 session free num: 11999
  SESSIONS:
LOCAL-ID REM-ID  TUN-ID  IF-NAME          SYSTEMID          IMSI/CALLING-
NUM      STATE
79       4236    21      virtual-access2  --------          ----------------
ESTABLISHED
L2tp total Tunnel and Session Information. Tunnel 1 Session 1
```

You can see that the L2TP tunnel is set up between the operator LAC and Device2 successfully.

#On Device1, you can ping the virtual-template0 interface address of Device2.

```
Device1#ping 172.16.20.1


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172.16.20.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

## 17.3.3. Configure L2TP Spontaneous Mode Combining 4G Public Network

### Network Requirements

- Device1 acts as the 4G client, Device1 as the LAC, and Device2 as the LNS. The LNS can visit the public network. Network-Center is the data center.
- Device1 dials in the operator through 4G. After connecting with Device2, Device1 builds the L2TP connection with Device2, realizing the communication between 4G client and Network-Center.

### Network Topology

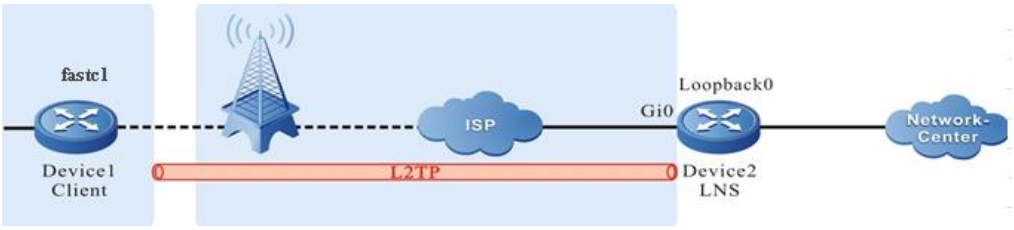



Figure 17-5 Networking of configuring the L2TP spontaneous mode combining the 4G public network

| Device | Interface | IP Address | Device | Interface | IP Address |
|---|---|---|---|---|---|
| Device2 | Gi0 | 171.217.50.205/24 | Device2 | Loopback0 | 100.0.0.1/32 |

## Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the auto dialing of the 4G interface.

```
Device1#configure terminal
Device1(config)#interface fastcellular1
Device1(config-if-fastcellular1)#dialer config authtype pap
Device1(config-if-fastcellular1)#dialer config ipfamily ipv4
Device1(config-if-fastcellular1)#dialer mode auto
Device1(config-if-fastcellular1)#ip address dhcp
Device1(config-if-fastcellular1)#exit
```

#Configure the one route to Device2, pointing to 4G interface.

```
Device1(config)#ip route 171.217.50.205 255.255.255.255 fastcellular1
```

#Check the 4G interface information of Device1.

```
Device1#show  interface fastcellular 1
Fastcellular1:
    line protocol is up
    Flags: (0xc208063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 10.159.125.38/30
    Broadcast address: 10.159.245.251
    Metric: 0, MTU: 1500, BW: 100000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 0001.7ab8.d858
    Last clearing of "show interface" counters never
    input peak rate 596 bits/sec,  1 hour 58 minutes 8 seconds ago
    output peak rate 715 bits/sec,  1 hour 58 minutes 8 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec
    5 minutes output rate 0 bit/sec, 0 packet/sec
    618 packets received; 1807 packets sent
    4 multicast packets received
    29 multicast packets sent
    0 input errors; 0 output errors
```

0 collisions; 0 dropped

Unknown protocol 0

Rate: auto Duplex: auto

rxframes 618, rx bytes 52160, rx arps 21

txframes 1807, tx bytes 308654, tx arps 25

rx errors 0, tx errors 0

You can see that the 4G interface dials successfully and the negotiated IP address is 10.159.125.38.

#On Device1, view the global route table.

Device1#show ip route

Codes: C - connected, S - static, R - RIP, O - OSPF, OE-OSPF External, M - Management

D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

S 171.217.50.205/32 [1/26041] is directly connected, 00:02:19, fastcellular1

C 10.159.125.38/32 is directly connected, 00:02:19, fastcellular1

C 172.22.209.210/32 is directly connected, 00:02:19, fastcellular1

#On Device1, you can ping the public network address of Device2.

Device1#ping 171.217.50.205

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 171.217.50.205 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.

**Step 3:** Configure LAC abd client.

#Configure Device1.

Configure the pseudo wire template name as l2tp-1, the encapsulation protocol type as l2tpv2, the key as admin, the local name as lac, and borrow fast cellular 1.

Device1(config)#pseudowire-class l2tp-1

Device1(config-pw-class)#encapsulation l2tpv2

Device1(config-pw-class)#password 0 admin

Device1(config-pw-class)#hostname lac

Device1(config-pw-class)#ip local interface fastcellular1

Device1(config-pw-class)#exit

Configure the virtual PPP interface virtual-ppp0, encapsulate the PPP protocol, and use the PAP authentication method to send the user name as admin and password as admin; Use pseudo wire template l2tp-1 to establish L2TP tunnel with Device2.

```
Device1(config)#interface virtual-ppp0
Device1(config-if-virtual-ppp0)#encapsulation ppp
Device1(config-if-virtual-ppp0)#ip address negotiated
Device1(config-if-virtual-ppp0)#ppp pap sent-username admin password 0 admin
Device1(config-if-virtual-ppp0)#pseudowire 171.217.50.205 1 pw-class l2tp-1
Device1(config-if-virtual-ppp0)#exit
```

Configure the default route out interface as virtual-ppp0.

```
Device1(config)#ip route 0.0.0.0 0.0.0.0 virtual-ppp0
```

**Step 4:** Configure the LNS device.

#Configure Device2.

Configure the PPP-authenticated user name as admin and password as admin.

```
Device2#configure terminal
Device2(config)#local-user admin class network
Device2(config-user-network-admin)#service-type ppp
Device2(config-user-network-admin)#password 0 admin
Device2(config-user-network-admin)#exit
```

Configure virtual-template 0 and use the PAP authentication mode.

```
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#ip unnumber loopback0
Device2(config-if-virtual-template0)#peer default ip address 100.0.0.2
Device2(config-if-virtual-template0)#ppp authentication pap
Device2(config-if-virtual-template0)#exit
```

Enable the VPDN. Create the VPDN group lns and configure it as the accept dial-in request mode. Use the application L2TP protocol and virtual-template0. Configure the shared key between the LAC and LNS as admin.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group lns
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol l2tp
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn-acc-in)#exit
Device2(config-vpdn)#local name lns
Device2(config-vpdn)#l2tp tunnel password 0 admin
Device2(config-vpdn)#exit
```

**Step 5:** Check the result.

#View the information of interface virtual-ppp0 on Device1.

```
Device1#show interface virtual-ppp 0
virtual-ppp0:
    line protocol is up
    Flags: (0x80080f1) POINT-TO-POINT MULTICAST RUNNING
    Type: PPP
    Internet address: 100.0.0.2/24
    Destination Internet address: 100.0.0.1
    Metric: 0, MTU: 1500, BW: 64 Kbps, DLY: 20000 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Last clearing of "show interface" counters never
    input peak rate 649 bits/sec,  0 hour 1 minute 11 seconds ago
    output peak rate 649 bits/sec,  0 hour 1 minute 11 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec
    5 minutes output rate 0 bit/sec, 0 packet/sec
    59 packets received; 64 packets sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
     LCP:OPENED
     IPCP:OPENED
     encap-type: simply PPP
```

#View the L2TP information on Device1.

```
Device1#show vpdn detail
L2TP MaxTun1000, MaxSes1000:
 tunnel free num: 999
  TUNNELS:
LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP       PORT
SES-CNT  STATE      REM-ADDR
1        33      lac              lns                          1701   1       ESTABLISHED
171.217.50.205
 session free num: 999
  SESSIONS:
LOCAL-ID REM-ID  TUN-ID  IF-NAME          SYSTEMID          IMSI/CALLING-
NUM       STATE
```

```
81      66      1       virtual-ppp0        --------        ----------------
ESTABLISHED


L2TP total Tunnel and Session Information. Tunnel 1  Session 1
```

It can be observed that the L2TP tunnel between Device1 and Device2 is successfully established.

#The IP address of the interface virtual-template0 on Device2 can be pinged through on Device1.

```
Device1#ping 100.0.0.1


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 100.0.0.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

### 17.3.4. Configure Android Terminal Establishing L2TP over IPSec Connection on Public Network

#### Network Requirements

- Android is the Android terminal, Device is the LNS, and Network-Center is the data center. The IPSec is established between Android and Device through the public network address. And then the L2TP connection is established. The IPSec transmission mode is used to protect the L2TP data communication between Android and Device.
- The L2TP tunnel is established between Android and Device, realizing the communication between Android and data center.

#### Network Topology

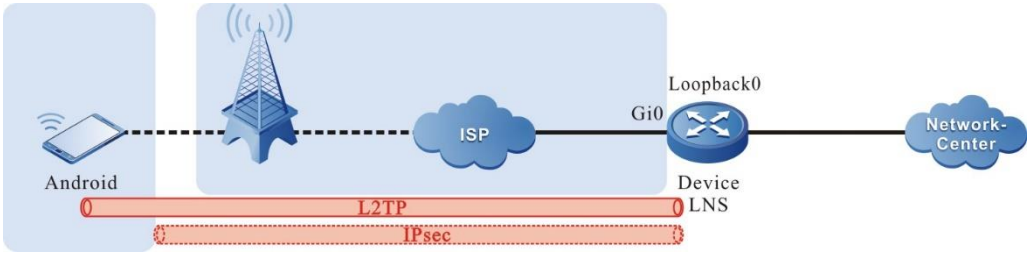
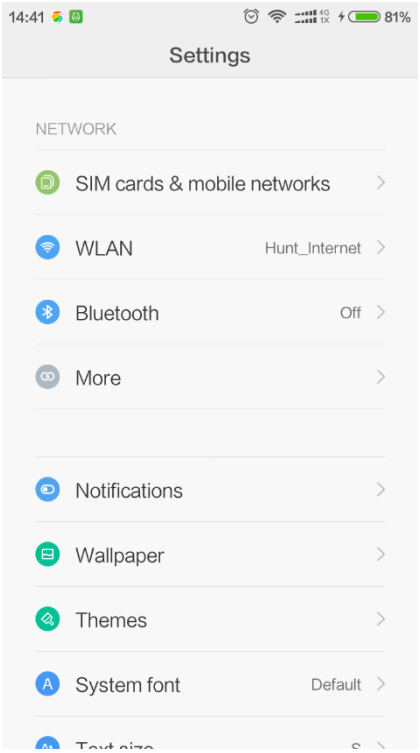



Figure 17-6 Networking of configuring Android establishing the L2TP over IPSec on the public network

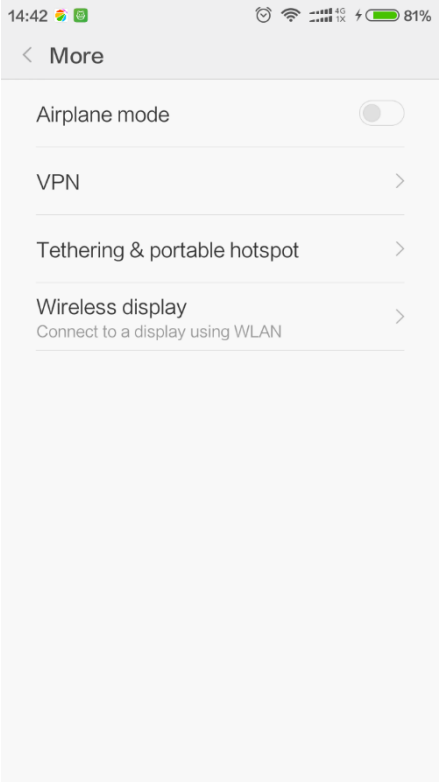
#### Configuration Steps

**Step 1:** Configure Android.

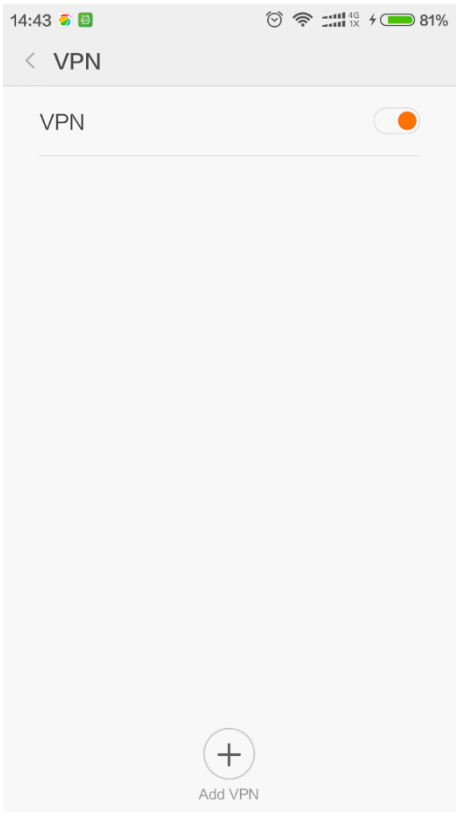
Choose **Settings** > **WLAN** on Android, as shown in the following figure.



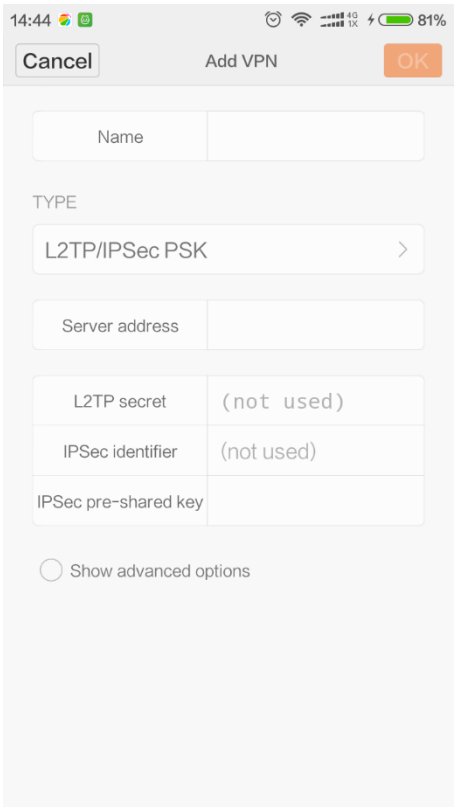

Enter the **Wireless and network settng** interface and click **VPN**, as shown in the following figure.



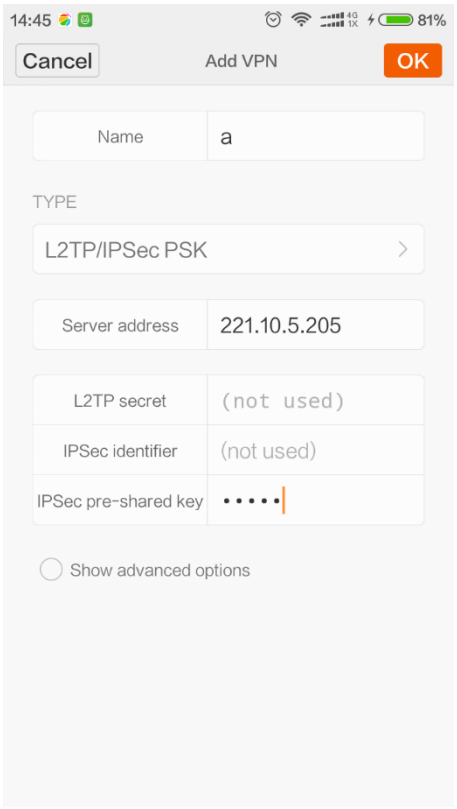

Enter the VPN interface and click **Add VPN**, as shown in the following figure.



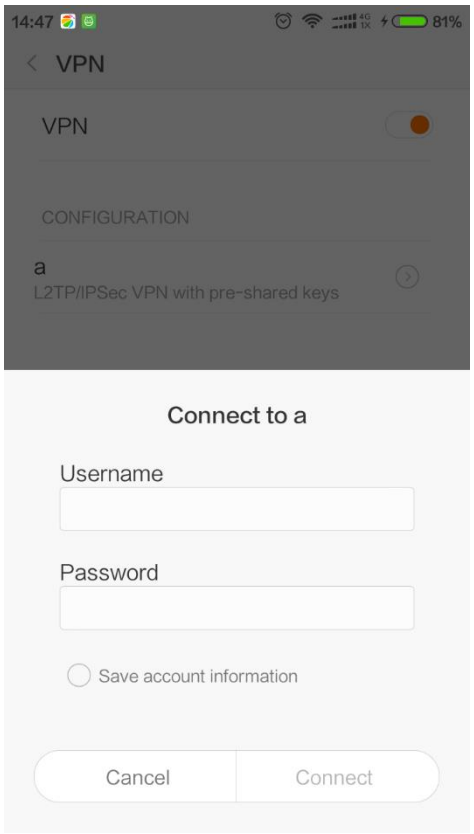

Enter the **Add VPN** interface and click **Add VPN L2TP/IPsec PSK**, as shown in the following figure.



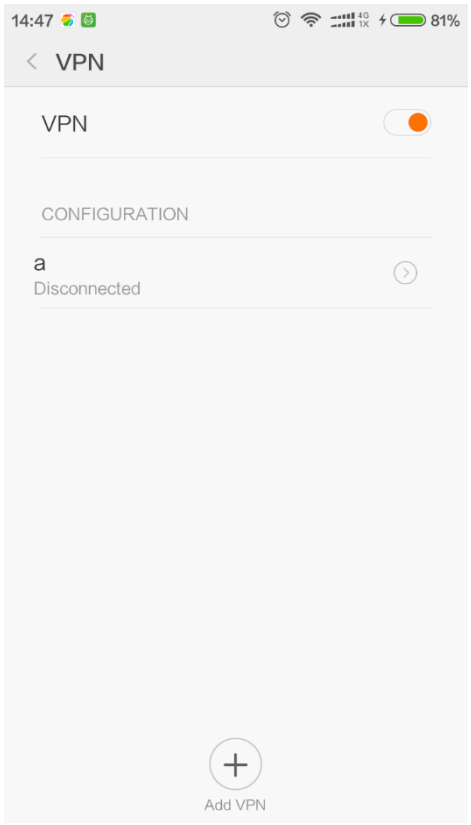

Enter the **Add VPN L2TP/IPsec PSK"** interface, and configure the VPN name, VPN server, and IPSec pre-shared key, as shown in the following figure.




Save the preceding configurations and set the password, as shown in the following figure.




After the password is successfully set, click the preceding configured network on the **VPN** interface to connect to the VPN, as shown in the following figure.




**Step 2:** Configure the LNS device.

#Configure Device.

Configure the IP address of the interface gigabitethernet0 connecting to the public network.

```
Device#configure terminal
Device(config)#interface gigabitethernet0
Device(config-if-gigabitethernet0)#ip address 221.10.5.205 255.255.255.0
Device(config-if-gigabitethernet0)#exit
```

Configure the PPP-authenticated user name as admin and password as admin.

```
Device(config)#local-user admin class network
Device(config-user-network-admin)#service-type ppp
Device(config-user-network-admin)#password 0 admin
Device(config-user-network-admin)#exit
```

Configure the loopback interface loopback0.

```
Device(config)#interface loopback0
Device(config-if-loopback0)#ip address 172.16.30.1 255.255.255.255
Device(config-if-loopback0)#exit
```

Create virtual-template0 and encapsulation the PPP protocol. Use the CHAP authentication mode and Loopback0.

```
Device(config)#interface virtual-template 0
```

```
Device(config-if-virtual-template0)#encapsulation ppp
Device(config-if-virtual-template0)#peer default ip address 2.2.2.2
Device(config-if-virtual-template0)#ppp authentication chap
Device(config-if-virtual-template0)#ppp mtu adaptive proxy
Device(config-if-virtual-template0)#ip unnumber loopback0
Device(config-if-virtual-template0)#exit
```

Enable the VPDN. Create the VPDN group lns and configure it as the accept dial-in request mode. Use the application L2TP protocol and virtual-template0. Disable the L2TP authentication.

```
Device(config)#vpdn enable
Device(config)#vpdn-group lns
Device(config-vpdn)#accept-dialin
Device(config-vpdn-acc-in)#protocol l2tp
Device(config-vpdn-acc-in)#virtual-template 0
Device(config-vpdn-acc-in)#exit
Device(config-vpdn)#local name lns
Device(config-vpdn)#terminate-from hostname anonymous
Device(config-vpdn)#no l2tp tunnel authentication
Device(config-vpdn)#lcp renegotiation on-mismatch
Device(config-vpdn)#exit
```

**Step 3:** Configure the IPSec. (For details, refer to the IPSec chapter in the configuration manual.

#Configure Device.

Configure the preshared key on Device with the key as 123, permitting all peer ends using the key.

```
Device(config)#crypto ike key 123 any
```

Configure the IKE proposal android on Device. Configure the encryption algorithm 3DES and authentication algorithm MD5.

```
Device(config)#crypto ike proposal android
Device(config-ike-prop)#encryption 3des
Device(config-ike-prop)#integrity md5
Device(config-ike-prop)#group group2
Device(config-ike-prop)#exit
```

Configure the IPSec proposal android on Device. Configure the ESP security protocol, encryption algorithm 3DES, and authentication algorithm SHA1.

```
Device(config)#crypto ipsec proposal android
Device(config-ipsec-prop)#esp 3des sha1
Device(config-ipsec-prop)#mode transport
```

```
Device(config-ipsec-prop)#exit
```

Configure the tunnel android on Device. Use the IP address 221.10.5.205 as the local end IP address of the tunnel and the IP address of the tunnel as any. Configure the authentication mode as pre-shared key authentication. The IKE proposal uses android and IPSec uses android.

```
Device(config)#crypto tunnel android
Device(config-tunnel)#local address 221.10.5.205
Device(config-tunnel)#peer any
Device(config-tunnel)#set authentication preshared
Device(config-tunnel)#set ike proposal android
Device(config-tunnel)#set ipsec proposal android
Device(config-tunnel)#exit
```

Configure the security policy pad on Device to protect the IP communication from the host 221.10.5.205 to any network. Associate the tunnel android.

```
Device(config)#crypto policy pad
Device(config-policy)#flow host 221.10.5.205 any udp 1701 any tunnel android bypass
Device(config-policy)#exit
```

- At last of the configuration, a routing to Android needs to be configured.

**Step 4:** Check the result.

Configure Android and build the L2TP connection.

Enter theVPN established in step 1 and enter the user name and password. Click **OK**. It can be viewed that the c**onnecting** is displayed under the VPN name. After the connection succeeds, **the connected** is displayed, as shown in the following figure.


#View the IPSec information on Device.

```
Device#show crypto ike sa
sa-id   negotiation-state    localaddr       peeraddr        peer-identity
 9409    STATE_QUICK_R2      221.10.5.205    119.4.252.213   10.231.253.187
 9408    STATE_MAIN_R3       221.10.5.205    119.4.252.213   10.231.253.187
Device#show crypto ipsec sa
policy name : pad
  f (src, dst, protocol, src port, dst port) :  221.10.5.205/32  0.0.0.0/0 udp  1701  any
policy name : subflow-1610612776, the parent policy name : pad
  f (src, dst, protocol, src port, dst port) :  221.10.5.205/32  119.4.252.213/32 udp  1701
any
  local tunnel endpoint : 221.10.5.205 remote tunnel endpoint : 119.4.252.213
  the pairs of ESP ipsec sa : id : 9409, algorithm : 3DES HMAC-SHA1-96
    inbound esp ipsec sa :  spi : 0x4cb51778(1286936440)  crypto context : 0x1dc0a620
       current input 256 packets, 23 kbytes
       encapsulation mode : UDP-Encapsulation-Transport
       replay protection : ON
       remaining lifetime (seconds/kbytes) : 27985/4294967271
       uptime is 0 hour 13 minute 35 second
```

```
        outbound esp ipsec sa :  spi : 0x556d8d9(89577689)  crypto context : 0x1bfb9d40
          current output 216 packets, 24 kbytes
          encapsulation mode : UDP-Encapsulation-Transport
          replay protection : ON
          remaining lifetime (seconds/kbytes) : 27985/4294967270
          uptime is 0 hour 13 minute 35 second
      total sa and sa group is 1
```

#View the L2TP information on Device.

```
Device#show vpdn detail
L2tp MaxTun 1000, MaxSes 1000:
 tunnel free num: 999
  TUNNELS:
LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP       PORT
SES-CNT  STATE          REM-ADDR
94     18578   lns          anonymous       lns          42719   1     ESTABLISHED
119.4.252.213
 session free num: 999
  SESSIONS:
LOCAL-ID REM-ID   TUN-ID   IF-NAME          SYSTEMID          IMSI/CALLING-
NUM    STATE
28       51669    94       virtual-access1       --------          ----------------
ESTABLISHED


L2TP total Tunnel and Session Information. Tunnel 1  Session 1
```

It can be observed that the L2TP tunnel between Android and Device is successfully established.

### 17.3.5. Configure Android Terminal Establishing L2TP over IPSec Connection on Private Network

#### Network Requirements

- Android is the Android terminal, LAC is the operator device, Device is the LNS, and Network-Center is the data center. The IPSec is established between Android and Device through the private network IP address. And then the L2TP connection is established. The IPSec transmission mode is used to protect the L2TP data communication between Android and Device.
- The L2TP tunnel is established between Android and Device, realizing the communication between Android and Network-Center.

## Network Topology

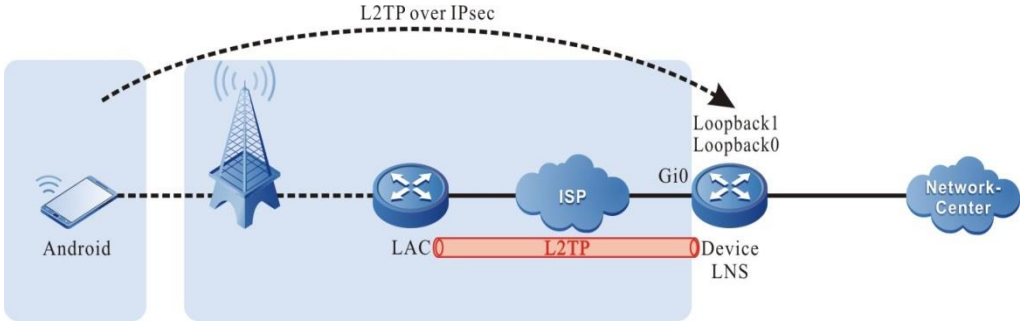


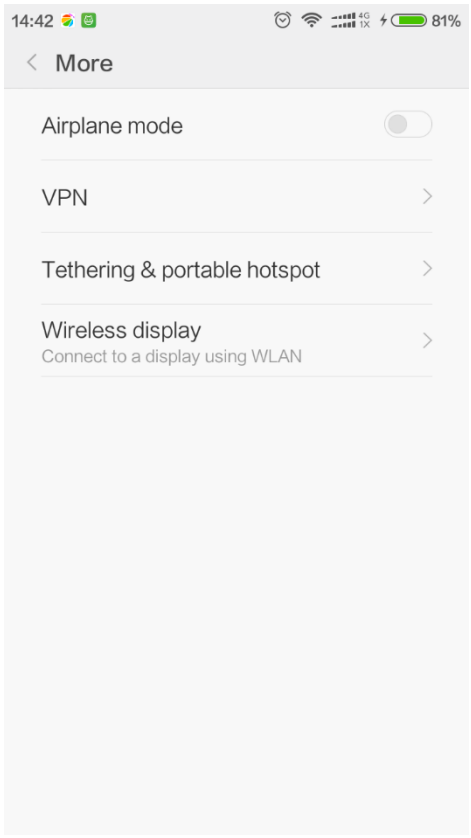

Figure 17-7 Networking of configuring Android establishing the L2TP over IPSec on the private network

## Configuration Steps

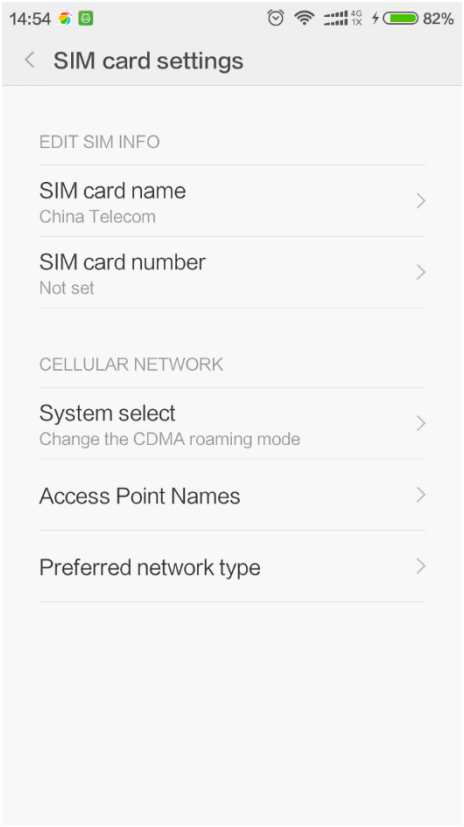
**Step 1:** Configure Android.

Because of the dialing in the private network, the terminals of some operators need to change the ANP required for dialing the private network.

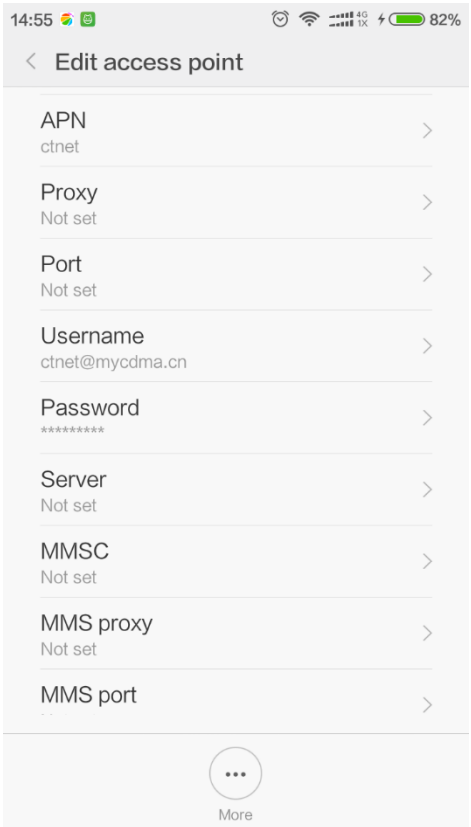
On the **Settings** interface on Android, click **VPN,** as shown in the following figure.




Enter the **SIM card settings** interface and click **Access Point Name (APN)**, as shown in the following figure.




On the **Edit access point** interface, set APN, user name, and password, as shown in the following figure.



After the modification, choose the APN for access (G here). Then the terminal will initiate the connection.

**Note:**

- In this case, the private network of Rostelecomis used. Therefore, the APN for Android needs to be set.

**Step 2:** Configure the LNS information on Device.

#Configure Device.

Configure the IP address of gigabitethernet0 connecting to the public network.

```
Device#configure terminal
Device(config)#interface gigabitethernet0
Device(config-if-gigabitethernet0)#ip address 221.10.5.205 255.255.255.0
Device(config-if-gigabitethernet0)#exit
```

Configure the PPP-authenticated user name as admin and password as admin. Configure the VPDN user name and password consistent with Android.

```
Device(config)#local-user admin class network
Device(config-user-network-admin)#service-type ppp
Device(config-user-network-admin)#password 0 admin
Device(config-user-network-admin)#exit
Device(config)#
Device(config)#local-user admin@cctv.vpdn.sc class network
Device(config-user-network-admin@cctv.vpdn.sc)#service-type ppp
Device(config-user-network-admin@cctv.vpdn.sc)#password 0 admin
Device(config-user-network-admin@cctv.vpdn.sc)#exit
```

Configure the loopback interfaces loopback0 and loopback1.

```
Device(config)#interface loopback0
Device(config-if-loopback0)#ip address 172.16.10.1 255.255.255.255
Device(config-if-loopback0)#exit
Device(config)#interface loopback1
Device(config-if-loopback1)#ip address 172.16.30.1 255.255.255.255
Device(config-if-loopback1)#exit
```

Create virtual-template0, encapsulation PPP protocol. Use the CHAP authentication mode and use Loopback0.

```
Device(config)#interface virtual-template 0
Device(config-if-virtual-template0)#encapsulation ppp
Device(config-if-virtual-template0)#peer default ip address 1.1.1.1
Device(config-if-virtual-template0)#ppp authentication chap
```

```
Device(config-if-virtual-template0)#ppp mtu adaptive proxy
Device(config-if-virtual-template0)#ip unnumber loopback0
Device(config-if-virtual-template0)#exit
```

Create virtual-template1 and encapsulation PPP protocol. Use the CHAP authentication mode and Loopback1.

```
Device(config)#interface virtual-template 1
Device(config-if-virtual-template1)#encapsulation ppp
Device(config-if-virtual-template1)#peer default ip address 22.22.22.22
Device(config-if-virtual-template1)#ppp authentication chap
Device(config-if-virtual-template1)#ppp mtu adaptive proxy
Device(config-if-virtual-template1)#ip unnumber loopback1
Device(config-if-virtual-template1)#exit
```

Enable the VPDN. Create the VPDN group lns0 and configure it as the accept dial-in request mode. Use the application PPP protocol and virtual-template0. Disable the L2TP authentication. This VPDN group is used for the first time dialing in the private network.

```
Device(config)#vpdn enable
Device(config)#vpdn-group lns0
Device(config-vpdn)#accept-dialin
Device(config-vpdn-acc-in)#protocol l2tp
Device(config-vpdn-acc-in)#virtual-template 0
Device(config-vpdn-acc-in)#exit
Device(config-vpdn)#local name lns
Device(config-vpdn)#no l2tp tunnel authentication
Device(config-vpdn)#lcp renegotiation on-mismatch
Device(config-vpdn)#exit
```

Create the VPDN group lns1 and configure it as the accept dial-in request mode. Use the application L2TP protocol and virtual-template1. Disable the L2TP authentication. This VPDN group is used for anonymous terminal dialing.

```
Device(config)#vpdn-group lns1
Device(config-vpdn)# accept-dialin
Device(config-vpdn-acc-in)#protocol l2tp
Device(config-vpdn-acc-in)#virtual-template 1
Device(config-vpdn-acc-in)#exit
Device(config-vpdn)#local name lns1
Device(config-vpdn)#terminate-from hostname anonymous
Device(config-vpdn)#no l2tp tunnel authentication
Device(config-vpdn)#lcp renegotiation on-mismatch
Device(config-vpdn)#exit
```

**Step 3:** Configure the IPSec. (For details, see the IPSec chapter in the configuration manual.)

#Configure Device.

Configure the pre-shared key on Device with the key as 123, permitting all peer ends to use the key.

```
Device(config)#crypto ike key 123 any
```

Configure the IKE proposal pas on Device, using the encryption algorithm 3DES and authentication algorithm MD5.

```
Device(config)#crypto ike proposal pad
Device(config-ike-prop)#encryption 3des
Device(config-ike-prop)#integrity md5
Device(config-ike-prop)#group group2
Device(config-ike-prop)#exit
```

Configure the IPSec proposal pad on Device. Use the ESP security protocol and encryption algorithm 3DES and authentication algorithm SHA1.

```
Device(config)#crypto ipsec proposal pad
Device(config-ipsec-prop)#esp 3des sha1
Device(config-ipsec-prop)#mode transport
Device(config-ipsec-prop)#exit
```

Configure the tunnel android on Device. Use the IP address 172.16.10.1 as the local end IP address of the tunnel. Configure the peer end IP address of the tunnel as any and the authentication mode as pre-shared key authentication. The IKE proposal uses the android and the IPSec proposal uses the android.

```
Device(config)#crypto tunnel android
Device(config-tunnel)#local address 172.16.10.1
Device(config-tunnel)#peer any
Device(config-tunnel)#set authentication preshared
Device(config-tunnel)#set ike proposal pad
Device(config-tunnel)#set ipsec proposal pad
Device(config-tunnel)#exit
```

Configure the security policy android on Device to protect the IP communication from the IP address 172.16.10.1 to any network. Associate the tunnel android.

```
Device(config)#crypto policy android
Device(config-policy)#flow host 172.16.10.1 any udp 1701 any tunnel android
Device(config-policy)#exit
```

**Note:**

- At last of the configuration, a routing to Android needs to be configured.

**Step 4:** Check the result. Android initiates the connection.

Long press the established VPN and the network connection succeeds. Enter the ppp user name and password and then click **OK**. It can be viewed that **the connecting** is displayed under the VPN name. After the connection succeeds, **the connected** is displayed, as shown in the following figure.


#View the IPSec information on Device.

```
Device#show crypto ike sa
sa-id   negotiation-state    localaddr      peeraddr       peer-identity
 10290   STATE_QUICK_R2      172.16.10.1    1.1.1.1        1.1.1.1
 10289   STATE_MAIN_R3       172.16.10.1    1.1.1.1        1.1.1.1
Device#show crypto ipsec sa
policy name : pad
  f (src, dst, protocol, src port, dst port) :  172.16.10.1/32  0.0.0.0/0 udp  1701  any
policy name : subflow-1610612823, the parent policy name : pad
  f (src, dst, protocol, src port, dst port) :  172.16.10.1/32  1.1.1.1/32 udp  1701  any
  local tunnel endpoint : 172.16.10.1 remote tunnel endpoint : 1.1.1.1
  the pairs of ESP ipsec sa : id : 10290, algorithm : 3DES HMAC-SHA1-96
    inbound esp ipsec sa :  spi : 0x60111953(1611733331)  crypto context : 0x1b8b08e0
       current input 14 packets, 0 kbytes
       encapsulation mode : Transport
       replay protection : ON
```

```
        remaining lifetime (seconds/kbytes) : 28785/4294967294
        uptime is 0 hour 0 minute 15 second
      outbound esp ipsec sa :  spi : 0xfd3817c(265519484)  crypto context : 0x1b8b0860
        current output 14 packets, 0 kbytes
        encapsulation mode : Transport
        replay protection : ON
        remaining lifetime (seconds/kbytes) : 28785/4294967294
        uptime is 0 hour 0 minute 15 second
    policy name : test
     f (src, dst, protocol, src port, dst port) :  0.0.0.0/0  18.0.0.0/24  ip  any  any
    total sa and sa group is 1
```

#View the L2TP information on Device.

```
    Device#show vpdn detail
    L2TP MaxTun1000, MaxSes1000:
     tunnel free num: 998
      TUNNELS:
    LOCAL-ID REM-ID  LOCAL-NAME      REM-NAME       VPDN-GROUP       PORT
    SES-CNT  STATE      REM-ADDR
    28    17701   lns1          anonymous       lns1          58419  1     ESTABLISHED
    1.1.1.1
    109   85      lns0          GGSNCD02        lns0          1701   1     ESTABLISHED
    119.6.10.2
     session free num: 998
      SESSIONS:
    LOCAL-ID REM-ID   TUN-ID   IF-NAME          SYSTEMID          IMSI/CALLING-
    NUM    STATE
    445      19270    28      virtual-access785      --------          ----------------
    ESTABLISHED
    1084     173      109     virtual-access660      --------          ----------------
    ESTABLISHED


    L2TP total Tunnel and Session Information. Tunnel 2  Session 2
```

It can be observed that the L2TP tunnel is established between Android and Device.

# 18. L2TPV3

## 18.1. Overview

The L2TPv3 (Layer 2 Tunneling Protocol version 3) is the previous Cisco proprietary protocol of the UTI (Universal Transport Interface). The UTI aims to provide the IP-based tunnel technology mechanism of high performance for the connectivity of layer 2 similar to the circuit on the group-based core network. It is an application mode for the layer 2 VPN (Virtual Private Network) transmission technology.

The L2TPv3 is the extension of the layer 2 tunnel UTI by the IETF standard. To overcome the UTI restrictions, the L2TPv3 builds the UTI encapsulation format and combines with a large number of the optional signaling mechanism from the L2TPv2 control plane to provide the pseudo-wire connectivity. The basic L2TPv3 protocol is completed by the control channel signaling of the extended L2TPv2. The signaling supports other attributes which are transferred in the AVP (Attribute Value Pairs) message format.

The L2TPv3 control channel signaling operates on two stages: control connection establishment and session connection establishment. It uses three message exchanges to build the control connection. The following figure shows the tunnel establishment process.

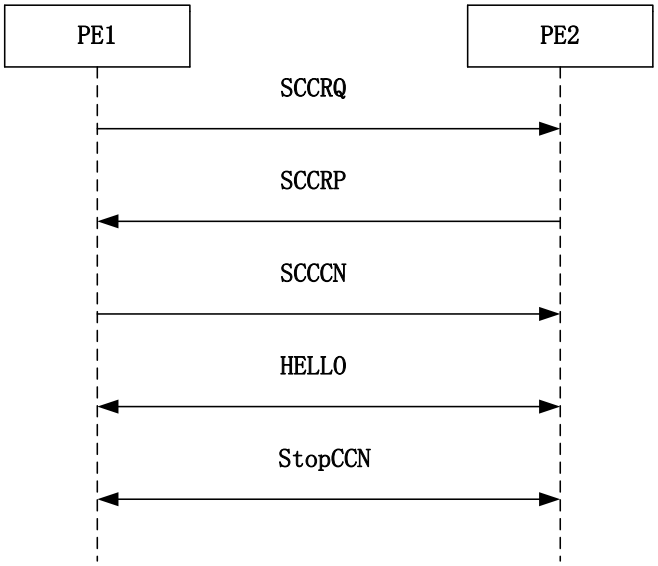



Figure 18-1 The L2TPv3 tunnel establishment process

- After the L2TPv3 node is defined on the PE (Provider Edge), the SCCRQ (Start Control Connection Request) message is sent to initiate the control channel to the PE and inform the performance supported by the local PE.
- The PE sends the SCCRP (Start Control Connection Reply) message to announce its performance settings. This message indicates that the control connection can continue.
- At last, the SCCCN (Start Control Connection Connected) message is the response message of the SCCRP message. The SCCCN message confirms that the SCCRP message is received and the control connection establishment is completed.

After the control co9nnection is established, the two nodes send the keep-alive message to detect the invalid node at a regular interval, which is called the keep-alive mechanism. Due to the overdue keep-alive timer or other key errors, use the StopCCN (Stop Control Connection Notification) message to release the control channel.

After the tunnel is established successfully, the independent L2TPv3 session can be established. The session negotiation uses the three-way handshake mechanism. The following figure shows the session establishment process.

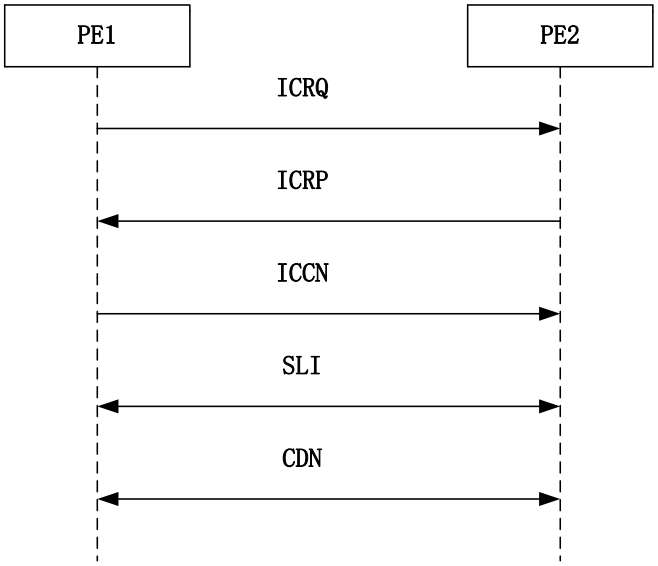



Figure 18-2 The L2TPv3 session establishment process

- When the junction circuit switches to the active state, the PE router sends CRQ (Incoming Call Request) message to exchange the parameter information of the session.
- When the remote PE receives the ICRQ message, the PE sends an ICRP (Incoming Call Reply) message to indicates that the ICRQ message is received.
- The ICCN (Incoming Call Connected) message is sent as the response message for receiving the ICRQ message.

Each pseudo-wire that needs to be dynamically established requires three-way handshake. To send the signaling of the single session state, any PE can send the SLI (Set Link Information) message to indicate the junction circuit state changes. When a node receives the message, the session and related resources must be released.

## 18.2. L2TPv3 Function Configuration

Table 18-1 L2TPv3 function list

| Configuration Task | |
|---|---|
| Configure the L2TPv3 template | Configure the L2TPv3 basic template |
| | Configure the L2TPv3 packet not allowing fragmentation |
| | Configure the cookie |

| Configuration Task | |
|---|---|
| Configure the L2TPv3 template | Configure the control connection authentication |
| | Configure the timeout retransmission parameter |
| Enable the L2TPv3 function | Enable the dynamic L2TPv3 session |
| | Enable the static L2TPv3 session |

### 18.2.1. Configure L2TPv3 Template

#### Configuration Condition

None

#### Configure L2TPv3 Basic Template

The L2TPv3 template is the aggregation to save the tunnel and session related features, including the data encapsulation type, tunnel and session signaling protocol, source IP address, and other related attributes, such as authentication, fragmentation, and serial number mechanism.

Table 18-2 Configure the L2TPv3 basic template

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Create the pseudo-wire template and enter the pseudo-wire configuration mode | **pseudowire-class** *class-name* | Mandatory<br>By default, the pseudo-wire template is not created. |
| Configure the data encapsulation type | **encapsulation l2tpv3** | Mandatory<br>By default, the data encapsulation is not configured. |

| Step | Command | Description |
|---|---|---|
| Configure the signaling protocol | **protocol { l2tpv3 \| none }** | Optional<br>By default, the static session mode (**none**) is used. |
| Associate the interface of the source IP address | **ip local interface** *interface-name* | Mandatory<br>By default, the interface of the source IP address if not associated. |

**Note:**

- When enabling the dynamic session, run the **protocol { l2tpv3 | none }** command to configure the L2TPv3 signaling protocol. The static session is configured as none.

### Configure L2TPv3 Packet not Allowing Fragmentation

The IP packet, the encapsulated L2 packet by the L2TPv3, can be set to allow or mot allow the fragmentation in the introduced pseudo-wire template. By default, the packet is allowed to be fragmented.

Table 18-3 Configure the L2TPv3 packet not allowing fragmentation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Configure the L2TPv3 packet not allowing fragmentation | **ip dfbit set** | Mandatory<br>By default, the L2TPv3 packet is allowed to be fragmented. |

### Configure Cookie

The cookie filed is the optional field in the L2TPv3 protocol header. This field is extended and the value is random with a maximum of 64 bits. The cookie value further ensures that the received data is correctly associated with the local circuit based on the session ID. The randomly selected cookie can only prevent blind insertion attack. Like the session ID, the cookie value only can be used locally. During the session, the AVP local session ID and specified cookie are informed to the peer PE.

Table 18-4 Configure the cookie

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Configure the cookie | **cookie size { 4 \| 8 }** | Mandatory<br>By default, the cookie is not configured. |

### Configure Control Connection Authentication

The L2TPv3 uses a simple and configurable CHAP in the control connection establishment process. If both ends or any end of the PE need authentication, then an AVP containing the CHAP message exist in the SCCRQ or SCCRP control packet. The end initiates the authentication will transmit data to the peer end randomly and the expected hashed value is calculated based on the random number and shared password on the local end. Upon receiving such type of authentication, the SCCRP or SCCCN control packet with the hashed value is sent to respond, respectively. If the responded hashed value does not match the expected hashed value, the tunnel cannot be established.

Table 18-5 Configure the control connection authentication

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Co9nfigure the host name | **hostname** *host-name* | Optional<br>By default, the host name is used. |

| Step | Command | Description |
|---|---|---|
| Enable the control connection authentication | **authentication** | Mandatory<br>By default, the control connection authentication is not enabled. |
| Configure the authentication password | **password 0** *password* | Mandatory<br>By default, the authentication password is not configured. |

### Configure Timeout Retransmission Parameter

To ensure the reliable transmission of the protocol control packet, the L2TPv3 adopts the packet retransmission mechanism. The retransmission times and retransmission interval of the protocol control packet are set in the pseudo-wire template as required.

Table 18-6 Configure the timeout retransmission parameter

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Configure the session connection establishment interval | **retransmit initial timeout** *timeout-value* | Optional<br>By default, the timeout time for the session is 5s. |
| Configure the retransmission times of the tunnel connection establishment | **retransmit initial retries** *retries-value* | Optional<br>By default, the retransmission times for the tunnel connection establishment request packet is 2. |

| Step | Command | Description |
|---|---|---|
| Configure the retransmission times of the protocol packet | **retransmit retries** *retry-value* | Optional<br>By default, the retransmission number of the control packet (except the tunnel connection establishment request packet) is 10. |
| Configure the timeout time of the protocol packet | **retransmit timeout** { **max** \| **min** } *timeout-value* | Optional<br>By default, the maximum timeout time and minimum timeout time for the control packet is 8s and 1s. |

**Configure Tunnel Keepalive Time**

After the L2TPv3 tunnel is successfully established, in order to detect whether the peer device can communicate normally, the L2TPv3 device uses the method of sending the keepalive packets regularly to confirm the connectivity of the peer device.

Table 18-7 Configure the tunnel keepalive time

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Configure the tunnel keepalive time | **hello** *time-value* | Mandatory<br>By default, the tunnel keepalive time is 60s.<br>After the L2TPv3 tunnel is created, send the packets regularly to confirm whether the tunnel can communicate normally. |

**Note:**

- The **no hello** command is used to prohibit detecting the connectivity of the tunnel.

### Configure Tunnel Connection Status Detection

The command is used to configure the packets passing the tunnel to detect the connectivity of the tunnel. Once the local tunnel can receive the data, it is considered that the tunnel can communicate normally and does not actively send the tunnel keepalive packets.

Table 18-8 Configure the tunnel connection status detection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the pseudo-wire template configuration mode | **pseudowire-class** *class-name* | - |
| Configure the tunnel connection status detection | **predictive hello** | Optional<br>By default, enable the tunnel connection status detection function. |

**Note:**

- The **no predictive hello** command is used to prohibit detecting the connectivity of the tunnel by the packet.

## 18.2.2. Enable L2TPv3 Function

### Configuration Condition

Before enabling the L2TPv3 function, first complete the following task:

- Configure the L2TPv3 template to enable the normal communication between local pseudo-wire source IP address and the peer pseudo-wire source IP address.

### Enable Dynamic L2TPv3 Session

The dynamic L2TPv3 session negotiated to establish the tunnel and session by the L2TPv3 protocol and negotiates the session-related attribute, including the session ID, cookie value, and pseudo-wire type. When the L2TPv3 function is enabled on the VLAN interface, if Ethernet without VLAN tag is required for the pseudo-wire type, choose the Ethernet parameter option; if Ethernet with VLAN tag is required for the pseudo-wire type, choose the vlan parameter option or do not set the parameter. There are corresponding pseudo-wire types for other types of interfaces.

Table 18-9 Enable the dynamic L2TPv3 session

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the dynamic L2TPv3 session | **xconnect** *peer-ip-address vc-id* **pw-class** *pw-name* [ **vlan** \| **ethernet** ] | Mandatory<br>By default, the dynamic L2TPv3 session on the interface is not enabled. |

**Note:**

- When the L2TPv3 session is enable, a tunnel can be established between the source IP address and the destination IP address. Considering the simple and clear configuration, a destination can only configure a tunnel. Otherwise corresponding error message will be displayed.
- Only when the L2TPv3 function is enabled on the VLAN interface, the Ethernet or vlan parameter can be chose. The VLAN interface is the pseudo-wire of Ethernet with VLAN tag by default.

### Enable Static L2TPv3 Session

Different from the dynamic session, the static session does not use the L2TPv3 protocol to negotiate the session establishment process or the session related attribute. For example, parameters such as the session ID and cookie value must be configured and ensured the correctness when the static session is enabled. That is, the session ID and cookie value at both ends are consistent.

Table 18-10 Enable the static L2TPv3 session

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |

| Step | Command | Description |
|---|---|---|
| Enable the static L2TPv3 session | **xconnect** *peer-ip-address vc-id* **pw-class** *pw-name* **manual** [ **vlan** \| **ethernet** ] | Mandatory<br>By default, the static L2TPv3 session is not enabled on the interface. |
| Configure the session ID at both ends | **l2tp id** *local-id remote-id* | Mandatory<br>By default, the session ID is not configured for the static session. |
| Configure the session cookie at the local end | **l2tp cookie local** { **4** *cookie-value* \| **8** *low-cookie-value high-cookie-value* } | Optional<br>By default, the session cookie is not configured for the local end. |
| Configure the session cookie at the peer end | **l2tp cookie remote** { **4** *cookie-value* \| **8** *low-cookie-value high-cookie-value* } | Optional<br>By default, the session cookie is not configured for the peer end. |

### 18.2.3. L2TPv3 Monitoring and Maintaining

Table 18-11 The L2TPv3 monitoring and maintaining

| Command | Description |
|---|---|
| **clear l2tpv3** [ **session** *vc-id* \| **tunnel** *tunnel-id* ] | Clear all sessions and tunnels and the specified tunnels and sessions |
| **show l2tun detail** | Display the detailed information of all the L2TPv3 session and tunnel |
| **show l2tun session** [ *vc-id* \| **all** \| **brief** \| **circuit** \| **state \| sesid** *ses-id* ] [ **ip-addr** *ip-address* [ **vcid** *vc-id* ] \| **tunnel** { **id** *tunnel-id session-id* \| **remote-name** *remote-name local-name* } \| **vcid** *vc-id* ] | Display the related information of the L2TPv3 session |

| Command | Description |
|---|---|
| **show l2tun tunnel** [ *tunnel-id* \| **all** \| **state** \| **summary** ] [ **id** *tunnel-id* \| **local-name** *local-name remote-name* \| **remote-name** *remote-name local-name* ] | Display the related information of the L2TPv3 tunnel |

## 18.3. L2TPv3 Typical Configuration Example

### 18.3.1. Configure Dynamic Session Established over Ethernet Interface

#### Network Requirements

- The OSPF is used for router advertisement between PE1 and PE2. The dynamic L2TPv3 session is established between PE1 and PE2, enabling the communication between CE1 and CE2.
- The IP addresses of CE1 and CE2 exist in the same network segment.

#### Network Topology

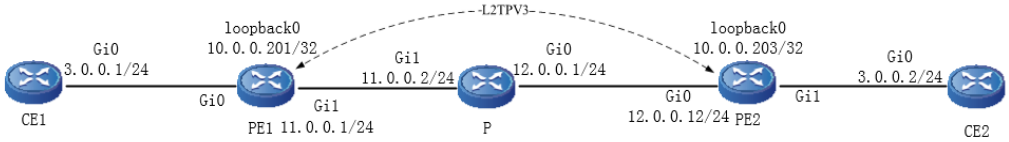



Figure 18-3 Networking of configuring the dynamic session over the Ethernet interface

#### Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the OSPF to enable the route between the loopback interfaces of PE1 and PE2 is reachable.

#Configure PE1.

```
PE1#configure terminal
PE1(config)#router ospf 100
PE1(config-ospf)#network 10.0.0.201 0.0.0.0 area 0
PE1(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
PE1(config-ospf)#exit
```

#Configure P.

```
P#configure terminal
P(config)#router ospf 100
P(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
P(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
P(config-ospf)#exit
```

#Configure PE2.

```
PE2#configure terminal
```

```
PE2(config)#router ospf 100
PE2(config-ospf)#network 10.0.0.203 0.0.0.0 area 0
PE2(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
PE2(config-ospf)#exit
```

#View the global routing table of PE1.

```
PE1#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C   11.0.0.0/24 is directly connected, 00:19:49, gigabitethernet1
O   12.0.0.0/24 [110/2] via 11.0.0.1, 00:18:59, gigabitethernet1
C   127.0.0.0/8 is directly connected, 00:21:05, lo0
C   10.0.0.201/32 is directly connected, 00:19:55, loopback0
O   10.0.0.203/32 [110/3] via 11.0.0.1, 00:18:59, gigabitethernet1
```

It can be observed that PE1 has learnt the route of Loopback0 to PE2.

**Note:**

- P and PE2 are viewed in the same way as PE1. The viewing process is omitted.

**Step 3:** Configure L2TPv3.

#Configure PE1.

Configure the pseudo-wire template as l2tpv3-1 and encapsulated protocol type as L2TPv3. Use Loopback0. Configure L2TPv3 over the gigabitethernet0 interface and use l2tpv3-1 to establish the dynamic L2TPv3 session with PE2.

```
PE1(config)#pseudowire-class l2tpv3-1
PE1(config-pw-class)#encapsulation l2tpv3
PE1(config-pw-class)#protocol l2tpv3
PE1(config-pw-class)#ip local interface loopback0
PE1(config-pw-class)#exit
PE1(config)#interface gigabitethernet 0
PE1(config-if-gigabitethernet0)#xconnect 10.0.0.203 370 pw-class l2tpv3-1
PE1(config-if-gigabitethernet0)#exit
```

#Configure PE2.

Configure the pseudo-wire template as l2tpv3-1 and encapsulated protocol type as L2TPv3. Use Loopback0. Configure L2TPv3 over the gigabitethernet1 interface and use l2tpv3-1 to establish the dynamic L2TPv3 session with PE1.

```
PE2(config)#pseudowire-class l2tpv3-1
```

```
PE2(config-pw-class)#encapsulation l2tpv3
PE2(config-pw-class)#protocol l2tpv3
PE2(config-pw-class)#ip local interface loopback0
PE2(config-pw-class)#exit
PE2(config)#interface gigabitethernet 1
PE2(config-if-gigabitethernet1)#xconnect 10.0.0.201 370 pw-class l2tpv3-1
PE2(config-if-gigabitethernet1)#exit
```

**Note:**

- The VCID of the L2TPv3 session must be consistent at both ends. Otherwise, the dynamic session fails to be established.

**Step 4:** Check the result.

#View the L2TPv3 tunnel information of PE1.

```
PE1#show l2tun detail
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

 TUNNELS:
    LocID  LocName      RemID      RemName      RemAddr      Port   Sess
State
    7      PE1      67         PE2      10.0.0.203     0      1      ESTAB
 SESSIONS:
    LocID  RemID       TunID     State  Type   Vcid        Interface
    1098   1030        7         ESTAB  dynamic 370       gigabitethernet1
```

It can be observed that the L2TPv3 tunnel is successfully established between PE1 and PE2.

#View the L2TPv3 session information of PE1.

```
PE1#show l2tun session
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

Session id 1098 is up, tunnel id 7
Call serial number is 1763798543
Remote tunnel name is PE2
    Internet address is 10.0.0.203
Local tunnel name is PE1
    Internet Address is 10.0.0.201
    Session is L2TP signaled
    Session state is established, time since change 00:09:47
```

```
Session vcid is 370
Session Layer 2 circuit, type is Ethernet, name is gigabitethernet0
Circuit state is UP
    Remote session id is 1030, remote tunnel id 67
DF bit off
Session cookie information:
Ses if_tag_len 0, if_tag NULL
Ses psal bit 1
Ses sync bit 0
Ses batch sync bit 0
```

It can be observed that the dynamic L2TPv3 session is successfully established between PE1 and PE2.

**Note:**

- PE2 is viewed in the same way as PE1. The viewing process is omitted.

#The interface address of gigabitethernet0 of CE2 can be pinged through on PE1.

```
CE1#ping 3.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 3.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

## 18.3.2. Configure Dynamic Session Established over WAN Interface

### Network Requirements

- The OSPF is used for route advertisement between PE1 and PE2. The dynamic L2TPv3 session is established between PE1 and PE2, enabling the communication between CE1 and CE2.
- The IP addresses of CE1 and CE2 exist in the same network segment.

### Network Topology

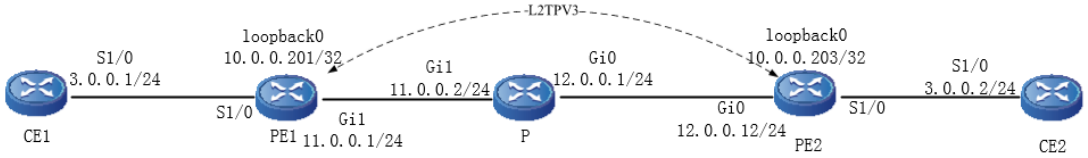



Figure 18-4 Networking of configuring the dynamic session established over the WAN interface

### Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure the OSPF to enable the routing between the loopback interfaces of PE1 and PE2 is reachable.

#Configure PE1.

```
PE1#configure terminal
PE1(config)#router ospf 100
PE1(config-ospf)#network 10.0.0.201 0.0.0.0 area 0
PE1(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
PE1(config-ospf)#exit
```

#Configure P.

```
P#configure terminal
P(config)#router ospf 100
P(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
P(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
P(config-ospf)#exit
```

#Configure PE2.

```
PE2#configure terminal
PE2(config)#router ospf 100
PE2(config-ospf)#network 10.0.0.203 0.0.0.0 area 0
PE2(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
PE2(config-ospf)#exit
```

#View the global routing table of PE1.

```
PE1#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
    D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C   11.0.0.0/24 is directly connected, 00:19:49, gigabitethernet1
O   12.0.0.0/24 [110/2] via 11.0.0.1, 00:18:59, gigabitethernet1
C   127.0.0.0/8 is directly connected, 00:21:05, lo0
C   10.0.0.201/32 is directly connected, 00:19:55, loopback0
O   10.0.0.203/32 [110/3] via 11.0.0.1, 00:18:59, gigabitethernet1
```

It can be observed that PE1 has learnt the routing of Loopback0 to PE2.

**Note:**

- P and PE2 are viewed in the same way as PE1. The viewing process is omitted.

**Step 3:** Configure L2TPv3.

#Configure PE1.

Configure the pseudo-wire template as l2tpv3-1 and the encapsulated protocol type as L2TPv3. Use Loopback0. Configure L2TPv3 on the serial1/0 interface. Use the pseudo-wire template l2tpv3-1 to establish the dynamic L2TPv3 session with PE2.

```
PE1(config)#pseudowire-class l2tpv3-1
PE1(config-pw-class)#encapsulation l2tpv3
PE1(config-pw-class)#protocol l2tpv3
PE1(config-pw-class)#ip local interface loopback0
PE1(config-pw-class)#exit
PE1(config)#interface serial 1/0
PE1(config-if-serial1/0)#encapsulation  hdlc
PE1(config-if-serial1/0)#xconnect 10.0.0.203 380 pw-class l2tpv3-1
PE1(config-if-serial1/0)#exit
```

#Configure PE2.

Configure the pseudo-wire template as l2tpv3-1 and the encapsulated protocol type as L2TPv3. Configure L2TPv3 on the serial1/0 interface. Use the pseudo-wire template l2tpv3-1 to establish the dynamic L2TPv3 session with PE1.

```
PE2(config)#pseudowire-class l2tpv3-1
PE2(config-pw-class)#encapsulation l2tpv3
PE2(config-pw-class)#protocol l2tpv3
PE2(config-pw-class)#ip local interface loopback0
PE2(config-pw-class)#exit
PE2(config)#interface serial 1/0
PE2(config-if-serial1/0)#encapsulation  hdlc
PE2(config-if-serial1/0)#xconnect 10.0.0.201 380 pw-class l2tpv3-1
PE2(config-if-serial1/0)#exit
```

**Step 4:** Check the result.

#View the L2TPv3 tunnel information of PE1.

```
PE1#show l2tun  detail
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

 TUNNELS:
    LocID  LocName      RemID      RemName      RemAddr      Port   Sess
State
    98     PE1      3          PE2       10.0.0.203     0      1      ESTAB
```

```
SESSIONS:
    LocID   RemID       TunID       State   Type    Vcid       Interface
    1045    1095        98          ESTAB   dynamic 380        serial1/0
```

It can be observed that the L2TPv3 tunnel is successfully established between PE1 and PE2.

#View the L2TPv3 session information of PE1.

```
PE1#show l2tun session
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1
Session id 1045 is up, tunnel id 98
Call serial number is 1763799104
Remote tunnel name is PE2
    Internet address is 10.0.0.203
Local tunnel name is PE1
    Internet Address is 10.0.0.201
    Session is L2TP signaled
    Session state is established, time since change 00:00:22
    Session vcid is 380
    Session Layer 2 circuit, type is HDLC, name is serial1/0
    Circuit state is UP
        Remote session id is 1095, remote tunnel id 3
    DF bit off
    Session cookie information:
    Ses if_tag_len 0, if_tag NULL
    Ses psal bit 1
    Ses sync bit 0
    Ses batch sync bit 0
```

It can be observed that the dynamic L2TPv3 session is successfully established between PE1 and PE2.

**Note:**

- PE2 is viewed in the same way as PE1. The viewing process is omitted.

#The serial1/0 interface of CE2 can be pinged through on CE1.

```
CE1#ping 3.0.0.2Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 3.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

## 18.3.3. Configure Static Session Established over Ethernet Sub Interface

### Network Requirements

- The OSPF is used for route advertisement between PE1 and PE2. The L2TPv3 static session is established between PE1 and PE2, enabling the communication between CE1 and CE2.
- The IP addresses of CE1 and CE2 exist in the same network segment.

### Network Topology

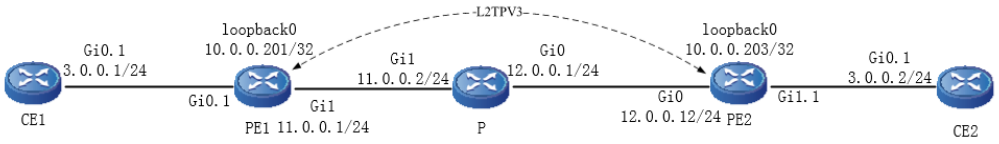



Figure 18-5 Networking of configuring static session established ober Ethernet sub interface

### Configuration Steps

**Step 1:** Configure the IP addresses for all interfaces. (Omitted)

**Step 2:** Configure OSPF to enable the route between the loopback interfaces of PE1 and PE2 is reachable.

#Configure PE1.

```
PE1#configure terminal
PE1(config)#router ospf 100
PE1(config-ospf)#network 10.0.0.201 0.0.0.0 area 0
PE1(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
PE1(config-ospf)#exit
```

#Configure P.

```
P#configure terminal
P(config)#router ospf 100
P(config-ospf)#network 11.0.0.0 0.0.0.255 area 0
P(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
P(config-ospf)#exit
```

#Configure PE2.

```
PE2#configure terminal
PE2(config)#router ospf 100
PE2(config-ospf)#network 10.0.0.203 0.0.0.0 area 0
PE2(config-ospf)#network 12.0.0.0 0.0.0.255 area 0
PE2(config-ospf)#exit
```

#View the global route table of PE1.

```
PE1#show ip route
Codes: C - connected, S - static, R - RIP,  O - OSPF, OE-OSPF External, M - Management
```

```
D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

Gateway of last resort is not set

C   11.0.0.0/24 is directly connected, 00:19:49, gigabitethernet1
O   12.0.0.0/24 [110/2] via 11.0.0.1, 00:18:59, gigabitethernet1
C   127.0.0.0/8 is directly connected, 00:21:05, lo0
C   10.0.0.201/32 is directly connected, 00:19:55, loopback0
O   10.0.0.203/32 [110/3] via 11.0.0.1, 00:18:59, gigabitethernet1
```

It can be observed that PE1 learns the route to the loopback0 interface of PE2.

**Note:**

- The viewing methods of P and PE2 are the same as that of PE1, so the viewing process is omitted.

**Step 3:** Configure L2TPv3.

#Configure PE1.

Configure the pseudo-wire template as l2tpv3-1 and the encapsulated protocol type as L2TPv3. Use Loopback0. Configure L2TPv3 on the gigabitethernet0.1 interface. Use the pseudo-wire template l2tpv3-1 to establish the dynamic L2TPv3 session with PE2.

```
PE1(config)#pseudowire-class l2tpv3-1
PE1(config-pw-class)#encapsulation l2tpv3
PE1(config-pw-class)#protocol none
PE1(config-pw-class)#ip local interface loopback0
PE1(config-pw-class)#exit
PE1(config)#interface gigabitethernet0.1
PE1(config-if-gigabitethernet0.1)#encapsulation dot1q 1
PE1(config-if-gigabitethernet0.1)#xconnect 10.0.0.203 370 pw-class l2tpv3-1 manual
PE1(config-if-xconn)#l2tp id 1 2
PE1(config-if-xconn)#l2tp cookie local 4 1111
PE1(config-if-xconn)#l2tp cookie remote 8 2222 2222
PE1(config-if-xconn)#exit
PE1(config-if-gigabitethernet0.1)#exit
```

#Configure PE2.

Configure the pseudo-wire template as l2tpv3-1 and the encapsulated protocol type as L2TPv3. Use Loopback0. Configure L2TPv3 on the gigabitethernet1.1 interface. Use the pseudo-wire template l2tpv3-1 to establish the dynamic L2TPv3 session with PE2.

```
PE2(config)#pseudowire-class l2tpv3-1
PE2(config-pw-class)#encapsulation l2tpv3
PE2(config-pw-class)#protocol none
```

```
PE2(config-pw-class)#ip local interface loopback0
PE2(config-pw-class)#exit
PE2(config)#interface gigabitethernet1.1
PE2(config-if-gigabitethernet1.1)#encapsulation dot1q 1
PE2(config-if-gigabitethernet1.1)#xconnect 10.0.0.201 370 pw-class l2tpv3-1 manual
PE2(config-if-xconn)#l2tp id 2 1
PE2(config-if-xconn)#l2tp cookie remote 4 1111
PE2(config-if-xconn)#l2tp cookie local 8 2222 2222
PE2(config-if-xconn)#exit
PE1(config-if-gigabitethernet1.1)#exit
```

**Step 4:** Check the result.

#View the L2TPv3 tunnel information of PE1.

```
PE1#show l2tun detail
L2TPv3 established Tunnel 0. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 0  Session 1

 TUNNELS:
    LocID  LocName      RemID       RemName      RemAddr      Port   Sess
State
    NO TUNNELS
 SESSIONS:
    LocID  RemID       TunID       State  Type   Vcid        Interface
    1    1          0          ESTAB  manual 370         gigabitethernet0.1
```

It can be observed that the L2TPv3 static session is successfully established between PE1 and PE2.

#View the L2TPv3 session information of PE1.

```
PE1#show l2tun session
L2TPv3 established Tunnel 0. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 0  Session 1

Session id 1 is up, tunnel id 0
Call serial number is 1763798462
Remote tunnel
    Internet address is 10.0.0.203
Local tunnel
    Internet address is 10.0.0.201
```

Session is manually signalled

Session state is established, time since change 00:55:46

Session vcid is 370

Session Layer 2 circuit, type is Ethernet-Tagged-Mode, name is gigabitethernet0.1, VLAN 1

Circuit state is UP

Remote session id is 2, remote tunnel id 0

DF bit off

Session cookie information:

local cookie, size 4 bytes, value 1111

remote cookie, size 8 bytes, value 2222 2222

Ses if_tag_len 4, if_tag 0x81000006

Ses psal bit 1

Ses sync bit 0

Ses batch sync bit 0

Ses timer real batch send bit 0

It can be observed that the L2TPv3 static session is successfully established between PE1 and PE2.

**Note:**

- The viewing method of PE2 is the same as that of PE1, so the viewing process is omitted.

#On CE1, you can ping Gi0.1 interface address of CE2.

CE1#ping 3.0.0.2

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 3.0.0.2 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.

## 18.3.4. Configure L2TPv3 Over L2TPv2

### Network Requirements

- PE1 acts as LAC, PE2 acts as LNS, set up L2TPv2, and establish the L2TPv3 tunnel.
- CE1 and CE2 are in the same network segment and can communicate normally.

### Network Topology

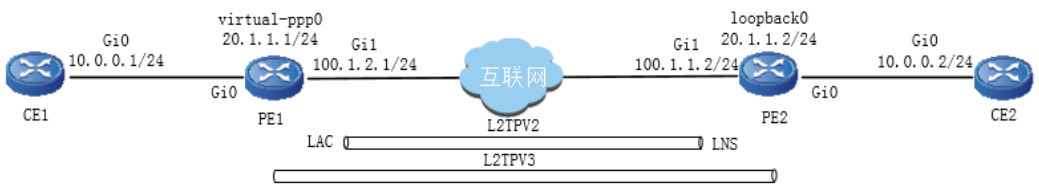



Figure 18-6 Networking of configuring L2TPv3 Over L2TPv2

## Configuration Steps

**Step 1:** Configure the IP address and route of the interfaces, ensuring that PC1 can communicate with PE2 (omitted).

**Step 2:** Configure L2TPv2.

#Configure PE1 as LAC.

Configure the the pseudo-wire template lac.

```
PE1#configure terminal
PE1(config)#pseudowire-class lac
PE1(config-pw-class)#encapsulation l2tpv2
PE1(config-pw-class)#hostname lac
PE1(config-pw-class)#ip local interface gigabitethernet1
PE1(config-pw-class)#password 0 admin
PE1(config-pw-class)#exit
```

Create virtual-ppp interface, configure the authentication user name and password, and reference the pseudo-wire template.

```
PE1(config)#interface virtual-ppp0
PE1(config-if-virtual-ppp0)#ppp chap hostname qtech
PE1(config-if-virtual-ppp0)#ppp chap password qtech123
PE1(config-if-virtual-ppp0)#ip address 20.1.1.1 255.255.255.0
PE1(config-if-virtual-ppp0)#pseudowire 100.1.1.2 1 pw-class lac
PE1(config-pw-class)#exit
```

#Configure PE2 as LNS.

Configure the user name and password.

```
PE2#configure terminal
PE2(config)#local-user qtech class network
PE2(config-user-network-admin@cctv.com)#service-type ppp
PE2(config-user-network-admin@cctv.com)#password 0 qtech123
PE2(config-user-network-admin@cctv.com)#exit
```

Create virtual template Virtual-template0, encapsulate PPP, and borrow the IP address of the Loopback0 interface.

```
PE2(config)#interface virtual-template 0
PE2(config-if-virtual-template0)#encapsulation ppp
PE2(config-if-virtual-template0)#ppp authentication chap
PE2(config-if-virtual-template0)#ip unnumbered loopback0
PE2(config-if-virtual-template0)#exit
```

Enable VPDN, create a VPDN group lns, and configure it to accept dial-in requests. Apply L2TP protocol and borrow Virtual-template0. Only L2TP connection requests with the name lac are accepted. Configure the shared key of LAC and LNS as admin.

```
PE2 (config)#vpdn enable
PE2 (config)#vpdn-group lns
PE2 (config-vpdn)#accept-dialin
PE2 (config-vpdn-acc-in)#protocol l2tp
PE2 (config-vpdn-acc-in)#virtual-template 0
PE2 (config-vpdn-acc-in)#exit
PE2 (config-vpdn)#local name lns
PE2 (config-vpdn)#terminate-from hostname lac
PE2 (config-vpdn)#l2tp tunnel password 0 admin
PE2 (config-vpdn)#exit
```

**Note:**

- The shared keys of LAC and LNS must be consistent. Otherwise, the L2TP tunnel cannot be successfully established.

#View the L2TP information on the PE1 device.

```
PE1#show vpdn detail

 L2tp MaxTun 1000, MaxSes 1000:

 tunnel free num: 999
 TUNNELS:
     LOCAL-ID REM-ID   LOCAL-NAME          REM-NAME          VPDN-GROUP
PORT   SES-CNT  STATE      REM-ADDR
     81      512      lac              Router                    1701   1      ESTABLISHED
100.1.1.2

 session free num: 999
 SESSIONS:
     LOCAL-ID REM-ID  TUN-ID  IF-NAME          SYSTEMID          IMSI/CALLING-
NUM       STATE
     58       249      81       virtual-ppp0          --------          ----------------
ESTABLISHED

 L2tp total Tunnel and Session Information. Tunnel 1  Session 1
```

It can be observed that the L2TP tunnel is set up between PE1 and PE2 sucessfully.

#On the PE1 device, view the global route table.

```
PE1#show ip route
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
```

O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

C 20.1.1.0/24 is directly connected, 00:03:06, virtual-ppp0

L 20.1.1.1/32 is directly connected, 00:03:06, virtual-ppp0

C 20.1.1.2/32 is directly connected, 00:03:06, virtual-ppp0

#On PE1, you can ping the virtual-template0 interface address of PE2.

PE1#ping 20.1.1.2

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 20.1.1.2 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 1/1/2 ms.

**Note:**

- The viewing method of PE2 is the same as that of PE1, so the viewing process is omitted.

**Step 3:** Configure L2TPV3.

#Configure PE1.

Configure the pseudo wire template test, and the encapsulated protocol type is L2TPv3. When the l2tpv2 session is successfully established, virtual-ppp0 is used as the L2TPv3 tunnel interface. Configure L2TPv3 on the interface gigabitethernet0, and use seudo wire template l2tpv3-1 to establish L2TPv3 dynamic session with PE2.

PE1(config)#pseudowire-class test

PE1(config-pw-class)#encapsulation l2tpv3

PE1(config-pw-class)#protocol l2tpv3

PE1(config-pw-class)#ip local interface virtual-ppp0

PE1(config-pw-class)#exit

PE1(config)#interface gigabitethernet 0

PE1(config-if-gigabitethernet0)#xconnect 20.1.1.2 1 pw-class test

PE1(config-if-gigabitethernet0)#exit

#Configure PE2.

Configure the pseudo wire template test, and the encapsulated protocol type is L2TPv3. The L2TPV3 tunnel is established on loopback0 borrowed by the L2TPv2 tunnel. Configure L2TPv3 on the interface gigabitethernet0, and use the pseudo wire template test to establish L2TPv3 dynamic session with PE1.

PE2(config)#pseudowire-class test

PE2(config-pw-class)#encapsulation l2tpv3

PE2(config-pw-class)#protocol l2tpv3

PE2(config-pw-class)#ip local interface loopback0

```
PE2(config-pw-class)#exit
PE2(config)#interface gigabitethernet 0
PE2(config-if-gigabitethernet0)#xconnect 20.1.1.1 1 pw-class test
PE2(config-if-gigabitethernet0)#exit
```

**Note:**

- The VCIDs at the two ends of the L2TPv3 session must be consistent. Otherwise, the dynamic session cannot be set up.

**Step 4:** Check the result.

#View the L2TPv3 tunnel information of PE1.

```
PE1#show l2tun detail
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

 TUNNELS:
    LocID  LocName      RemID       RemName      RemAddr      Port   Sess
State
    25     PE1          312         PE2          20.1.1.2     0      1
    ESTAB
 SESSIONS:
    LocID  RemID        TunID       State  Type   Vcid         Interface
    1115   1551         25          ESTAB  dynamic 1           gigabitethernet0
```

It can be observed that the L2TPv3 tunnel is established between PE1 and PE2.

#View the L2TPv3 session information of PE1.

```
PE1#SHOW l2tun session
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

Session id 1115 is up, tunnel id 25
Call serial number is 910772403
Remote tunnel name is PE2
    Internet address is 20.1.1.2
Local tunnel name is PE1
    Internet Address is 20.1.1.1
    Session is L2TP signaled
    Session state is established, time since change 00:03:06
    Session vcid is 1
    Session Layer 2 circuit, type is Ethernet, name is gigabitethernet0
    Circuit state is UP
```

```
        Remote session id is 1551, remote tunnel id 312
    DF bit off
    Vrf index 0
    Session cookie information:
    Ses if_tag_len 4, if_tag 0x81000002
    Ses psal bit 1
    Ses sync bit 0
    Ses batch sync bit 0
    Ses timer real batch send bit 0
```

It can be observed that the L2TPv3 dynamic session is set up between PE1 and PE2 successfully.

**Note:**

- The viewing method of PE2 is the same as that of PE1, so the viewing process is omitted.

#On CE1, you can ping the gigabitethernet0 interface address of CE2.

```
CE1#ping 10.1.1.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 10.1.1.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

## 18.3.5. Configure L2TPv3 over IPsec

### Network Requirements

- The IPsec tunnel is established between PE1 and PE2, and then L2TPv3 connection is established to protect L2TPv3 data communication between PE1 and PE2 by IPSec.
- Establish L2TPv3 dynamic session between PE1 and PE2, CE1 and CE2 are in the same network segment, and CE1 and CE2 can communicate normally.

### Network Topology

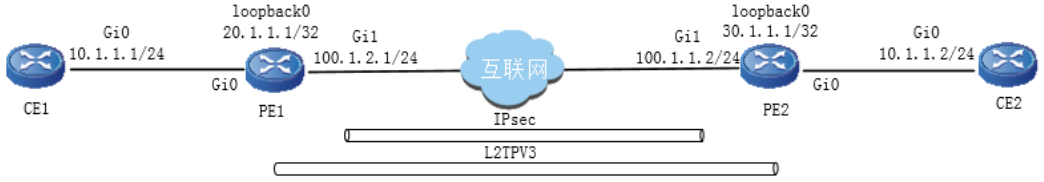



Figure 18-7 Networking of configuring L2TPv3 over IPsec

### Configuration Steps

**Step 1:** Configure the IP address and the route of the interface, ensuring that PE1 can communicate with PE2 (omitted).

**Step 2:** Configure IPSEC.

#On PE1, configure the IKE proposal ikepro, use the encryption algorithm SM1, authentication algorithm SHA1; configure the IPsec proposal ippro, use the ESP security protocol, use the encryption algorithm SM4 and authentication algorithm SHA1.

```
PE1#configure terminal
PE1(config)#crypto ike proposal ikepro
PE1(config-ike-prop)#encryption sm1
PE1(config-ike-prop)#integrity sha1
PE1(config-ike-prop)#exit
PE1(config)#crypto ipsec proposal ippro
PE1(config-ipsec-prop)#esp sm4 sha1
PE1(config-ipsec-prop)#exit
```

#On PE2, configure the IKE proposal ikepro, use the encryption algorithm SM1, authentication algorithm SHA1; configure the IPsec proposal ippro, use the ESP security protocol, use the encryption algorithm SM4 and authentication algorithm SHA1.

```
PE2#configure terminal
PE2(config)#crypto ike proposal ikepro
PE2(config-ike-prop)#encryption sm1
PE2(config-ike-prop)#integrity sha1
PE2(config-ike-prop)#exit
PE2(config)#crypto ipsec proposal ippro
PE2(config-ipsec-prop)#esp sm4 sha1
PE2(config-ipsec-prop)#exit
```

#On PE1, configure the pre-shared key, the key is admin, and permit all peers to use the key.

```
PE1(config)#crypto ike key admin any
```

#On PE2, configure the pre-shared key, the key is admin, and permit all peers to use the key.

```
PE2(config)#crypto ike key admin any
```

#Configure tunnel tun on PE1, use the address of gigabitethernet1 as the local address of the tunnel, configure the peer address of the tunnel as 100.1.1.2, configure the authentication method as pre-shared key authentication, IKE propose to use ikepro, IPSec propose to use ippro, and start auto negotiation.

```
PE1(config)#crypto tunnel tun
PE1(config-tunnel)#local address 100.1.2.1
PE1(config-tunnel)#peer address 100.1.1.2
PE1(config-tunnel)#set authentication preshared
PE1(config-tunnel)#set ike proposal ikepro
PE1(config-tunnel)#set ipsec proposal ippro
PE1(config-tunnel)#set auto-up
PE1(config-tunnel)#exit
```

#Configure tunnel tun on PE2, use the address of gigabitethernet1 as the local address of the tunnel, configure the peer address of the tunnel as any, IKE propose to use ikepro, and IPSec propose to use ippro.

```
PE2(config)#crypto tunnel tun
PE2(config-tunnel)#local address 100.1.1.2
PE2(config-tunnel)#peer any
PE2(config-tunnel)#set ike proposal ikepro
PE2(config-tunnel)#set ipsec proposal ippro
PE2(config-tunnel)#exit
```

#Configure security policy policy1 on PE1 to protect IP communication from network 20.1.1.1/32 to network 30.1.1.1/32 and associate the tunnel tun.

```
PE1(config)#crypto policy policy1
```

```
PE1(config-policy)#flow 20.1.1.1 255.255.255.255 30.1.1.1 255.255.255.255 ip tunnel tun
PE1(config-policy)#set reverse-route
PE1(config-policy)#exit
```

#Configure security policy policy1 on PE2 to protect IP communication from network 30.1.1.1/32 to network 20.1.1.1/32, associate tunnel Tun, and automatically add the route to the peer protection network.

```
PE2(config)#crypto policy policy1
PE2(config-policy)#flow 30.1.1.1 255.255.255.255 20.1.1.1 255.255.255.255 ip tunnel tun
PE2(config-policy)#set reverse-route
PE2(config-policy)#exit
```

#On PE1 device, view the IPsec tunnel.

```
PE1#show crypto ike sa
sa-id   negotiation-state    localaddr      peeraddr       peer-identity
 3      STATE_QUICK_R2       100.1.2.1      100.1.1.2      100.1.1.2
 2      STATE_MAIN_R3        100.1.2.1      100.1.1.2      100.1.1.2
PE1#show crypto ipsec sa
policy name : policy1
 f (src, dst, protocol, src port, dst port) :  20.1.1.1/32  30.1.1.1/32  ip  any  any
    local tunnel endpoint : 100.1.2.1 remote tunnel endpoint : 100.1.1.2, fabric lpu-node : 1
    the pairs of ESP ipsec sa : id : 5, algorithm : SM4 HMAC-SHA1-96
  inbound esp ipsec sa :  spi : 0x354c5d0f(894197007)
    current input 0 packets, 0 kbytes
    encapsulation mode : Tunnel
    replay protection : ON
    remaining lifetime (seconds/kbytes) : 28737/4294967295
    uptime is 0 hour 1 minute 3 second
  outbound esp ipsec sa :  spi : 0xfbb98906(4223240454)
    current output 0 packets, 0 kbytes
    encapsulation mode : Tunnel
    replay protection : ON
    remaining lifetime (seconds/kbytes) : 28737/4294967295
    uptime is 0 hour 1 minute 3 second
total sa and sa group is 1
```

It can be observed that the IPsec tunnel is established between PE1 and PE2 succesfully.

#On the PE1 device, view the route table.

```
PE1#show ip route static
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
```

U - Per-user Static route

O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

S 30.1.1.1/32 [1/0] via 100.1.1.2, 00:04:34, gigabitethernet0/1/1

#On PE1, you can ping the loopback0 interface address of PE2.

```
PE1#ping 30.1.1.1 -s 20.1.1.1

Press key (ctrl + shift + 6) interrupt it.
Reply from 30.1.1.1: bytes = 76 time = 1 ms
Reply from 30.1.1.1: bytes = 76 time = 1 ms
Reply from 30.1.1.1: bytes = 76 time = 1 ms
Reply from 30.1.1.1: bytes = 76 time = 1 ms
Reply from 30.1.1.1: bytes = 76 time = 1 ms
Success rate is 100% (5/5). Round-trip min/avg/max = 1/1/1 ms.
```

**Note:**

- The viewing method of PE2 is the same as that of PE1, so the viewing process is omitted here.

**Step 3:** Configure L2TPv3.

#Configure PE1.

Configure the pseudo wire template test, and the encapsulated protocol type is L2TPv3. Use the loopback0 protected by IPSec as the L2TPv3 tunnel interface. Configure L2TPv3 on the interface gigabitethernet0, and use the pseudo wire template test to establish L2TPv3 dynamic session with PE2.

```
PE1(config)#pseudowire-class test
PE1(config-pw-class)#encapsulation l2tpv3
PE1(config-pw-class)#protocol l2tpv3
PE1(config-pw-class)#ip local interface loopback0
PE1(config-pw-class)#exit
PE1(config)#interface gigabitethernet 0
PE1(config-if-gigabitethernet0)#xconnect 30.1.1.1 1 pw-class test
PE1(config-if-gigabitethernet0)#exit
```

#Configure PE2.

Configure the pseudo wire template test, and the encapsulated protocol type is L2TPv3. Use the loopback0 protected by IPSec as the L2TPv3 tunnel interface. Configure L2TPv3 on the interface gigabitethernet0, and use the pseudo wire template test to establish L2TPv3 dynamic session with PE2.

```
PE2(config)#pseudowire-class test
PE2(config-pw-class)#encapsulation l2tpv3
```

```
PE2(config-pw-class)#protocol l2tpv3
PE2(config-pw-class)#ip local interface loopback0
PE2(config-pw-class)#exit
PE2(config)#interface gigabitethernet 0
PE2(config-if-gigabitethernet0)#xconnect 20.1.1.1 1 pw-class test
PE2(config-if-gigabitethernet0)#exit
```

**Note:**

- The VCIDs at both ends of L2TPv3 session must be consistent. Otherwise, the dynamic session cannot be set up.

**Step 4:** Check the result.

#View the L2TPv3 tunnel information of PE1.

```
PE1#show l2tun detail
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

 TUNNELS:
    LocID  LocName      RemID       RemName      RemAddr      Port   Sess
State
    25     PE1          312         PE2          30.1.1.1     0      1     ESTAB
 SESSIONS:
    LocID  RemID        TunID       State  Type   Vcid         Interface
    1115   1551         25          ESTAB  dynamic 1           gigabitethernet0
```

It can be observed that the L2TPv3 tunnel is established between PE1 and PE2 successfully.

#View the L2TPv3 session information of PE1.

```
PE1#show l2tun session
L2TPv3 established Tunnel 1. established Session 1
L2TPv3 total Tunnel and Session Information. Tunnel 1  Session 1

Session id 1115 is up, tunnel id 25
Call serial number is 910772403
Remote tunnel name is PE2
    Internet address is 30.1.1.1
Local tunnel name is PE1
    Internet Address is 20.1.1.1
    Session is L2TP signaled
    Session state is established, time since change 00:03:06
    Session vcid is 1
```

```
Session Layer 2 circuit, type is Ethernet, name is gigabitethernet0
Circuit state is UP
    Remote session id is 1551, remote tunnel id 312
DF bit off
Vrf index 0
Session cookie information:
Ses if_tag_len 4, if_tag 0x81000002
Ses psal bit 1
Ses sync bit 0
Ses batch sync bit 0
Ses timer real batch send bit 0
```

It can be observed that the L2TPv3 dynamic session is established between PE1 and PE2 successfully.

**Note:**

- The viewing method of PE2 is the same as that of PE1, so the viewing process is omitted.

#On CE1, you can ping the gigabitethernet0 interface address of CE2.

```
CE1#ping 10.1.1.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 10.1.1.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

# 19. NAT

## 19.1. Overview

The NAT (Network Address Translate) translates the IP address of the IP packet header to another IP address. In actual application, a private IP address is translated to a public Ip address. The intranet only needs to use a few public IP addresses to communicate with the Internet. The NAT can relieve the IP address exhaustion problem.

The NAT generally locates at the boundary of the Internet and intranet. The intranet can visit the Internet after the IP address is translated by the NAT. the Internet only knows the translated IP address, but do not know the IP address of the intranet. Therefore, the NAT hide the internal private network, improving the security.

Generally, the NAT supports the IP address mapping relationship of static creating and dynamic generating. Based on the generating methods of the IP address mapping relationship, the IP addresses can be divided into static NAT and dynamic NAT.

- Static NAT

The IP address mapping relationship between the Internet and intranet is determined and statically corresponds with each other. Each intranet IP address corresponds to an Internet IP address.

- Dynamic NAT

The IP address mapping relationship between the intranet and Internet is determined by packets. The dynamic translation rule needs to associate to visit the ACL and the IP address pool. Through the packets filtered in the ACL, an IP address is selected from the IP address pool to create the mapping relationship. When the session of this IP address visiting the Internet ends, the IP address resource in the IP address pool is released to other users.

The NAT locates at the boundary of the Internet and intranet and divided into inside source NAT and outside source NAT.

- Inside source NAT

The inside source translation translates the private IP address of the intranet to the public IP address of the Internet. Generally, the internal IP address adopts the reserved network segment IP address such as 192 and 127. However, compared with the external IP address, these IP addresses do not have corresponding routings. To enable the intranet visits the Internet, the inside source translation is required. The source IP address that the intranet visiting the Internet packet is translated by the NAT to a corresponding Internet IP address.

- Outside source NAT

The outside source translation translates the public IP address of the Internet to the private IP address of the intranet. The source IP address that the Internet visiting the intranet packet is translated by the NAT to a corresponding intranet IP address.

The following describes some NAT-related terms:

IL (Inside Local): indicates the intranet IP address, which will not be publicized to the Internet.

IG (Inside Global): indicates the IP address that is translated from the intranet IP address by the NAT, through which the Internet can visit the intranet devices.

OL (Outside Localind): indicates the Internet IP address allocated to the Internet devices, which will not be publicized to the intranet.

OG (Outside Global): indicates the IP address that is translated from the Internet IP address by the NAT, through which the intranet can visit the Internet devices.

ALG (Application Layer Gateway): mainly handles the application layer protocol packets that are incompatible with the NAT, enabling the application layer data flow can be handled by the NAT normally like the common data flow. The following describes the detailed functions:

- Parse the IP address port information contained in the data packet and translate the IP address as required.
- Extract the data channel information and establish a connection channel for the subsequent packets.
- Detect the application layer status of the packet and drop the status error packet.

Translation table: After the IP data flow is translated by the NAT, a translation table entry is created. The subsequent packets of the data flow are translated by the IP address port information recorded in the translation table entry and all the translation table entries are saved in the translation table.

## 19.2. NAT Function Configuration

Table 19-1 NAT function list

| Configuration Task | |
|---|---|
| Configure the NAT interface | Configure the NAT internal interface |
| | Configure the NAT external interface |
| | Configure NAT internal backflow function |
| Configure the NAT address pool | Configure the NAT address pool |
| | Configure the addresses of the NAT address pool |
| | Configure the port range of the NAT pool addresses |
| | Configure NAT pool address port block size |

| Configuration Task | |
|---|---|
| Configure the static NAT | Configure the inside source NAT static port translation |
| | Configure the inside source NAT static address translation |
| | Configure the outside source NAT static port translation |
| | Configure the outside source NAT static address translation |
| Configure the dynamic NAT | Configure the inside source NAT dynamic port translation |
| | Configure the inside source NAT dynamic address translation |
| | Configure the outside source NAT dynamic address translation |
| Configure the destination NAT | Configure the destination address NAT |
| Configure NAT444 translation | Configure NAT444 static address translation |
| | Configure NAT444 dynamic address translation |
| Configure the NAT ALG function | Configure the TFTP ALG function |
| | Configure the FTP ALG function |
| | Configure the SIP ALG function |
| Configure the NAT logs | Configure the NAT sending logs |

## 19.2.1. Configure NAT Interface

The NAT locates at the boundary of the Internet and intranet. To realize the NAT function, at least an internal interface (connecting to the intranet) and an external interface (connecting to the Internet) need to be configured.

### Configuration Condition

None

### Configure NAT Internal Interface

After the internal interface is configured, the network connected to the NAT internal interface is called intranet.

Table 19-2 Configure the NAT internal interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | Mandatory |
| Configure the NAT internal interface | **ip nat inside** | Mandatory<br>By default, the current interface is not configured as the NAT internal interface. |

### Configure NAT External Interface

After the external interface is configured, the network connecting to the NAT external interfaces is called Internet.

Table 19-3 Configure the NAT external interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | Mandatory |

| Step | Command | Description |
|---|---|---|
| Configure the NAT external interface | **ip nat outside** | Mandatory<br>By default, the current interface is not configured as the NAT external interface. |

### Configure NAT Backflow Function

After configuring the NAT internal interface, you can access the intranet server through the global address of the intranet by configuring the NAT backflow function.

Table 19-1 Configure the NAT backflow function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | Mandatory |
| Configure the NAT internal interace | **ip nat inside** | Mandatory<br>By default, the current interface is not configured as the NAT internal interface. |
| Enable the backflow function on the internal interface | **ip nat hairpin** | Mandatory<br>By default, the current interface does not enable the NAT backflow function. |

**Note:**

The command **ip nat hairpin** can only be configured on the ip nat inside interface, that is, you can only configure the NAT backflow function on the inside interface.

## 19.2.2. Configure NAT Address Pool

### Configuration Condition

None

### Configure the NAT IP address pool

A defined IP address pool needs to be associated when configuring the dynamic NAT rule.

Table 19-4 Configure the NAT IP address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT address pool | **ip nat pool** *pool-name* | Mandatory<br>By default, the NAT address pool is not configured. |

**Note:**

- An address cannot be defined in two different address pools.
- An address pool can only be associated for one time and different NAT rules must associate different IP address pool.
- The port range configured in the same address pool cannot be smaller than the port block size.
- For the address pool bound to the static NAT444 rules, the port range, port block size and address segment must be configured.
- If port range, port block size and address segment are configured in the address pool at the same time, the rules associated with this address pool will automatically become dynamic NAT444 rules.
- The configured port block size and port range take effect only under the rules of static NAT444 translation and dynamic NAT444 translation.

### Configure Addresses of NAT Address Pool

After configuring the address pool, you need to configure the available address of the address pool. When there is a packet matching dynamic rules, you can get a new address from the available address.

Table 19-2 Configure the addresses of the NAT address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT address pool | **ip nat pool** *pool-name* | Mandatory<br>By default, the NAT address pool is not configured. |
| Configure the addresses of the NAT address pool | **address** *start-IP end-IP* [ **match interface** *interface-name* ] | Mandatory<br>By default, the address pool does not have addresses. |

**Note:**

- One address cannot be defined in two different address pools.

### Configure Port Range of NAT Pool Addresses

After the address pool is configured, if you need to configure the port range under the address pool, when associating the PAT rules, the same IP address can apply for different ports to be assigned to different users from the port range.

Table 19-3 Configure the port range of the NAT address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT address pool | **ip nat pool** *pool-name* | Mandatory<br>By default, the NAT address pool is not configured. |
| Configure the port range of the NAT address pool | **port-range** *start-number end-number* | Mandatory<br>By default, do not set the port range. |

**Note:**

- The port range configured in the same address pool cannot be smaller than the port block size.
- The configured port range in the address pool take effect only for static NAT444 and dynamic NAT444.

### Configure Port Block Size of NAT Address Pool

After the address pool is configured, the dynamic rules associated with the address pool are changed into dynamic nat444 rules by configuring the port block size, port range and available addresses.

Table 19-4 Configure the port block size of NAT address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT address pool | **ip nat pool** *pool-name* | Mandatory<br>By default, the NAT address pool is not configured. |
| Configure the port block size of NAT address pool | **port-block block-size** *block-size* **[extended-block-number** *extended-block-number***]** | Optional<br>By default, the address pool is not distributed with port block. |

**Note:**

- Only when the port block size and port range are configured at the same time, the port block will take effect.
- The **extend-block-number** parameter only takes effect under the dynamic NAT444 translation rule.

## 19.2.3. Configure Static NAT

The NAT shields the intranet host. If the intranet needs to provide external services, such as FTP service and web service, static NAT translation rules can play a role. Through configuring the static NAT rule, an internal address can be mapped to a fixed external address, providing external services. The static NAT rule can map the "internal address+port" to the fixed "external address+port", providing external services.

### Configuration Condition

None

### Configure Inside Source NAT Static Port Translation

Internal source NAT static address translation (SAT), which maps "internal address" to "external address".

After the intranet packet is handled by NAT, the source address is converted to the configured external address; after the response packet arrives, the destination address is converted to the corresponding internal address.

Table 19-5 Configure the inside source NAT static port translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the inside source NAT static port translation rule | **ip nat inside source static** { **network** *local-subnet* [**local-vrf** *local-vrf*] *global-subnet netmask* [**global-vrf** *global-vrf*] **\|** *local-IP* [**local-vrf** *local-vrf*] *global-IP* [**global-vrf** *global-vrf*] } | Mandatory<br>By default, the inside source NAT static port translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

### Configure Inside Source NAT Static Address Translation

Internal source NAT static port translation (SPT) maps "internal address + port" to "external address + port".

After the packets of internal network are handled by NAT, the source address and port are converted to the configured external address and port; after the response packet arrives, the destination address and port are converted to the corresponding internal address and port.

Table 19-6 Configure the inside source NAT static address translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the inside source NAT static address translation rule | **nat inside source static { tcp \| udp } {** *local-IP local-port* **[ local-vrf** *local-vrf***] {***global-IP* **\| interface** *interface-name***}** *global-port* **[ global-vrf** *global-vrf* **] \| range** *local-IP local-port-min local-port-max* **[ local-vrf** *local-vrf* **] {** *global-IP* **\| interface** *interface-name* **}** *global-port-min global-port-max* **[ global-vrf** *global-vrf***]}** | Mandatory<br>By default, the inside source NAT static address translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

### Configure Outside Source NAT Static Address Translation

The external source NAT static address translation (OSAT) is similar to SAT, but the translation direction is different. OSAT maps "external address" to "internal address".

After the packet of the external network is handled by NAT, the source address is converted to the configured internal address; after the response packet arrives, the destination address is converted to the corresponding external address.

Table 19-7 Configure the outside source NAT static port translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the outside source NAT static address translation rule | **ip nat outside source static** { **network** *global-subnet* [**global-vrf** *global-vrf*] *local-subnet netmask* [**local-vrf** *local-vrf*] \| *global-IP* [**global-vrf** *global-vrf*] *local-IP* [**local-vrf** *local-vrf*]} | Mandatory<br>By default, the outside source NAT static address translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

### Configure Outside Source NAT Static Port Translation

External source NAT static port conversion rule (OSPT) is similar to SPT function, but the translation direction is different. OSPT rules map external address + port to internal address + port.

After the packet of external network is handled by NAT, the source address and port are converted to the configured internal address and port; the destination address and port are converted to corresponding external addresses and ports after the response packet arrives.

Table 19-8 Configure the outside source NAT static port translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the outside source NAT static port translation rule | **ip nat outside source static** { **tcp** \| **udp** } { *global-IP global-port* [**global-vrf** *global-vrf*] *local-IP local-port* [**local-vrf** *local-vrf* ]\| **range** *global-IP global-port-min global-port-max* [**global-vrf** *global-vrf*] *local-IP local-port-min local-port-max* [**local-vrf** *local-vrf* ]} | Mandatory<br>By default, the outside source NAT static port translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

## 19.2.4. Configure Dynamic NAT

After the NAT dynamic translation rule is configured, the response packet of the intranet visiting the Internet packet can be handled by the NAT. when the reversible parameter of the NAT dynamic translation rule is configured, the Internet is restrictedly allowed to visit the hosts in the intranet. The reversible parameter can be set to two levels: restrict (partially allowed) and full (completely allowed). When the intranet IP address visits the Internet, a mapping relationship of intranet IP address to the Internet IP address is established. Before the mapping relationship breaks, the restrict level allows the Internet IP addresses accessed by the intranet to access the intranet by the mapped external IP address. The full level allows other Internet IP address accesses the intranet by the mapped external address.

### Configuration Condition

Before configuring the dynamic NAT, first complete the following tasks:

- Configure the ACL.
- Configure the NAT IP address pool.

### Configure Inside Source NAT Dynamic Port Translation

The inside source NAT dynamic port address translation (PAT) allows multiple intranet IP addresses mapping to the same Internet IP address. When the PAT rule is configured, a few public IP addresses can satisfy the requirements of intranet visiting the Internet.

The PAT translates the "intranet IP address+port" to "Internet IP address+port". Source IP addresses of packets from different intranet IP address are mapped to the same Internet IP address, but the port numbers are translated to different port numbers of the same IP address. This reserves the difference between packets. When the response packet arrives, the PAT can identify to which Internet IP address the packet should be translated based on the IP address and port number of the response packet.

The mapping relationship between the PAT "intranet IP address+port" and "Internet IP address+port" is generated dynamically. When the session of the intranet IP address visiting the network ends, the corresponding mapping relationship breaks and the IP address in the IP address pool can be used by other users.

Table 19-9Configure the inside source NAT dynamic port translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the inside source NAT dynamic port translation rule | **ip nat inside source list** *access-list-name* [ **local-vrf** *local-vrf* ] { **pool** *pool-name* [**overload**] \| **interface** *interface-name* **overload**} [ **global-vrf** *global-vrf* ] [ **reversible** { **full** \| **restrict** } ] | Mandatory<br>By default, the inside source NAT dynamic address translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

### Configure Outside Source NAT Dynamic Address Translation

The outside source NAT dynamic address translation (ONAT) translates the Internet IP address to the intranet IP address dynamically. The number of intranet IP addresses in the IP address pool is limited. When all the pool addresses establish mapping relationship with the Internet IP addresses, the new NAT translation request will be rejected.

The mapping relationship between the ONAT intranet IP address and Internet IP address is generated dynamically. When the session of the Internet IP address accessing the intranet ends, the corresponding mapping relationship breaks and the IP address in the IP address pool can be used by other users.

The ONAT does not support reversible access.

Table 19-10Configure the outside source NAT dynamic address translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the outside source NAT dynamic address translation rule | **ip nat outside source list** *access-list-name* [**global-vrf** *global-vrf*] **pool** *pool-name* [ **local-vrf** *local-vrf* ] | Mandatory<br>By default, the outside source NAT dynamic address translation rule is not configured. |

**Note:**

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

## 19.2.5. Configure NAT Destination Address Translation

If multiple hosts providing the same service exist in the intranet (multiple web servers with continuous intranet IP addresses for example), simple TCP load sharing can be realized by configuring the SERVER (destination NAT) rule. The SERVER rule maps multiple host IP addresses in the intranet to a virtual IP address. External services are provided through the virtual IP address.

When configuring the ACL associated with the SERVER rule, the IP packet whose destination address is the virtual address is allowed for DAT.

### Configuration Condition

Before configuring the destination NAT, first complete the following tasks:

- Configure the ACL.

### Configure Destination NAT

Table 19-11 Configure the destination NAT

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the destination IP address NAT rule | **ip nat server list** *access-list-name* **[global-vrf** *global-vrf* ] {*local-ip* [ *local-port* ] \| **pool** *pool-name* } [ **local-vrf** *local-vrf* ] | Mandatory<br>By default, the destination NAT rule is not configured. |

**Note:**

- If there is only one host in the intranet, the TCP load sharing is unnecessary. Do not perform this configuration.
- This configuration is only valid for the TCP/UDP packet. If the host provides services of other protocols, do not perform this configuration.
- The destination NAT does not support the ICMP protocol.
- If this rule references an address pool, only the first address segment of the address pool takes effect.
- In the cross VRF scenario of IP network, local VRF is recommended to be configured as the VRF of the inside interface, and global VRF is recommended to be configured as the VRF of the outside interface.

## 19.2.6. **onfigure NAT444 Translation**

### Configuration Condition

Before configuring NAT444 translation, first complete the following task:

- Configure an address pool
- Configure the address of the address pool
- Configure the port range of the address pool
- Configure the port block size of the address pool

### Configure Static NAT444 Translation

The configuration takes effect on all **ip nat outside** interfaces. According to the configuration data in the port block group, a port block is assigned to each private IP address according to a fixed algorithm, and a port block table entry is created. When a private IP address initiates a connection to the public network, it searches the port block table entry through the private IP address, uses the public IP address recorded in the table entry for address translation, and dynamically allocates a port from the corresponding port block for TCP/UDP port translation.

**Note that:**

- Only TCP/UDP/ICMP packet is supported, in which ICMP packet uses ID as port.
- How to allocate port blocks: sort all global IP port blocks from small to large according to IP and from small to large according to port, and search for the corresponding unique port block according to local IP.
- How to allocate ports in a port block: Search for free ports from small to large to assign.
- When the IP connections of a private network exceed the size of the allocated port block, new connection requests cannot be allocated to the port through NAT.
- The public IP port block to which the private network IP is assigned will not cross public IP, that is, if the number of remaining ports of public address 1 is less than the size of port block, the number of ports of port block size will not be allocated to private IP from public address 2.
- Static NAT444 translation does not support initiating the connection from public network to intranet.

Table 19-12 Configure the static NAT444 translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the static NAT444 translation | **ip nat inside source local-ip** *start-ip end-ip* [**local-vrf** *local-vrf* ] **pool** *pool-name* [ **global-vrf** *global-vrf* ] } | Mandatory<br>By default, do not configure the static NAT444 translation. |

**Note:**

- This configuration is only valid for the TCP/UDP/ICMP packet. If the host provides services of other protocols, do not perform this configuration.
- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

### Configure Dynamic NAT444 Translation

The configuration takes effect on all **ip nat outside** interfaces. According to the configuration data in the port block group, a port block is assigned to each private IP address. When a private IP address initiates a connection to the public network, it searches the port block table entry through the private IP address, and dynamically allocates a port from the corresponding port block for TCP/UDP port translation.

**Note that:**

- Only TCP/UDP/ICMP packet is supported, in which ICMP packet uses ID as port.
- How to allocate ports in a port block: Search for free ports from small to large to assign.
- When the IP connections of a private network exceed the size of the allocated port block, new connection requests cannot be allocated to the port through NAT.
- The public IP port block to which the private network IP is assigned will not cross public IP, that is, if the number of remaining ports of public address 1 is less than the size of port block, the number of ports of port block size will not be allocated to private IP from public address 2.
- Static NAT444 translation does not support initiating the connection from public network to intranet.

Table 19-13 Configure the dynamic NAT444 translation

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the dynamic NAT444 translation | **ip nat inside source list** *access-list-name* [ **local-vrf** *local-vrf* ] { **pool** *pool-name* [ **global-vrf** *global-vrf* ] | Mandatory<br>By default, do not configure the dynamic NAT444 translation. |

**Note:**

- This configuration is only valid for the TCP/UDP/ICMP packet. If the host provides services of other protocols, do not perform this configuration.
- When dynamic NAT444 and internal source NAT dynamic translation rules are configured at the same time, and the ACLs referenced by the two rules are effective at the same time, the ACLs associated with dynamic NAT444 rules should be configured with the specific service protocol types supported.

- In the cross VRF scenario of IP network, local-vrf is recommended to be configured as the VRF of the inside interface, and global-vrf is recommended to be configured as the VRF of the outside interface.

## 19.2.7. Configure NAT ALG Function

### Configuration Condition

None

### Configure NAT ALG Function

Configure the NAT ALG function, and you can enable the function of accessing TFTP, FTP and SIP server through NAT device.

Table 19-14 Configure NAT ALG

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the FTP ALG function | **ip nat alg ftp** | Optional<br>By default, do not enable the ftp alg function. |
| Configure the TFTP ALG function | **ip nat alg tftp** | Optional<br>By default, do not enable the tftp alg function. |
| Configure the SIP ALG function | **ip nat alg sip** | Optional<br>By default, do not enable the sip alg function. |

## 19.2.8. Configure NAT Logs

### Configuration Condition

Before configuring NAT to send logs, first complete the following task:

- Configure the log server

### Configure NAT Logs

There are two main types of NAT logs:

1. NAT translation table entries, used to track and record users' network access related information;
2. The allocation information of NAT444 port block is used to record the port block used by private network IP. By sending these records to the external log server in the form of log messages, the network access behavior of the user can be traced through these records when necessary.

After configuring the NAT send log function, NAT sends the log to the log server.

Table 19-15 Configure NAT to send logs

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Send logs when creating the NAT table entry | **ip nat logging security-data connection-begin** | Optional<br>By default, do not send logs when creating NAT table entries. |
| Send logs when the NAT entries are aged | **ip nat logging security-data connection-end** | Optional<br>By default, do not send logs when NAT entries are aged. |
| Send logs when assigning the NAT port blocks | **ip nat logging security-data port-block–assign** | Optional<br>By default, do not send logs when assigning NAT port blocks. |
| Send logs when recycling the NAT port block | **ip nat logging security-data port-block-withdraw** | Optional<br>By default, do not send logs when recycling the NAT port blocks. |

## 19.2.9. **NAT Monitoring and Maintaining**

Table 19-16 The NAT monitoring and maintaining

| Command | Description |
|---|---|
| **clear ip nat statistics** | Clear the NAT statistics data |
| **show running-config ip nat** | Display the NAT part in the configuration file |
| **show ip nat translation** | Display the NAT translation table |

| Command | Description |
|---|---|
| **show ip nat pool** | Display all NAT address pools |
| **show ip nat port-block** | Display the NAT444 port block |
| **show ip nat rule** | Display all NAT rules |
| **show ip nat statistics** | Display the NAT statistics data |

## 19.3. NAT Typical Configuration Example

### 19.3.1. Configure Inside Source NAT Static Address/Port Translation

#### Network Requirements

- PC1 and PC2 are hosts in the internal private network, Web Server is the web server in the public network, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the static IP address/port translation using the inside source NAT.
- PC1 in the internal private network can successfully visit Web Server and the source IP address of the packet is translated from the private IP address 1.1.1.2 to the public IP address 201.16.16.2. PC2 can successfully visit Web Server through the source port 1024. The source IP address of the packet is translated from the private IP address 1.1.1.3 to the public IP address 201.16.16.2 and the source port 1024 is translated to 2048.

#### Network Topology

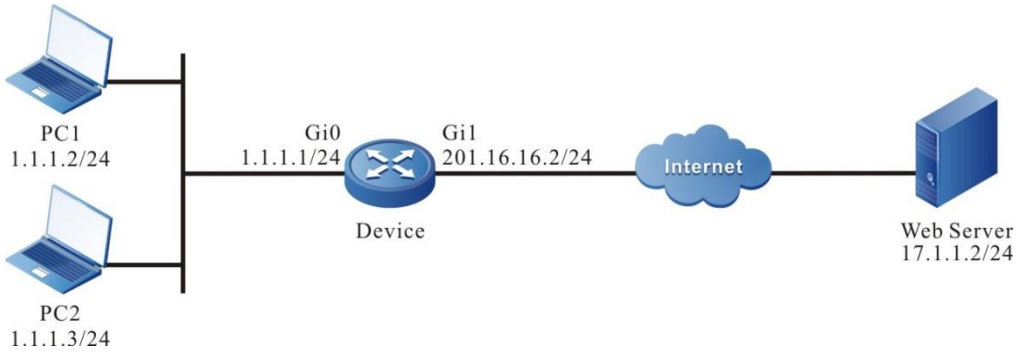



Figure 19-5 Networking of configuring the inside source NAT static address/port translation

#### Configuration Steps

**Step 1:** Configure the IP addresses and routes for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure the inside interface with gigabitethernet0 as the NAT.

```
Device#configure terminal
```

```
Device(config)#interface gigabitethernet0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure the outside interface with gigabitethernet1 as the NAT.

```
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the inside source NAT static address/port translation rule, the internal private IP addresses 1.1.1.2 and 1.1.1.3 are translated to the public IP address 201.16.16.2 and the source port 1024 of 1.1.1.3 is translated to 2048.

```
Device(config)#ip nat inside source static 1.1.1.2 201.16.16.2
Device(config)#ip nat inside source static tcp 1.1.1.3 1024 201.16.16.2 2048
```

**Step 3:** Check the result.

#When PC1 of the internal private network accesses the Web Server, it can see the NAT translation table entry of type SAT on the device, which converts the internal local address 1.1.1.2 to the internal global address 201.16.16.2. When PC2 uses the source port 1024 to access the web server, it can see the NAT translation table entry of type SPT on the device, converting the internal local address 1.1.1.3 to the internal global address 201.16.16.2, and converting the source port 1024 to 2048.

```
Device#show ip nat translation
 Type          Pro    Inside global:port      Inside local:port       Outside local:port
Outside global:port     lifetime
 -------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
 SAT           TCP    [201.16.16.2]:1587      [1.1.1.2]:1587          [17.1.1.2]:80          [17.1.1.2]:80
3600
 SPT           TCP    [201.16.16.2]:2048      [1.1.1.3]:1024          [17.1.1.2]:80          [17.1.1.2]:80
3600


 Valid/Total: 2/2
```

## 19.3.2. **Configure NAT444 Static Port Block Translation**

### Network Requirements

- PC1 and PC2 are hosts in the internal private network, Web Server is the web server in the public network, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the NAT444 static port block translation.
- All users in the 1.1.1.0/24 network segment of the internal private network can successfully access the web server. By searching for the static port block table entry,

the public IP address 201.16.16.3 recorded in the table entry is used for source address translation, and a port is dynamically allocated from the corresponding port block for port translation.

## Network Topology

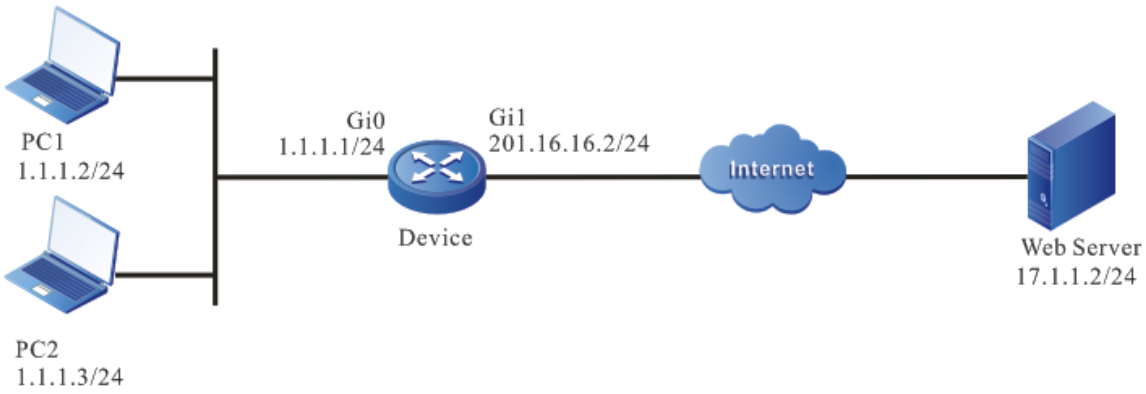



Figure 19-17 Networking of configuring NAT444 static port block translation

## Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT interface attribute and NAT translation rules of Device.

#Configure interface gigabitethernet0 as the inside interface of NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as outside interface of NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Create NAT address pool pool1.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#port-range 1 1000
Device(config-nat-pool)#port-block block-size 500
Device(config-nat-pool)#address 201.16.16.3 201.16.16.5
Device(config-nat-pool)#exit
```

#Configure NAT444 static port block mapping rules to convert internal private network address 1.1.1.2 and 1.1.1.3 to public network address 201.16.16.3.

```
Device (config)#ip nat inside source local-ip 1.1.1.2 1.1.1.3 pool pool1
```

#View the distribution of the static port block.

```
Device#show ip nat port-block static
```

```
No.   Pool name                 Local IP       Global IP      Port block   Connections
----- ------------------------- -------------- -------------- ------------ -----------
1     pool1                     1.1.1.2        201.16.16.3    1-500        0
2     pool1                     1.1.1.3        201.16.16.3    501-1000     0
```

#View the address pool.

```
Device#show ip nat pool
No.  Type     Pool name                 Pool index  Version  Start IP       End IP
Interface                 Interface index
---- -------- ------------------------- ----------- -------- -------------- --------------
-- -------------- -------------------------------- --------------
1    SNAT444  pool1                     0           12       201.16.16.3    201.16.16.5    N/A
N/A
```

**Step 3:** Check the result.

#When PC1 and PC2 of the internal private network access the web server, you can see the NAT translation table entry of type SNAT444 on the device, converting the internal local address 1.1.1.2 of PC1 to the internal global address 201.16.16.3, source port 1024 to 1, the internal local address 1.1.1.3 of PC2 to the global address 201.16.16.3, and source port 1024 to 501.

```
Device#show ip nat translation
Type          Pro    Inside global:port     Inside local:port      Outside local:port
Outside global:port    lifetime
------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
SNAT444       TCP    [201.16.16.3]:501      [1.1.1.3]:1024         [17.1.1.2]:80          [17.1.1.2]:80
3600
SNAT444       TCP    [201.16.16.3]:1        [1.1.1.2]:1024         [17.1.1.2]:80          [17.1.1.2]:80
3600


Valid/Total: 2/2
```

**Note:**

- When there are many intranet hosts accessing the Internet service, after the port segment of one address is distributed, start to distribute and convert from the next address.

## 19.3.3. Configure Inside Source NAT Dynamic Address Translation

### Network Requirements

- PC1 and PC2 are hosts in the internal private network, Web Server is the web server in the public network, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the inside source NAT dynamic address translation.

- The users of network segment 1.1.1.0/24 in the internal private network can successfully access Web Server. The used public network addresses are 201.16.16.2 and 201.16.16.3.

## Network Topology

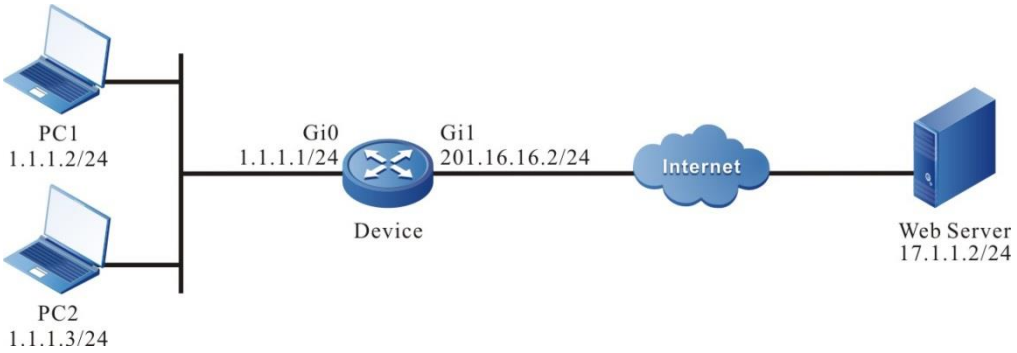



Figure 19-6 Networking of configuring the inside source NAT dynamic address translation

## Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure the inside interface with gigabitethernet0 as the NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure the outside interface with gigabitethernet1 as the NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT IP address pool, pool1, containing two public network IP addresses 201.16.16.2 and 201.16.16.3.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#address 201.16.16.2 201.16.16.3
Device(config-nat-pool)#exit
```

#Configure the ACL 1001, only permitting PCs in the internal private network segment 1.1.1.0/24 to access Web Server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 1.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure the inside source NAT dynamic address translation rule, translating the internal IP addresses 1.1.1.2 and 1.1.1.3 as the public IP addresses 201.16.16.2 and 201.16.16.3, respectively.

```
Device(config)#ip nat inside source list 1001 pool pool1
```

#View the address pool.

```
Device#show ip nat  pool
 No.  Type     Pool name                      Pool index  Version  Start IP       End IP
Interface                 Interface index
 --- ------- ------------------------------ ---------- ------- -------------
-- --------------- -------------------------------- ---------------
 1    NAT      pool1                    0         12       201.16.16.2      201.16.16.3      N/A
N/A
```

**Step 3:** Check the result.

#When PC1 and PC2 of the internal private network access the web server, you can see the NAT translation table entry on Device, converting the internal local address 1.1.1.2 of PC1 to the internal global address 201.16.16.2, and the internal local address 1.1.1.3 of PC2 to the global address 201.16.16.3.

```
Device#show ip nat translation
 Type          Pro     Inside global:port      Inside local:port       Outside local:port
Outside global:port     lifetime
 -------------- ------ ---------------------- ---------------------- ------
----------------- ---------------------- --------
 NAT           TCP     [201.16.16.3]:1024      [1.1.1.3]:1024          [17.1.1.2]:80          [17.1.1.2]:80
3600
 NAT           TCP     [201.16.16.2]:1024      [1.1.1.2]:1024          [17.1.1.2]:80          [17.1.1.2]:80
3600


 Valid/Total: 2/2
```

## 19.3.4. **Configure Inside Source NAT Dynamic Port Translation**

### Network Requirements

- PC1 and PC2 are hosts in the internal private network, Web Server is the web server in the public network, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the inside source NAT dynamic port translation.
- The users of the network segment 1.1.1.0/24 in the internal private network can successfully access Web Server, map the TCP port number to multiplex the same public network address, and the public network address used is 201.16.16.2.

## Network Topology

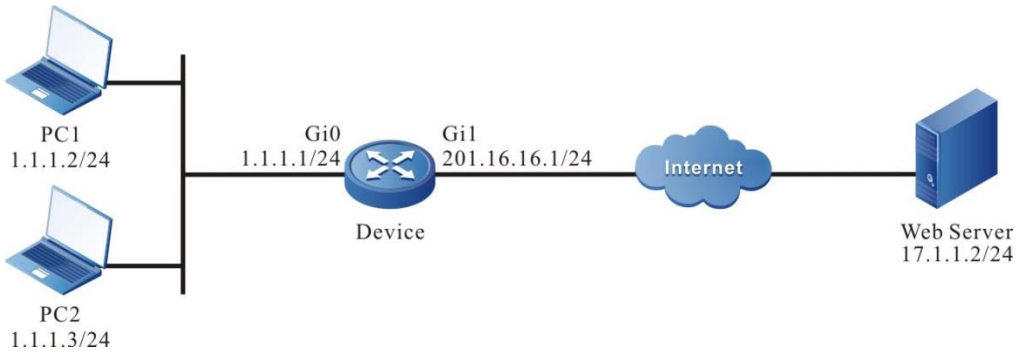



Figure 19-7 Networking of configuring the inside source NAT dynamic port translation

## Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure the inside interface with gigabitethernet0 as the NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure the outside interface with gigabitethernet1 as the NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if- gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT IP address pool, pool1, containing only one public network IP address 201.16.16.2.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#address 201.16.16.2 201.16.16.2
Device(config-nat-pool)#exit
```

#Configure the ACL 1001, only permitting PCs in the internal private network segment 1.1.1.0/24 to access Web Server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 1.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure the PAT translation rule, translating the internal IP addresses 1.1.1.2 and 1.1.1.3 as the public IP addresses 201.16.16.1.

```
Device(config)#ip nat inside source list 1001 pool pool1 overload
```

# View the address pool.

```
Device#show ip nat  pool
 No.  Type     Pool name                      Pool index  Version  Start IP       End IP
Interface                 Interface index
 ---  -------  -----------------------------  ----------  -------  -------------
-- ---------------  --------------------------------  ---------------
 1    PAT      pool1                          0           12       201.16.16.2    201.16.16.2    N/A
N/A
```

**Step 3:** Check the result.

#When PC1 and PC2 of the internal private network access the web server, you can see the NAT translation table entry of type PAT on the device, converting the internal local address 1.1.1.2 of PC1 to the internal global address 201.16.16.2, source port 1024 to 10001, the internal local address 1.1.1.3 of PC2 to the internal global address 201.16.16.2, and source port 1024 to 10002.

```
Device#show ip nat translation
 Type          Pro    Inside global:port     Inside local:port      Outside local:port
Outside global:port    lifetime
 -------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
 PAT           TCP    [201.16.16.2]:10002    [1.1.1.3]:1024         [17.1.1.2]:80          [17.1.1.2]:80
3600
 PAT           TCP    [201.16.16.2]:10001    [1.1.1.2]:1024         [17.1.1.2]:80          [17.1.1.2]:80
3600


 Valid/Total: 2/2
```

## 19.3.5. Configure Intranet Users to Access Extranet via Multiple Exports

### Network Requirements

- PC1 is the internal private network host, web server is the web server on the public network, Device is the NAT device, gigabitethernet0 is the internal interface of NAT, and gigabitethernet1 and gigabitethernet2 are the external interfaces of NAT. The internal source NAT dynamic port translation is configured, and the address pool referenced by NAT rules uses the mode of matching the out interface.
- All users in the 1.1.1.0/24 network segment of the internal private network can successfully access the web server. When they go out from the outgoing interface of the telecom operator, they use the public address 201.16.16.5 for source address translation. When they go out from the outgoing interface of the Unicom operator, they use the public address 202.16.16.5 for source address translation.

## Network Topology

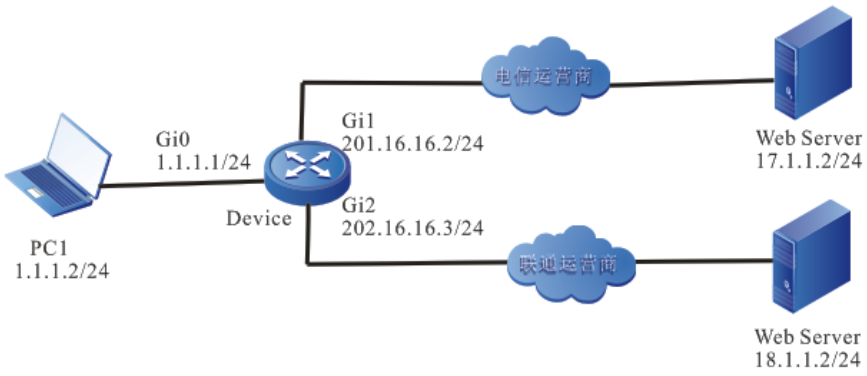



Figure 19-18 Networking of configuring intranet users to access extranet via multiple exports

## Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure interface gigabitethernet0 as the inside interface of NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1as the outsize interface of NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure interface gigabitethernet2as the outsize interface of NAT.

```
Device(config)#interface gigabitethernet 2
Device(config-if-gigabitethernet2)#ip nat outside
Device(config-if-gigabitethernet2)#exit
```

#Configure the NAT address pool pool1, and use the public address 201.16.16.5 when the outgoing interface is gigabitethernet1, and use the public address 202.16.16.5 when the outgoing interface is gigabitethernet2.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#address 201.16.16.5 201.16.16.5 match interface gigabitethernet1
Device(config-nat-pool)#address 202.16.16.5 202.16.16.5 match interface gigabitethernet2
Device(config-nat-pool)#exit
```

#Configure the ACL 1001, only permitting the PC of the network segment 1.1.1.0/24 in the private network to access Web Server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 1.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure the PAT translation rules. When the internal private network address 1.1.1.2 accesses the public network address and goes out from the interface gigabitethernet1, the source address is converted to the public network address 201.16.16.5. When the internal private network address 1.1.1.2 accesses the public network address and goes out from the interface gigabitethernet2, the source address is converted to the public network address 202.16.16.5.

```
Device(config)#ip nat inside source list 1001 pool pool1 overload
```

**Step 3:** Check the result.

#When the internal private network PC1 accesses Web Server 17.1.1.2, you can see the NAT translation table entry of PAT type on the device, which converts the internal local address 1.1.1.2 of PC1 to the internal global address 201.16.16.5, and the source port 1024 to 10001. When PC1 accesses web server 18.1.1.2, the internal local address 1.1.1.2 of PC1 is converted to the internal global address 202.16.16.5, and the source port 1024 is converted to 10001.

```
Device#show ip nat translation
 Type          Pro    Inside global:port      Inside local:port       Outside local:port
Outside global:port     lifetime
 -------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
 PAT           TCP    [201.16.16.5]:10001     [1.1.1.2]:1024         [17.1.1.2]:80          [17.1.1.2]:80
3600
 PAT           TCP    [202.16.16.5]:10001     [1.1.1.2]:1024         [18.1.1.2]:80          [18.1.1.2]:80
3600


 Valid/Total: 2/2
```

## 19.3.6. Configure Across-vrf Internal Source NAT Dynamic Port Translation

### Network Requirements

- PC1 and PC2 are internal private network hosts, web server is the web server of the public network, Device is the NAT device, gigabitethernet0 is NAT internal interface, in vrf a, gigabitethernet1 is NAT external interface, and in vrf b, configure internal source NAT dynamic port translation.
- All users in the 1.1.1.0/24 network segment of the internal private network can successfully access the web server and multiplex the same public network address by mapping the port number of TCP. The public network address used is 201.16.16.2.

## Network Topology

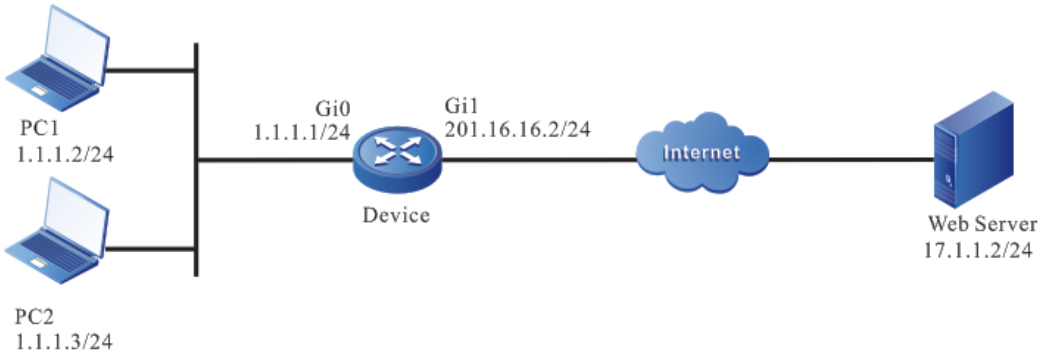



Figure 19-19 Networking of configuring the across-vrf internal source NAT dynamic port translation

## Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure vrf a and vrf b, and then, add the ingress port and egress port to the two vrfs respectively.

```
Device#configure terminal
Device(config)#ip vrf a
Device(config-vrf)#rd 1:1
Device(config-vrf)#exit

Device(config)#ip vrf b
Device(config-vrf)#rd 2:2
Device(config-vrf)#exit

Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip vrf forwarding a
Device(config-if-gigabitethernet0)#ip address 1.1.1.1 24
Device(config-if-gigabitethernet0)#exit

Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip vrf forwarding b
Device(config-if-gigabitethernet1)#ip address 201.16.16.2 24
Device(config-if-gigabitethernet1)#exit
```

#Configure interface gigabitethernet0 as the inside interface of NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
```

```
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure NAT address pool pool1, only containing one public network address 201.16.16.2.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#address 201.16.16.2 201.16.16.2
Device(config-nat-pool)#exit
```

#Configure access control list 1001, only permitting the PCs of the segment 1.1.1.0/24 in the internal private network to access web server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 1.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure the across-vrf PAT translation rules, converting the internal private network addresses 1.1.1.2 and 1.1.1.3 to public address 201.16.16.1.

```
Device(config)#ip nat inside source list 1001 local-vrf a pool pool1 overload global-vrf b
```

#Configure the route from vrf a to vrf b.

```
Device(config)#ip route vrf a 17.1.1.0 24 gigabitethernet 1
```

**Step 3:** Check the result.

#When the internal private networks PC1 and PC2 access the web server, they can see the NAT translation table entry of PAT type on Device, which converts the internal local address 1.1.1.2 of PC1 to the internal global address 201.16.16.2, and the source port 1024 to 10001. The internal local address 1.1.1.3 of PC2 is converted to the internal global address 201.16.16.2, and the source port 1024 is converted to 10002.

```
Device#show ip nat translation
 Type        Pro  Inside global:port     Inside local:port      Outside local:port
Outside global:port     lifetime
 -------------- ------ ---------------------- ----------------------
---------------------- ---------------------- --------
 PAT            TCP      [201.16.16.2]:10002      [1.1.1.3]:1024          [17.1.1.2]:80
[17.1.1.2]:80         3600
 PAT            TCP      [201.16.16.2]:10001      [1.1.1.2]:1024          [17.1.1.2]:80
[17.1.1.2]:80         3600


 Valid/Total: 2/2
```

**Note:**

- When the inside or outside interface is under VRF, the corresponding local-vrf and global-vrf must be carried when configuring NAT rules. The VRF name can be the actual VRF or any. When VRF is any, it means any VRF. By default, the corresponding VRF is global VRF.

## 19.3.7. Configure NAT444 Dynamic Port Block Translation

### Network Requirements

- PC1 and PC2 are internal private network hosts, web server is the web server on the public network, Device is the NAT device, gigabitethernet0 is NAT internal interface, gigabitethernet1 is NAT external interface, and configure NAT444 dynamic port block translation.
- All users in the 1.1.1.0/24 network segment of the internal private network can successfully access the web server. According to the configuration of the corresponding dynamic port block, the public IP address 201.16.16.5 and the port block are allocated, and a port is dynamically allocated from the port block for port translation.

### Network Topology

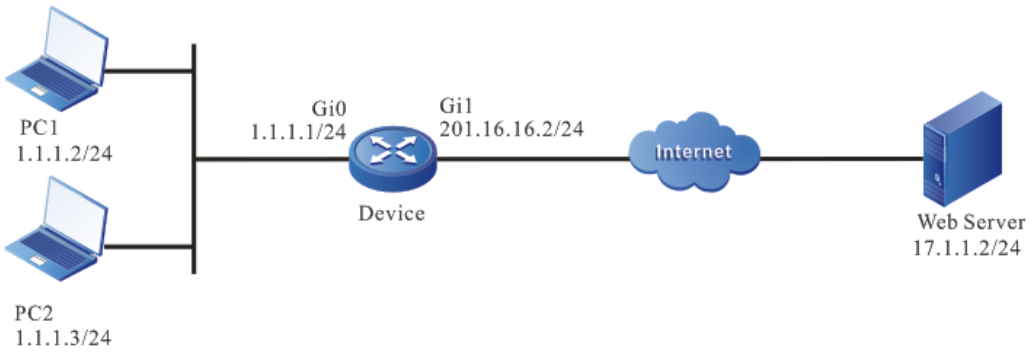



Figure 19-20 Networking of configuring NAT444 dynamic port block translation

### Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure interface gigabitethernet0 as the inside interface of NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Create NAT address pool pool1.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#port-range 2001 3500
Device(config-nat-pool)#port-block block-size 500
Device(config-nat-pool)#address 201.16.16.5 201.16.16.7
Device(config-nat-pool)#exit
```

#Configure access control list 1001, only permitting the PCs in 1.1.1.0/24 segment of internal private network to access web server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 1.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure NAT444 dynamic port block translation rules, converting internal private network addresses 1.1.1.2 and 1.1.1.3 to public network address 201.16.16.5.

```
Device(config)#ip nat inside source list 1001 pool pool1
```

#View the address pool.

```
Device#show ip nat  pool
 No.  Type     Pool name                    Pool index  Version  Start IP       End IP
Interface                 Interface index
 ---- -------- ---------------------------- ---------- ------- -------------
-- --------------- -------------------------------- ---------------
 1    DNAT444  pool1                        0          12       201.16.16.5     201.16.16.7     N/A
N/A
```

**Step 3:** Check the result.

#When the internal private networks PC1 and PC2 access the web server, you can see the NAT translation table entry of type DNAT444 on Device. The internal local address 1.1.1.2 of PC1 is converted to the internal global address 201.16.16.5, and the source port 1024 is converted to 2001. The internal local address 1.1.1.3 of PC2 is converted to the internal global address 201.16.16.5, and the source port 1024 is converted to 2501.

```
Device#show ip nat translation
 Type          Pro    Inside global:port      Inside local:port       Outside local:port
Outside global:port     lifetime
 ------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
 DNAT444       TCP    [201.16.16.5]:2501      [1.1.1.3]:1024          [17.1.1.2]:80           [17.1.1.2]:80
3599
 DNAT444       TCP    [201.16.16.5]:2001      [1.1.1.2]:1024          [17.1.1.2]:80           [17.1.1.2]:80
3599


 Valid/Total: 2/2
```

#View the distribution of the dynamic port block.

```
Device#show ip nat port-block dynamic
 No.  Pool name                  Local IP      Global IP      Port block   Connections
 ----- ------------------------------ --------------- --------------- ------
-----  -----------
 1    pool1                  1.1.1.2       201.16.16.5    2001-2500    1
 2    pool1                  1.1.1.3       201.16.16.5    2501-3000    1
```

## 19.3.8. Configure NAT Internal Server Translation

### Network Requirements

- Web server 1 and web server 2 are internal web servers and provide external web services. PC is the external host, Device is the NAT device, gigabitethernet0 is the internal interface of NAT, and gigabitethernet1 is the external interface of NAT.
- When the external public network host PC accesses the public network address 201.16.16.2, NAT will balance the load to the internal web server 1 and web server 2.

### Network Topology

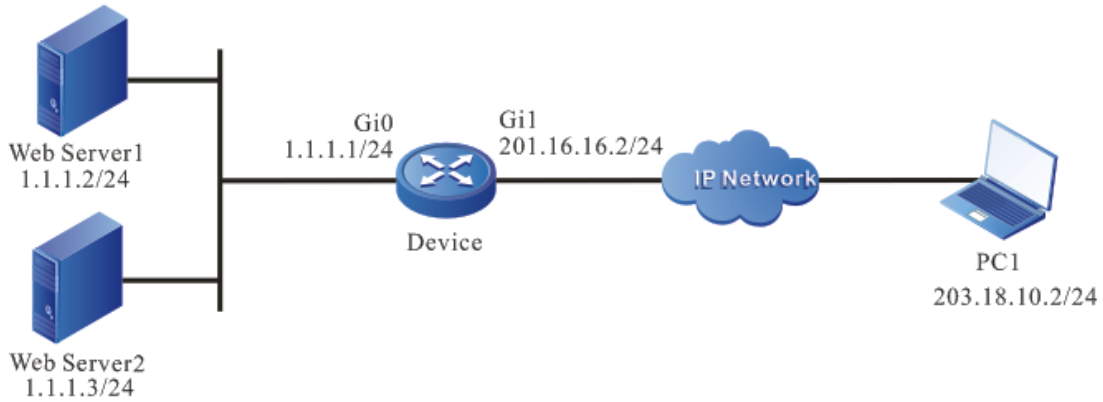



Figure 19-21 Networking of configuring NAT internal server translation

### Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure interface gigabitethernet0 as the inside interface of NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT address pool pool1, which contains two private network addresses 1.1.1.2 and 1.1.1.3, so that TCP flows can be evenly allocated to these two addresses.

```
Device(config)#ip nat pool pool1
Device(config-nat-pool)#address 1.1.1.2 1.1.1.3
Device(config-nat-pool)#exit
```

#Configure access control list 1001, only the PC of port 80 of the address 201.16.16.2 can access internal web server.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit tcp any host 201.16.16.2 eq 80
Device(config-ext-nacl)#exit
```

#Configure NAT internal server conversion rules to convert destination address 201.16.16.2 to internal private network addresses 1.1.1.2 and 1.1.1.3.

```
Device(config)# ip nat server list 1001 pool pool1
```

#View the address pool.

```
Device#show ip nat pool
 No.  Type    Pool name                    Pool index  Version  Start IP       End IP
Interface                 Interface index
 ---- ------- ---------------------------- ---------- ------- -------------
-- --------------- ------------------------------ --------------
 1    SERVER  pool1                        0          13       1.1.1.2        1.1.1.3        N/A
N/A
```

**Step 3:** Check the result.

#When the external public network PC accesses port 80 of the public network address 201.16.16.2, you can see the NAT translation table entry of type SERVER on Device, and map the load balancing of the destination address 201.16.16.2 of the external network PC to the internal network addresses 1.1.1.2 and 1.1.1.3.

```
Device#show ip nat translation
 Type          Pro    Inside global:port      Inside local:port       Outside local:port
Outside global:port    lifetime
 ------------- ------ ---------------------- ---------------------- ------
---------------- ---------------------- --------
 SERVER        TCP    [201.16.16.2]:80        [1.1.1.3]:80            [203.18.10.2]:1025
[203.18.10.2]:1025      3599
 SERVER        TCP    [201.16.16.2]:80        [1.1.1.2]:80            [203.18.10.2]:1024
[203.18.10.2]:1024      3599


 Valid/Total: 2/2
```

**Note:**

- When there are multiple discrete address segments in the address pool associated with NAT internal server rules, only the first address segment can be used for load balancing.
- NAT internal server rules can be associated with address pools or specific IP or IP + ports.

## 19.3.9. Configure Outside Source NAT Static Address/Port Translation

### Network Requirements

- PC1 is the internal host, PC2 and PC3 are the external hosts, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the outside source NAT static address/port translation.
- The external host PC2 can successfully visit PC1. The source IP address of the packet is translated from the Internet IP address 17.1.1.2 to the intranet IP address 110.10.10.1. PC3 can successfully visit PC1 through the source port 1024. The source IP address of the packet is translated from the Internet IP address 17.1.1.3 to the intranet IP address 110.10.10.1 and the source port 1024 is translated to 2048.

### Network Topology

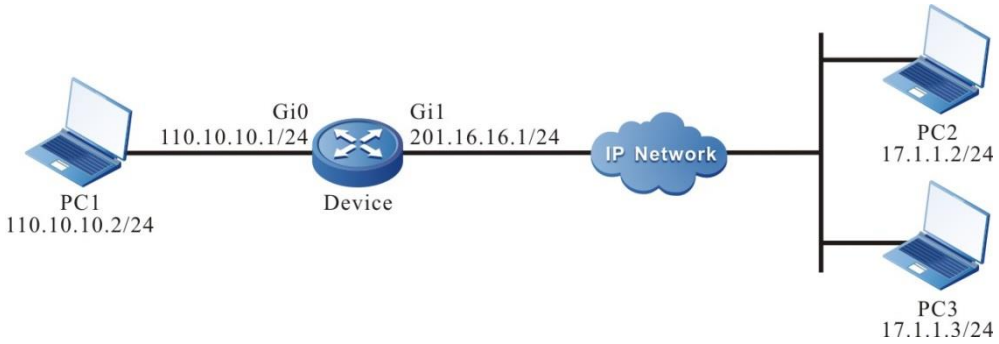



Figure 19-8 Networking of configuring outside source NAT static address/port translation

### Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure the inside interface with gigabitethernet0 as the NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure the outside interface with gigabitethernet1 as the NAT.

```
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the outside source NAT static address/port translation rule, translating the Internet IP address 17.1.1.2 and 17.1.1.3 to the intranet IP address 110.10.10.1 and the source port 1024 of 17.1.1.3is translated to 2048.

```
Device(config)#ip nat outside source static 17.1.1.2 110.10.10.1
Device(config)#ip nat outside source static tcp 17.1.1.3 1024 110.10.10.1 2048
```

**Step 3:** Check the result.

# When PC2 visits PC1, the NAT table entry with the type setting to OSAT can be viewed on Device. When PC3 visits PC1, the NAT table entry with the type setting to OSPT can be viewed on Device.

```
Device#show ip nat translations static
 Type  Pro      Inside Global       Inside Local       Outside Local      Outside Global
----- ---- -------------------- -------------------- -------------------- ---
-----------------
 OSAT (--)       (----------)       (----------)     110.10.10.1          17.1.1.2
OSPT  TCP        (----------)       (----------)     110.10.10.1:2048     17.1.1.3:1024

 Type  Pro      Inside Global       Inside Local       Outside Local      Outside Global
Age
----- ---- -------------------- -------------------- -------------------- ---
---------------- -----
 OSAT  253    110.10.10.2:0       110.10.10.2:0       110.10.10.1:0        17.1.1.2:0      600 (-
---)(2:1)
OSPT  TCP    110.10.10.2:1024    110.10.10.2:1024    110.10.10.1:2048     17.1.1.3:1024
1800 (NORMAL:0)(2:1)
```

## 19.3.10. Configure Outside Source NAT Dynamic Address Translation

### Network Requirements

- PC1 is the internal host, PC2 and PC3 are the external hosts, Device is the NAT device, gigabitethernet0 is the NAT internal interface, and gigabitethernet1 is the NAT external interface. Configure the outside source NAT dynamic address translation
- Users in the external IP address segment 17.1.1.0/24 can successfully visit PC1, using the internal IP address 110.10.10.2 and 110.10.10.3.

## Network Topology

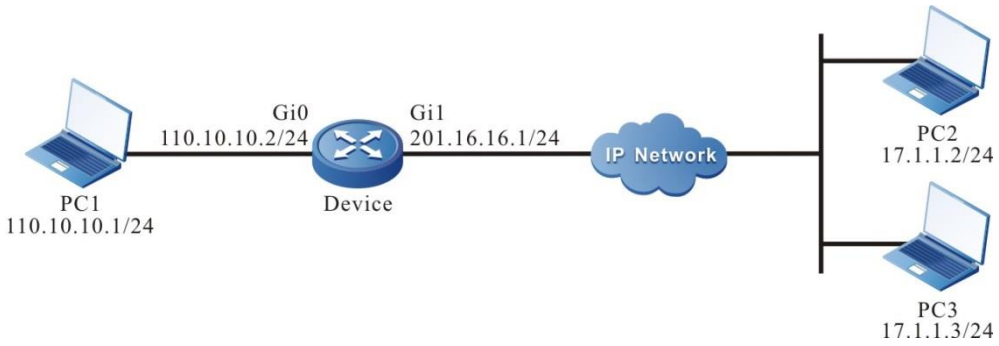



Figure 19-9 Networking of configuring the outside source NAT dynamic address translation

## Configuration Steps

**Step 1:** Configure the IP addresses and routings for all the interfaces. (Omitted)

**Step 2:** Configure the NAT interface attribute and NAT rule for Device.

#Configure the inside interface with gigabitethernet0 as the NAT.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
```

#Configure the outside interface with gigabitethernet1 as the NAT.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT IP address pool, pool1, containing two IP addresses 110.10.10.2 and 110.10.10.3.

```
Device(config)#ip nat pool pool1 110.10.10.2 110.10.10.3 netmask 255.255.255.0
```

# Configure the ACL 1001, only permitting PCs of the source IP address segment 17.1.1.0/24 to visit PC1.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 17.1.1.0 0.0.0.255 any
Device(config-ext-nacl)#exit
```

#Configure the outside source NAT dynamic address translation rule, translating the Internet IP addresses 17.1.1.2 and 17.1.1.3 to the intranet IP addresses 110.10.10.2 and 110.10.10.3.

```
Device(config)#ip nat outside source list 1001 pool pool1
```

**Step 3:** Check the result.

# When PC2 and PC3 visit PC1, the NAT table entry can be viewed on Device.

```
Device#show ip nat translations
 Type  Pro      Inside Global      Inside Local       Outside Local      Outside Global
Age
----- ---- -------------------- -------------------- -------------------- ---
---------------- -----
 ONAT  253     110.10.10.1:0      110.10.10.1:0      110.10.10.2:0      17.1.1.2:0      600 (-
---)(2:1)
 ONAT  253     110.10.10.1:0      110.10.10.1:0      110.10.10.3:0      17.1.1.3:0      600 (-
---)(2:1)
```

# 20. NAT64

## 20.1. Overview

With the exhaustion of IPv4 public addresses, operators can provide user access services in IPv6 mode; however, most services on the Internet are still pure IPv4 services. In order to realize the communication between IPv6 host and IPv4 server in the process of IPv4-IPv6 migration, you can adopt the NAT64 scheme.

### 20.1.1. NAT64 IPv6 Network Specific Prefix

NAT64 translation process requires IPv6 address translated by IPv4, which is composed of IPv6 network specific prefix (NSP: Network-Specific Prefix) and IPv4 address. The structure of the IPv6 address converted by IPv4 is shown in Figure 1-1.

NAT64 IPv6 network-specific prefixes are used to translate IPv4 addresses to IPv6 addresses:

- Translate the destination address of the IPv6 packet to the IPv4 address: When the IPv6 packet arrives at the NAT64 device, NAT64 resolves the IPv4 address from the IPv6 destination address (IPv6 address translated by IPv4 destination address) using IPv6 network specific prefix.
- Translate the source address of the IPv4 packet to the IPv6 address: When the IPv4 packet arrives at the NAT64 device, NAT64 uses the IPv6 network specific prefix to translate the IPv4 source address to the IPv6 address.

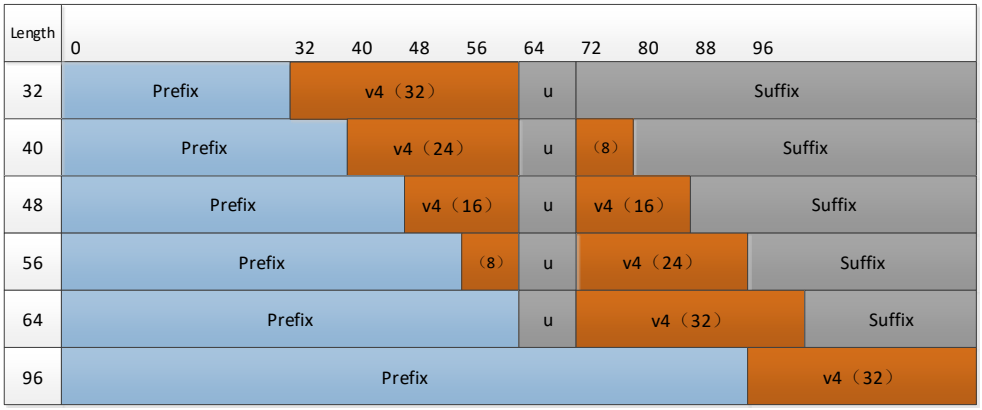



Figure 14-1 The structure of the IPv6 address translated by the IPv4 address

The length of the IPv6 network specific prefix can only be 32, 40, 48, 56, 64 or 96 bits, and the 8-bit u is the reserved bits, set to 0; suffix is currently not used, and set to 0. The v4 addresses are embedded in different locations of the v6 addresses according to the prefix length.

- When the prefix length is 32 bits, the v4 address is embedded in 32-63 bits.
- When the prefix length is 40 bits, the v4 address is divided into two parts. The first 24 bits are embedded in 40-63 bits, and the last 8 bits are embedded in 72-79 bits.
- When the prefix length is 48 bits, the v4 address is divided into two parts. The first 16 bits are embedded in 48-63 bits, and the last 16 bits are embedded in 72-87 bits.
- When the prefix length is 56 bits, the v4 address is divided into two parts. The first 8 bits are embedded in 56-63 bits, and the last 24 bits are embedded in 72-95 bits.
- When the prefix length is 64 bits, the v4 address is embedded in 72-103 bits.
- When the prefix length is 96 bits, the v4 address is embedded in 96-127 bits.

## 20.1.2. **NAT64 Packet Translation Process**

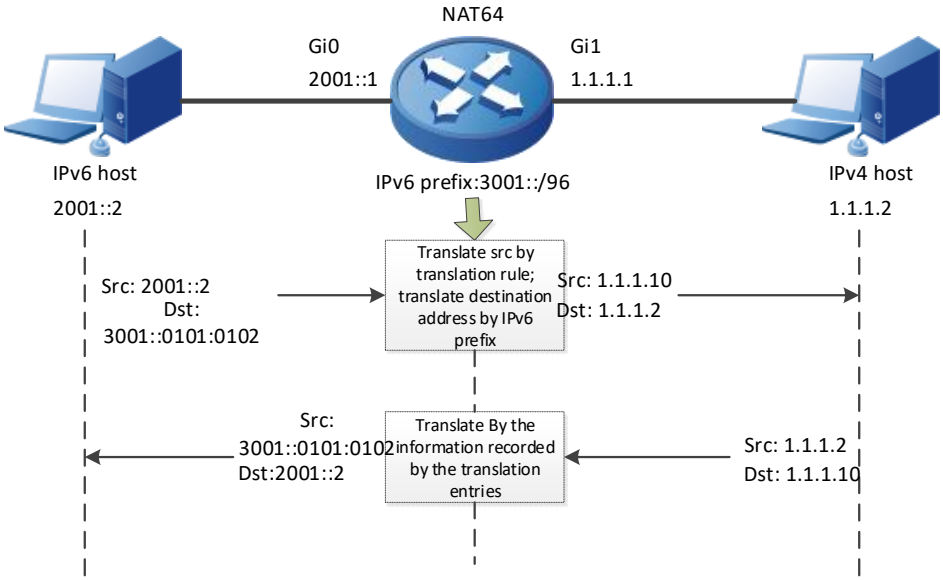



Figure 14-2 The packet translation process of IPv6 actively initiating access

1. After receiving the IPv6 packet, NAT64 determines whether the packet needs to be translated to the IPv4 packet. If the IPv6 packet matches the NAT64 static translation rule or dynamic translation rule, the packet needs NAT64 translation.
2. Translate source address: After the IPv6 packet matches the translation rule successfully, translate the source address by using the configuration in the translation rule. For the static translation rule, directly translate the source address of the IPv6 packet to the IPv4 address configured in the static rule; for the dynamic translation rule, translate the source address of IPv6 packet to the IPv4 address in the address pool, or to the IPv4 address of the interface associated with the dynamic rule.
3. Translate destination address: According to the NAT64 IPv6 prefix, resolve the IPv4 address from the IPv6 destination address as the translated destination IPv4 address.
4. After translation, NAT64 creates the translation entries and records the mapping relationship between IPv6 packets and IPv4 packets.
5. The response packet is translated from the IPv4 packet to the IPv6 packet via the translation entry.

## 20.2. NAT64 Function Configuration

Table 14-1 NAT64 function configuration

| Configuration Task | |
|---|---|
| Enable the NAT64 function | Enable the NAT64 function |
| Configure the NAT64 flow log function | Configure NAT64 new flow log function |
| | Configure NAT64 end flow log function |
| Configure NAT64 IPv6 network specific prefix | Configure NAT64 IPv6 network specific prefix |
| Configure the NAT64 filter policy | Configure the NAT64 filter policy |
| Configure the NAT64 address pool | Configure the NAT64 address pool |
| Configure the NAT64 available port range | Configure the NAT64 available port range |
| Configure NAT64 IPv6 internal server translation rule | Configure NAT64 IPv6 internal server translation rule |
| Configure the NAT64 static translation rule | Configure the NAT64 static translation rule |
| Configure the NAT64 dynamic translation rule | Configure the associated dynamic translation rule of IPv6 ACL |
| | Configure the associated dynamic translation rule of IPv6 prefix |

### 20.2.1. Enable the NAT64 Function

After enabling the NAT64 function on the interface connecting IPv6 and IPv4, the translation between the IPv6 packet and IPv4 packet can be performed.

**Configuration Condition**

None

### Enable the NAT64 Function

Table 14-2 Enable the NAT64 function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | configure terminal | - |
| Enter the interface configuration mode | interface *interface-name* | Mandatory |
| Enable the NAT64 function | nat64 enable | Mandatory<br>By default, do not enable the NAT64 function. |

## 20.2.2. Configure NAT64 Flow Log Function

After configuring the new flow log function of NAT64, the corresponding data log will be output when the translation table entry of NAT64 is generated. After configuring the end flow log function of NAT64, the corresponding data log will be output when the translation table entries of NAT64 are aged or cleared normally.

### Configuration Condition

None

### Configure NAT64 Flow Log Function

Table 20-2 Configure the NAT64 flow log function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT64 flow log function | **nat64 logging security-data { connection-begin \| connection-end } *** | Mandatory<br>By default, do not configure the NAT64 flow log function. |

## 20.2.3. Configure NAT64 IPv6 Network Specific Prefix

### Configuration Condition

None

### Configure NAT64 IPv6 Network Specific Prefix

Table 14-3 Configure the NAT64 IPv6 network specific prefix

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT64 IPv6 network specific prefix | **nat64 prefix** *ipv6-prefix / lenght* | Mandatory<br>By default, do not Configure the NAT64 IPv6 network specific prefix. |

**Note:**

- If no IPv6 network-specific prefix is configured, NAT64 adopts the well-known prefix 64:ff9b:/96 to translate packets.
- It is not recommended to use this rule together with the static source translation rule from V4 to V6. The usage scenarios of the two rules are different. If they are used together, unexpected results may appear.

## 20.2.4. Configure the NAT64 Filter Policy

After configuring NAT64 dynamic translation rules, the host of the IPv6 side can actively access the host of the IPv4 side, while the host of the IPv4 side actively accessing the host of the IPv6 side is limited. Whether to permit the host of the IPv4 side to initiate access actively is determined by the NAT64 filtering policy. NAT64 provides three filtering policies: endpoint-independent filtering, address-dependent filtering, and all.

Endpoint-independent filtering:

When the host of the IPv6 side accesses the host of the IPv4 side, NAT64 assigns an IPv4 address and port to it. All hosts of the IPv4 side can access the IPv6 host through this address and port.

Address-dependent filtering:

When the host of the IPv6 side accesses the host of the IPv4 side, NAT64 assigns an IPv4 address and port to it. The host of the IPv4 side can access the IPv6 host through this address and port, but the IPv4 host must have been accessed by the IPv6 host.

All:

The host of the IPv4 side (except for the response packets with quintuple matching) is not allowed to access the host of the IPv6 side through dynamic translation rules.

### Configuration Condition

It is necessary to configure the dynamic rule.

### Configure NAT64 Filter Policy

Table 14-4 Configure the NAT64 filter policy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT64 filter policy | **nat64 filter** { **address-dependent \| endpoint-independent \| all** } | Mandatory<br>By default, the NAT64 filtering policy is endpoint-independent filtering |

**Note:**

- The filtering policy is only for the TCP, UDP and ICMP packets, and the other packets cannot actively access the IPv6 side from the IPv4 side.

## 20.2.5. Configure the NAT64 Address Pool

### Configuration Condition

None

### Configure the NAT64 Address Pool

The NAT64 address pool consists of a set of consecutive IPv4 addresses. After dynamic NAT64 translation rules associate the address pool, and when packets are sent from the IPv6 network to the IPv4 network, NAT64 takes an IPv4 address from the address pool as the source address of the translated IPv6 packet.

Table 14-5 Configure the NAT64 address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT64 address pool | **nat64 v4 pool** *pool-name start-IP end-IP* { **netmask** *network-mask* \| **prefix-length** *prefix-length* } | Mandatory<br>By default, do not configure the NAT64 address pool. |

**Note:**

- The same address should not be defined in two different address pools.
- The same address pool can only be associated once, and different dynamic transformation rules can only associate different address pools.

## 20.2.6. Configure NAT64 Available Port Range

### Configuration Conditions

None

### Configure NAT64 Available Port Range

The port multiplexing of NAT64 will map different IPv6 addresses to different ports of the same IPv4 address to reduce the consumption of the IPv4 address to the greatest extent. The available port range of IPv4 address can be configured. After configuring the available port range of NAT64, the source ports of all packets converted by NAT64 are within the configured available port range. By configuring the available range, we can meet some requirements for the use of ports, such as not using well-known ports.

Table 20-3 Configure NAT64 available port range

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT64 available port range | **nat64 v4 port-range start** *start-port* **end** *end-port* | Mandatory<br>By default, the available ports of the NAT64 pool address are10001-65535. |

## 20.2.7. Configure NAT64 v4-v6 Source Static Translation Rule

NAT64 static translation rule converts the IPv4 address to the IPv6 address through static mapping, and the mapping relationship is fixed. The IPv4 side host can access the IPv6 side host through the source static translation rules, and the IPv6 side host can also actively access the IPv4 side host through the source static rules. The priority of source static translation rule is higher than that of the dynamic translation rule. When both source static translation rule and dynamic translation rule are configured, the source static rule is preferred.

### Configuration Conditions

None

### Configure NAT64 Static Translation Rules

Table 20-4 Configure NAT64 static translation rule

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT64 v4v6 source static translation rule | **nat64 v4v6 source** *ipv4-address* [**vrf** *ipv4-vrf-name*] *ipv6-address* [ **vrf** *ipv6-vrf-name* ] | Mandatory<br>By default, do not configure NAT64 v4v6 source static translation rule. |

**Note:**

- It is not recommended to use this rule together with IPv6 network specific prefix rule. The usage scenarios of the two rules are different. If they are used together, unexpected results may appear.

## 20.2.8. Configure NAT64 IPv6 Internal Server Translation Rule

NAT64 IPv6 internal server translation rules, through static mapping, the server address and port on IPv6 side are converted to corresponding IPv4 addresses and ports. The mapping relationship is fixed and unchanged. IPv4 side client hosts can access IPv6 side server hosts through internal server translation rules. The priority of IPv6 internal server translation rules is higher than that of the static translation rules and dynamic translation rules. When IPv6 internal server translation rules, static translation rules and dynamic translation rules are configured at the same time, IPv6 internal server translation rules are preferred for translation.

### Configuration Conditions

None

### Configure NAT64 IPv6 Internal Server Translation Rules

Table 20-5 Configure NAT64 IPv6 internal server translation rules

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT64 IPv6 internal server translation rules | **nat64 v6server protocol** *protocol-type ipv4-address ipv4-port* [ **vrf** *ipv4-vrf-name* ] *IPv6-address IPv6-port* [ **vrf** *IPv6-vrf-name* ] | Mandatory<br>By default, do not configure NAT64 IPv6 internal server translation rules. |

**Note:**

- IPv6 internal server translation rules are not limited by NAT64 filtering policy

## 20.2.9. Configure NAT64 Static Translation Rules

NAT64 static translation rules translate the IPv6 address to the IPv4 address via static mapping, and the mapping relationship is fixed. The host of the IPv6 side can access the host of the IPv4 side through static translation rules, and the host of the IPv4 side can actively access the host of the IPv6 side through static rules. The priority of the static translation rule is higher than that of the dynamic translation rule. When static translation rules and dynamic translation rules are configured simultaneously, static translation rules are preferred.

### Configuration Condition

None

### Configure NAT64 Static Translation Rules

Table 14-4Configure the NAT64 static translation rules

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT64 static translation rules | **nat64 static** *ipv4-address* [**vrf** *ipv4-vrf-name*] *IPv6-address* [ **vrf** *IPv6-vrf-name* ] | Mandatory<br>By default, do not configure the NAT64 static translation rules. |

**Note:**

- The static translation rule is not limited by the NAT64 filtering policy.

## 20.2.10. Configure NAT64 Dynamic Translation Rules

The address mapping relationship between the IPv6 network and the IPv4 network is dynamic. If the packet matches the translation rule, the source IPv6 address will be translated to the IPv4 address in the specified address pool or the IPv4 address of the specified interface. There are two ways of matching the translation rules: one is the configured ACL6 rule matching the source address of the IPv6 packet, the other is the destination address of the IPv6 packet matching the configured prefix in the rule.

NAT64 supports the following dynamic translations:

- The associated dynamic translation rule of IPv6 ACL
- The associated dynamic translation rule of the IPv6 prefix

### Configuration Condition

Before configuring the NAT64 dynamic translation rule, first complete the following task:

- Configure IPv6 ACL
- Configure NAT64 address pool

### Configure Associated Dynamic Translation Rules of IPv6 ACL

The source IPv6 address of the IPv6 packet matches the specified IPv6 ACL in the translation rule, and the source IPv6 address is translated to the IPv4 address in the address pool or to the IPv4 address of the specified interface. When the overload keyword is specified in the configuration, different IPv6 addresses can be translated to the same IPv4 address, which can be distinguished by different port numbers. Configuring the overload keyword can save the IPv4 address to the greatest extent.

Table 14-8 Configure the associated dynamic translation rule of IPv6 ACL

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the associated dynamic translation rule of IPv6 ACL | **nat64 v6v4 list** *access-list-name* [ **vrf** *IPv6-vrf-name* ] { **interface** *interface-name* **overload \| pool** *pool-name* [ **overload** ] } [ **vrf** *ipv4-vrf-name* ] | Mandatory<br>By default, do not configure the associated dynamic translation rule of IPv6 ACL. |

### Configure Associated Dynamic Translation Rules of IPv6 Prefix

The destination IPv6 address of the IPv6 packet matches the specified IPv6 prefix in the translation rule, and the source IPv6 address is translated to the IPv4 address in the address pool or to the IPv4 address of the specified interface. When the overload keyword is specified in the configuration, different IPv6 addresses can be translated to the same IPv4 address, which can be distinguished by different port numbers. Configuring the overload keyword can save the IPv4 address to the greatest extent.

Table 14-95 Configure the associated dynamic translation rule of IPv6 prefix

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the associated dynamic translation rule of the IPv6 prefix and address pool | **nat64 v6v4 prefix** *IPv6-prefix/length* [ **vrf** *IPv6-vrf-name* ] { **interface** *interface-name* **overload \| pool** *pool-name* [ **overload** ] } [ **vrf** *ipv4-vrf-name* ] | Mandatory<br>By default, do not configure the associated dynamic translation rule of the IPv6 prefix. |

## 20.2.11. Configure NAT64 IPv4 Internal Server Translation Rules

NAT64 IPv4 internal server translation rules, through the way of static mapping, the server address and port of the IPv4 side are converted to the corresponding IPv6 address and port, and the mapping relationship is fixed. IPv6 side client host can access the IPv4 side server host through internal server translation rules. The priority of IPv4 internal server translation rule is higher than that of static translation rule and dynamic translation rule. When IPv4 internal server translation rule, static conversion rule and dynamic translation rule are configured at the same time, IPv4 internal server translation rule is preferred.

### Configuration Condition

None

### Configure NAT64 IPv4 Internal Server Translation Rules

Table 20-6 Configure NAT64 IPv4 internal server translation rules

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT64 IPv4 internal server translation rules | **nat64 v4server list** access-*list-name* [ **vrf** *ipv6-vrf-name* ] *ipv4-address* [ *ipv4-port* ] [ **vrf** *ipv4-vrf-name* ] | Mandatory<br>By default, do not configure NAT64 IPv4 internal server translation rules. |

**Note:**

- The IPv4 internal server translation rule is not limited by the NAT64 filtering policy.

## 20.2.12. Configure NAT64 ALG Switch

NAT64 ALG switch is used to enable the processing function of application layer packet. At present, the application layer protocols supported by NAT64 include FTP, TFTP, DNS, HTTP and SIP.

### Configuration Condition

None

### Configure NAT64 ALG Switch

Table 20-7 Configure NAT64 ALG switch

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT64 ALG switch | **nat64 alg** { **ftp \| tftp \| dns \| http** \| **sip \| all** } | Mandatory<br>By default, the NAT64 ALG is not configured, and the ALG function is not enabled. |

## 20.2.13. NAT64 Monitoring and Maintaining

Table 14-12 NAT64 monitoring and maintaining

| Command | Description |
|---|---|
| **clear nat64 statistics** | Clears all statistics information of NAT64 |
| **show nat64 address-pool** | Displays the NAT64 address pool information |
| **show nat64 prefix** | Displays the NAT64 IPv6 prefix information |
| **show nat64 rule** | Displays the NAT64 translation rule information |
| **show nat64 statistics** [**verbose**] | Displays NAT64 global or board statistics |

| Command | Description |
|---|---|
| **show nat64 translation lpu** *lpu-unit* [ **proto** { **icmp \| tcp \| udp** \| **ip** } ] [ **v6src** *IPv6-address* ] [ **v6dst** *IPv6-address* ] [ **v4src** *ipv4-address* ] [ **v4dst** *ipv4-address* ] | Displays the NAT64 translation entries |
| **show running-config nat64** | Displays all configuration information of NAT64 |

## 20.3. NAT64 Typical Configuration Examples

### 20.3.1. Configure NAT64 Static Address Translation

#### Network Requirements

- The PC with the address 2001::2/64 in the IPv6 network hopes to access the server with the IPv4 network address 192.0.2.2/24. In order to meet the above requirement, it is necessary to deploy NAT64 translation device between the IPv4 network and the IPv6 network, and when configuring the static NAT64 address mapping on Device, ensure that the host in the IPv6 network can access network resources in IPv4.

#### Network Topology

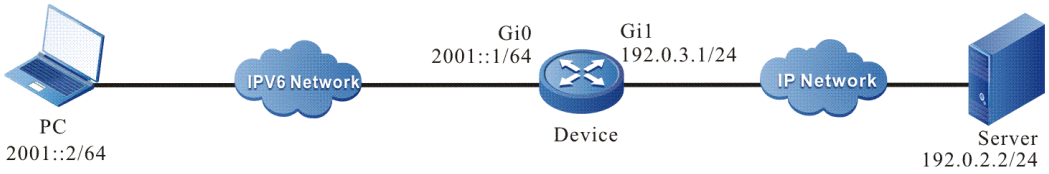



Figure 20-8 Networking of configuring the NAT64 static address translation

#### Configuration Steps

**Step 1:** Configure the interface IP address (omitted).

**Step 2:** Configure the IPv6 route of the PC host.

On the PC, configure the static route to the prefix 2001:1::/96.

**Step 3:** Configure the NAT64 prefix, NAT64 interface attribute, and NAT64 translation rules of Device.

#Configure the NAT64 prefix.

```
Device#configure terminal
Device(config)#nat64 prefix  2001:1::/96
```

#Enable the NAT64 function on interface gigabitethernet0.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Enable the NAT64 function on interface gigabitethernet1.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT64 static address translation rule.

```
Device(config)#nat64 static  192.0.3.2 2001::2
```

**Step 4:** Check the result.

#On PC, ping 2001:1::c000:0202, and the ping can be connected.

**Note:**

- IPv6 address 2001:1::c000:0202 is the corresponding IPv6 address of the server in the IPv4 network (consisting of network prefix + hexadecimal IPv4 address).

```
Pinging 2001:1::c000:0202 with 32 bytes of data:

Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms

Ping statistics for 2001:1::c000:0202:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On Device, you can see the corresponding NAT64 translation entry.

```
Device#show nat64 translation
Proto   IPv6 source:port              IPv4 source:port        Status       lifetime
        IPv6 destination:port         IPv4 destination:port
-----   --------------------          --------------------    ------       --------
ICMP    [2001::2]:195                 192.0.3.2:195           ----         43
        [2001:1::c000:202]:0          192.0.2.2:0
```

## 20.3.2. Configure NAT64 Dynamic Address Translation

### Network Requirements

- Many hosts in the IPv6 network, such as PC1 or PC2, need to actively access the servers in the IPv4 network and record the translation entries. To meet the above requirements, the NAT64 translation device needs to be deployed between the IPv4 network and the IPv6 network. As long as the number of the IPV6 hosts initiated to the

IPV4 server in the IPV6 domain is not more than the number of the IPV4 addresses in IPV4 address pool, you can configure the NAT64 dynamic address translation on the Device, meeting the requirement that multiple hosts in its IPv6 network actively access the network resources in the IPv4 network.

## Network Topology

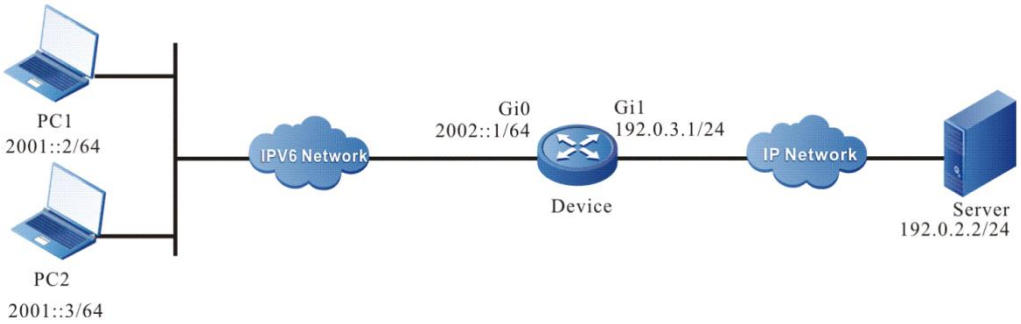



Figure 20-9 Networking of configuring the NAT64 dynamic address translation

## Configuration Steps

**Step 1:** Configure the interface IP address (omitted).

**Step 2:** Configure the IPv6 route of host PC1 and PC2.

On PC1 and PC2, configure the static route to the prefix 2001:1::/96.

**Step 3:** Configure the NAT64 prefix, NAT64 interface attribute, and NAT64 translation rule of Device.

#Configure the NAT64 prefix.

```
Device#configure terminal
Device(config)#nat64 prefix  2001:1::/96
```

#Configure interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure IPv6 ACL 7001, only permitting the PC of the 2001::/64 segment in the IPv6 network to access Server.

```
Device(config)#ipv6 access-list extended  7001
Device(config-v6-list)#permit ipv6 2001::/64 any
Device(config-v6-list)exit
```

#Configure NAT64 v4 address pool1, containing two IPv4 addresses 192.0.3.3 and 192.0.3.4.

```
Device(config)# nat64 v4 pool  pool1  192.0.3.3 192.0.3.4 netmask  255.255.255.0
```

#Configure the NAT64 dynamic address translation rule.

```
Device(config)#nat64 v6v4 list  7001 pool  pool1
```

**Step 4:** Check the result.

#On PC1, ping 2001:1::c000:0202, and the ping can be connected.

**Note:**

- IPv6 address 2001:1::c000:0202 is the corresponding IPv6 address of the server in the IPv4 network (consisting of network prefix + hexadecimal IPv4 address).

```
Pinging 2001:1::c000:0202 with 32 bytes of data:

Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms

Ping statistics for 2001:1::c000:0202:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On PC2, ping 2001:1::c000:0202, and the ping can be connected.

```
Pinging 2001:1::c000:0202 with 32 bytes of data:

Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms

Ping statistics for 2001:1::c000:0202:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On Device, you can see the corresponding NAT64 translation entry.

```
Device#show nat64 translation
Proto   IPv6 source:port          IPv4 source:port       Status       lifetime
```

```
        IPv6 destination:port       IPv4 destination:port
-----  --------------------      --------------------   ------      --------
ICMP    [2001::3]:195             192.0.3.4:195          ----        57
        [2001:1::c000:202]:0      192.0.2.2:0
ICMP    [2001::2]:195             192.0.3.3:195          ----        51
        [2001:1::c000:202]:0      192.0.2.2:0
```

## 20.3.3. **Configure NAT64 Dynamic Port Translation**

### Network Requirements

- Many hosts in the IPv6 network, such as PC1 or PC2, need to actively access the servers in the IPv4 network and record the translation entries. To meet the above requirements, the NAT64 translation device needs to be deployed between the IPv4 network and the IPv6 network. When the number of the IPV6 hosts initiated to the IPv4 server in the IPV6 domain is larger than the number of the IPV4 addresses in the IPv4 address pool, you can configure the NAT64 dynamic address translation on the Device, meeting the requirement that multiple hosts in its IPv6 network actively access the network resources in the IPv4 network.

### Network Topology

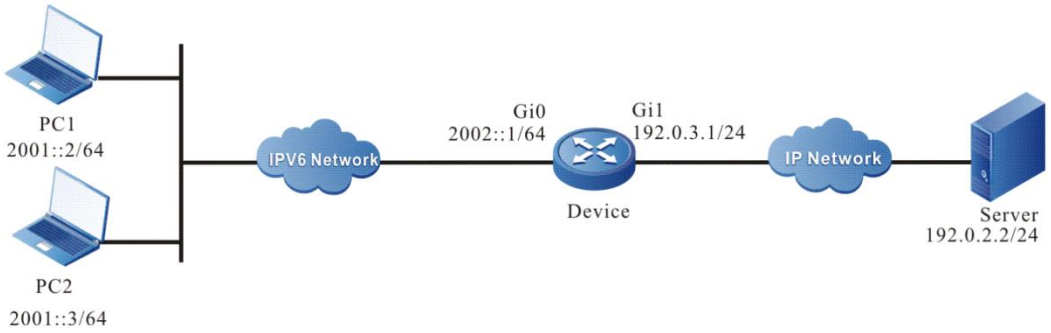



Figure 20-10 Networking of configuring the NAT64 static address translation

### Configuration Steps

**Step 1:** Configure the interface IP address (omitted).

**Step 2:** Configure the IPv6 route to the host PC1 and PC2.

On PC1 and PC2, configure the static route to the prefix 2001:1::/96.

**Step 3:** Configure the NAT64 prefix, NAT64 interface attribute, and NAT64 translation rule of Device.

#Configure the NAT64 prefix.

```
Device#configure terminal
Device(config)#nat64 prefix  2001:1::/96
```

#Configure the interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
```

```
Device(config-if-gigabitethernet0)#exit
```

#Configure the interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure IPv6 ACL 7001, only permitting the PC of the 2001::/64 segment in the IPv6 network to access Server.

```
Device(config)#ipv6 access-list extended  7001
Device(config-v6-list)#permit ipv6 2001::/64 any
Device(config-v6-list)exit
```

#Configure the NAT64 v4 address pool pool1, containing two IPv4 addresses 192.0.3.3 and 192.0.3.4.

```
Device(config)#nat64 v4 pool  pool1  192.0.3.3 192.0.3.4 netmask  255.255.255.0
```

#Configure the NAT64 dynamic address translation rule.

```
Device(config)#nat64 v6v4 list  7001 pool  pool1 overload
```

**Note:**

- When the dynamic rule is configured as overload (port multiplexing), ports will be allocated to the addresses of the address pool according to the port range (10001-65535 by default). When one address has occupied all the available ports, the next address will be used.

**Step 4:** Check the result.

#On PC1, ping 2001:1::c000:0202, and the ping can be connected.

**Note:**

- IPv6 address 2001:1::c000:0202 is the corresponding IPv6 address of the server in the IPv4 network (consisting of network prefix + hexadecimal IPv4 address).

```
Pinging 2001:1::c000:0202 with 32 bytes of data:

Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms

Ping statistics for 2001:1::c000:0202:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On PC2, ping 2001:1::c000:0202, and the ping can be connected.

```
Pinging 2001:1::c000:0202 with 32 bytes of data:

Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms
Reply from 2001:1::c000:0202: time<1ms

Ping statistics for 2001:1::c000:0202:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

#On Device, you can see the corresponding NAT64 translation entry.

```
Device#show nat64 translation
Proto  IPv6 source:port          IPv4 source:port       Status      lifetime
       IPv6 destination:port     IPv4 destination:port
-----  --------------------      --------------------   ------      --------
ICMP   [2001::3]:195             192.0.3.4:195          ----        56
       [2001:1::c000:202]:0      192.0.2.2:0
ICMP   [2001::2]:195             192.0.3.3:195          ----        50
       [2001:1::c000:202]:0      192.0.2.2:0
```

## 20.3.4. Configure IPV4 internet to Access IPV6 Internal Server

### Network Requirements

- Multiple hosts such as PC1 or PC2 in IPv4 network need to actively access the internal HTTP, FTP and other servers in IPv6 network, and record the translation table entries. In order to meet the above requirements, NAT64 translation device needs to be deployed between IPv4 network and IPv6 network. When initiating the access to IPv6 internal server in IPv4 domain, NAT64 internal server rules need to be configured on the device. It meets the requirement that the host in the IPv4 network can actively access the network resources in IPv6.

## Network Topology

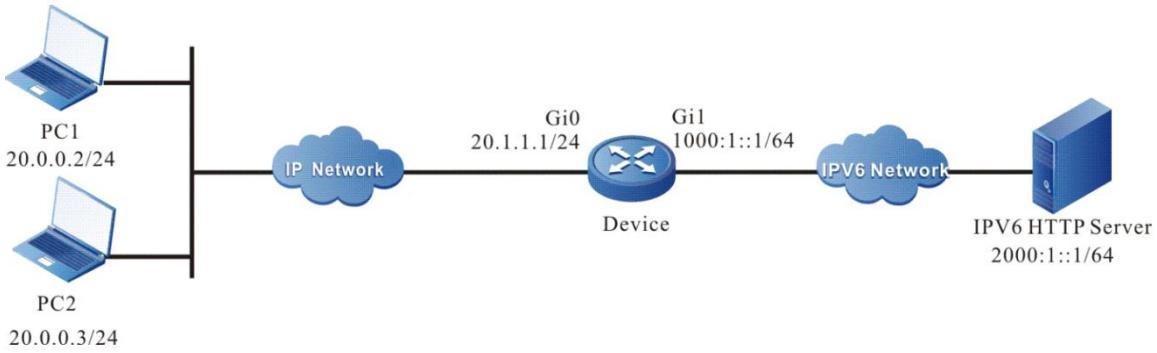



Figure 20-6 Networking of configuring IPV4 Internet to access IPv6 internal server

## Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the IPv4 route of the host PC1 and PC2.

On PC1 and PC2, configure the static route to 20.1.1.0/24.

**Step 3:** Configure the NAT64 prefix, NAT64 interface attribute, and NAT64 translation rule of Device.

#Configure the NAT64 prefix.

```
Device#configure terminal
Device(config)#nat64 prefix 2001:1::/96
```

#Configure the interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#ipv6 enable
Device(config-if-gigabitethernet0)#exit
```

#Configure the interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure the corresponding IPv4 address and port of the IPv6 internal server.

```
Device(config)# nat64 v6server protocol tcp 20.1.1.2 1024 2000:1::1 80
```

**Step 4:** Check the result.

#In PC1, access the IPv6 HTTP server through HTTP protocol. The address and port are 20.1.1.2 and 1024 respectively.

#In PC2, access the IPv6 HTTP server through HTTP protocol. The address and port are 20.1.1.2 and 1024 respectively.

#On Device, you can see the corresponding NAT64 translation table entries. When the PC accesses the IPv4 address 20.1.1.2 and port 1024, it will be converted to the address 2000:1:: 1 and port 80 of the internal server.

```
Device#show nat64 translation

Proto   IPv6 source:port          IPv4 source:port          lifetime
        IPv6 destination:port     IPv4 destination:port
-----   --------------------      --------------------      --------
TCP     [2000:1::1]:80            20.1.1.2:1024             7155
        [2001:1::1400:0002]: 45535    20.0.0.2:45535
TCP     [2000:1::1]:80            20.1.1.2:1024             7155
        [2001:1::1400:0003]: 45536    20.0.0.3:45536
Valid/Total: 2/2
```

## 20.3.5. Configure IPV6 internet to Access IPV4 Internal Server

### Network Requirements

- Multiple operators such as PC1 or PC2 in IPv6 network need to actively access the internal server in IPv4 network, and record the translation table entries. In order to meet the above requirements, NAT64 translation device needs to be deployed between IPv4 network and IPv6 network. When initiating the access to IPv4 internal server in IPv6 domain, NAT64 internal server rules need to be configured on Device. It meets the requirement that the host in the IPv6 network can actively access the network resources in IPv4.

### Network Topology

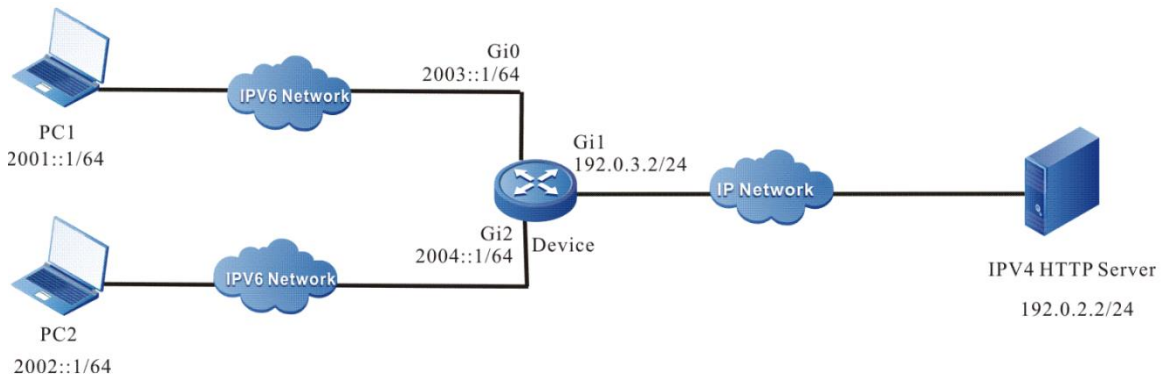



Figure 20-7 Networking of configuring IPV6 Internet to access IPv4 internal server

### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the IPv6 route of the host PC1 and PC2.

On PC1 and PC2, configure the routes to 2003::/64 and 2004::/64.

**Step 3:** Configure the destination address ACL, NAT64 interface attribute, and NAT64 translation rules of Device.

#Configure the ACL associated with the IPv4 internal server.

```
Device#configure terminal
Device(config)#ipv6 access-list extended 7001
Device(config-v6-list)#permit ipv6 any host 2003::1
Device(config-v6-list)#permit ipv6 any host 2004::1
Device(config-v6-list)#exit
```

#Configure the interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Configure the interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure interface gigabitethernet2 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 2
Device(config-if-gigabitethernet2)#nat64 enable
Device(config-if-gigabitethernet2)#exit
```

#Configure NAT64 V4server rules for IPv6 destination address translation.

```
Device(config)# nat64 v4server list 7001 192.0.2.2
```

#Configure the NAT64 IPv4 address pool.

```
Device(config)# nat64 v4 pool pool1 192.0.2.10 192.0.2.20 netmask 255.255.255.0
```

#Configure NAT64 V6V4 list rule for the IPv6 source address translation.

```
Device(config)# nat64 v6v4 list 7001 pool pool1
```

**Step 4:** Check the result.

#On PC1, access IPV4 HTTP Server via the HTTP protocol.

#On PC2, access IPV4 HTTP Server via the HTTP protocol.

#On Device, you can see the corresponding NAT64 translation table entries. When the PC accesses the IPv6 addresses 2003:: 1 and 2004:: 1, it converts them to the address 192.0.2.2 of the IPv4 internal server.

```
Device#show nat64 translation

Proto   IPv6 source:port          IPv4 source:port          lifetime
        IPv6 destination:port     IPv4 destination:port
-----   --------------------      --------------------      --------
TCP     [2001::1]:10054           192.0.2.10:10054          3599
```

```
        [2003::1]:80              192.0.2.2:80
TCP     [2002::1]:10055           192.0.2.11:10055          3599
        [2004::1]:80              192.0.2.2:80
Valid/Total: 2/2
```

## 20.3.6. Configure Exchange Access of IPV4 and IPv6 Networks

### Network Requirements

- IPv4 network and IPv6 network can access each other. In this scenario, multiple IPv4 users and multiple IPv6 users can access each other, and any party can initiate the access actively. nat64 v4v6 destination rule realizes the stateless conversion of IPv4 and IPv6 addresses.

### Network Topology

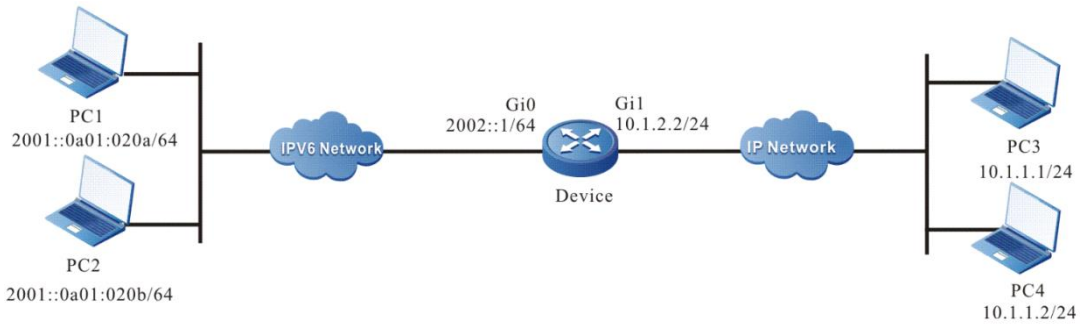



Figure 20-8 Networking of configuring the exchange access of IPv4 and IPv6 networks

### Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT64 interface attributes and NAT64 translation rules of Device.

#Configure the associated ACL of nat64 v4v6 destination rules.

```
Device#configure terminal
Device(config)#ip access-list extended list1
Device(config-std-nacl)# permit ip any 10.1.2.0 0.0.0.255
Device(config-std-nacl)# permit ip 10.1.2.0 0.0.0.255 any
Device(config-std-nacl)# exit
```

#Configure the NAT64 prefix and nat64 v4v6 destination translation rules.

```
Device#configure terminal
Device(config)#nat64 prefix 2002::/96
Device(config)#nat64 v4v6 destination list list1 prefix 2001::/96
```

#Configure the interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Configure the interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

**Step 3:** Check the result.

#On PC3, ping 10.1.2.10 and the ping can be connected.

```
Device#ping 10.1.2.10
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 10.1.2.10 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/2 ms.
```

#On PC4, ping 10.1.2.11, and the ping can be connected.

#On Device, you can see the corresponding NAT64 translation table entries.

```
Device#show nat64 translation

Proto   IPv6 source:port            IPv4 source:port            lifetime
        IPv6 destination:port       IPv4 destination:port
-----   --------------------        --------------------        --------
ICMP    [2002::a01:101]:734         10.1.1.1:734                55
        [2001::a01:020a]:734        10.1.2.10:734
ICMP    [2002::a01:102]:735         10.1.1.2:735                55
        [2001::a01:20b]:735         10.1.2.11:735
Valid/Total:2/2
```

#On PC1, ping 2002::a01:101, and the ping can be connected.

```
Device#ping 2002::a01:101
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2002::a01:101 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/2 ms.
```

#On PC2, ping 2002::a01:102 and the ping can be connected.

#On Device, you can see the corresponding NAT64 translation table entries.

```
Device#show nat64 translation

Proto   IPv6 source:port            IPv4 source:port            lifetime
        IPv6 destination:port       IPv4 destination:port
-----   --------------------        --------------------        --------
```

```
ICMP    [2001::a01:020a]:734        10.1.2.10:734          55
        [2002::a01:101]:734         10.1.1.1:734
ICMP    [2001::a01:020b]:735        10.1.2.11:735          55
        [2002::a01:102]:735         10.1.1.2:735
Valid/Total:2/2
```

## 20.3.7. Configure IPv6 internet to Access IPv4 Network

### Network Requirements

- Multiple hosts such as PC1 or PC2 in IPv6 network need to actively access the internal HTTP, FTP and other servers in IPv4 network, and record the translation table entries. In order to meet the above requirements, NAT64 translation device needs to be deployed between IPv4 network and IPv6 network. When the IPv6 domain initiates access to the IPv4 internal server, the target mapping from IPv6 to IPv4 cannot be converted by the way of IPv4 embedded in IPv6 prefix, only by static mapping, adding a nat64 v4v6 source address mapping to achieve static mapping, so as to meet the requirement that the host in the IPv6 network can actively access the network resources in IPv4.

### Network Topology

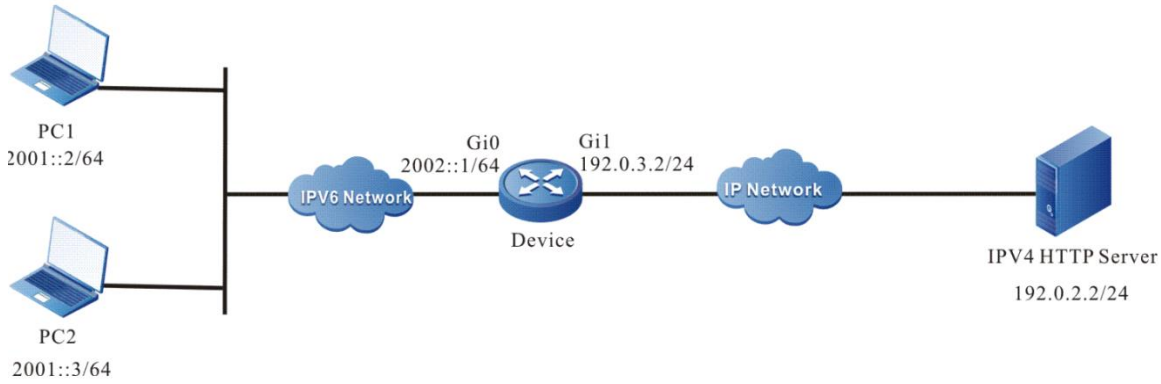



Figure 20-9 Networking of configuring IPV6 Internet to access the IPv4 network

### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the IPv6 route of the host PC1 and PC2.

On PC1 and PC2, configure the route to the prefix 2002::/64.

**Step 3:** Configure the NAT64 prefix, NAT64 attributes, and NAT64 translation rules of Device.

#Configure the NAT64 static translation rule to convert the IPv6 source address.

```
Device#configure terminal
Device(config)#nat64 static 192.0.3.10  2001::2
Device(config)#nat64 static 192.0.3.11  2001::3
```

#Configure the interface gigabitethernet0 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 0
```

```
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Configure the interface gigabitethernet1 to enable the NAT64 function.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#ipv6 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure the corresponding IPv6 address of the IPv4 internal server.

```
Device(config)# nat64 v4v6 source 192.0.2.2 2002::10
```

**Step 4:** Check the result.

#Access the IPV4 HTTP Server : 2002::10 through HTTP protocol.

#Access the IPV4 HTTP Server : 2002::10 through HTTP protocol.

#On Device, you can see the corresponding NAT64 translation table entries. When PC1 and PC2 access the IPv6 address 2002:10, it will be converted to the address 192.0.2.2 of the IPv4 internal server.

```
Device#show nat64 translation

Proto   IPv6 source:port          IPv4 source:port          lifetime
        IPv6 destination:port      IPv4 destination:port
-----   --------------------       --------------------      --------
TCP     [2001::2]:10054            192.0.3.10:10054          3599
        [2002::10]:80              192.0.2.2:80
TCP     [2001::3]:10054            192.0.3.11:10054          3599
        [2002::10]:80              192.0.2.2:80
Valid/Total:2/2
```

# 21. NAT66

## 21.1. Overview

The main function of the NAT66 module is to convert the IPv6 address of IPv6 packet header into another IPv6 address. In fact, the NAT66 module is the supplement of NAT (network address translation) for IPv6 protocol stack. The common use scenario is to convert the source IPv6 address of the internal network into the accessible IPv6 address of the external network. In this way, the internal network host can use a small number of public IPv6 addresses to connect with the Internet.

NAT66 is generally located at the junction of the internal network and the external network. The internal network accesses the external network through the address converted by NAT66. The external network only knows the converted IPv6 address, but does not know the address of the internal network. Therefore, NAT66 "hides" the internal private network and improves the security.

Due to the particularity of NAT66 deployment location, it also brings two kinds of network access requirements. One is that the internal network actively accesses the external network (that is, the internal host accesses the Internet), and the other is that the external network actively accesses the internal network server (that is, the internal server).

For the first kind of requirements (inside actively accesses outside), NAT66 supports statically establishing or dynamically generating the address mapping relationship. According to the generation method of address mapping relationship, address translation can be divided into two types: static translation and dynamic translation:

1. Static translation:

The address mapping relationship between inside and outside is determined in the configuration, which is static one-to-one correspondence. Each "internal address" has an "external address" corresponding to it.

2. Dynamic translation:

The address mapping relationship between inside and outside is determined by the packet. The dynamic translation rule needs to associate the access control list (ACL) with the address pool. The packet filtered by ACL selects an address in the address pool to establish the mapping relationship. After the session of the address accessing the external network, the address resource in the address pool is released to other users.

For the second requirement (outside actively accesses inside), NAT66 supports this requirement by configuring internal server rules. The rule publishes the address and port of the internal server, and enables users who meet the access conditions (matching ACL) to access the server deployed on the inside side normally.

## 21.2. NAT66 Function Configuration

Table 21-1 NAT66 function configuration

| Configuration tasks | |
|---|---|
| Configure the current interface as the NAT66 internal interface | Configure the current interface as the NAT66 internal interface |
| Configure the current interface as the NAT66 external interface | Configure the current interface as the NAT66 external interface |
| Configure the NAT66 internal source static translation rule | Configure the satic translation rule of the NAT66 internal source address |
| | Configure the internal source address static translation rule of the NAT66 TCP packet |
| | Configure the static translation rule of the internal source address of the NAT66 UDP packet |
| Configure the dynamic translation rule of NAT66 internal source address | Configure the NAT66 address pool |
| | Configure the available address segment of NAT66 address pool |
| | Configure IPv6 ACL |
| | Configure the dynamic translation rule of NAT66 internal source address |
| Configure NAT66 internal server rule | Configure NAT66 internal server rule |
| Configure NAT66 to enable the ALG switch | Configure NAT66 to enable the ALG switch |

| Configuration tasks | |
|---|---|
| Configure NAT66 flow log function | Enable NAT66 new flow log function |
| | Enable NAT66 end flow log function |

## 21.2.1. Configure Current Interface as NAT66 Internal Interface

Specify the current interface as the NAT66 internal interface. On the ingress and egress interfaces of the packet, configure nat66 inside and nat66 outside respectively, and then, the packet can be translated according to the NAT66 rule.

### Configuration Conditions

None

### Configure Current Interface as NAT66 Internal Interface

Table 21-2 Configure the current interface as NAT66 internal interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | Mandatory |
| Configure the current interface as NAT66 internal interface | **nat66 inside** | Mandatory<br>By default, do not configure nat66 inside. |

## 21.2.2. Configure Current Interface as NAT66 External Interface

Specify the current interface as the NAT66 external interface. On the ingress and egress interfaces of the packet, configure nat66 inside and nat66 outside respectively, and then, the packet can be translated according to the NAT66 rule.

### Configuration Conditions

None

### Configure Current Interface as NAT66 External Interface

Table 21-3 Configure the current interface as the NAT66 external interface

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | Mandatory |
| Configure the current interface as NAT66 external interface | **nat66 outside** | Mandatory<br>By default, do not configure nat66 outside. |

## 21.2.3. Configure NAT66 Internal Source Address Static Translation Rule

There are three kinds of internal source static translation rules, namely, NAT66 internal source address static translation rules that only do source address translation, of the source address port static translation rule of the TCP protocol, and the source address port static translation rules of the UDP protocol. When the packet passes from inside to outside, one of the three rules can be used to convert the source address (port). The internal source address static translation rule is referred to as SAT (source address translate) rule, and the source address port static translation rule of the TCP/UDP protocol is referred to as SPT (source port translate) rule. SPT rule has higher priority than SAT rule.

### Configuration Conditions

None

### Configure NAT66 Internal Source Address Static Translation Rule

Table 21-4 Configure NAT66 internal source static translation rules

| Step | Command | Description |
|---|---|---|
| **Enter the global configuration mode** | configure terminal | **-** |
| **Configure NAT66 internal source static translation rules** | nat66 inside source static **{** network ***local-subnet/ prefix-length*** **[**vrf ***vrf-name*** **]** ***global-subnet/prefix-length*** **[**vrf ***vrf-name*** **] |** ***local-IPv6*** **[**vrf ***vrf-name*** **]** ***global-IPv6*** **[**vrf ***vrf-name*** **] }** | **Mandatory**<br>**By default, do not configure NAT66 internal source static translation rules.** |

### Configure NAT66 TCP Protocol Internal Source Address Port Static Translation Rule

Table 21-5 Configure NAT66 TCP protocol internal source address port static translation rule

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 TCP protocol internal source address port static translation rule | **nat66 inside source static tcp** { *local-IPv6 local-port* [ **vrf** *vrf-name* ] { *global-IPv6* \| **interface** *interface-name* } *global-port* [ **vrf** *vrf-name* ]\| **range** *local-IPv6 local-port-min local-port-max* [ **vrf** *vrf-name* ] { *global-IPv6* \| **interface** *interface-name* } *global-port-min global-port-max* [ **vrf** *vrf-name* ]} | Mandatory<br>By default do not configure NAT66 TCP protocol internal source address port static translation rule. |

### Configure NAT66 UDP Protocol Internal Source Address Port Static Translation Rule

Table 21-6 Configure NAT66 UDP protocol internal source address port static translation rule

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 UDP protocol internal source address port static translation rule | **nat66 inside source static udp** { *local-IPv6 local-port* [ **vrf** *vrf-name* ] { *global-IPv6* \| **interface** *interface-name* } *global-port* [ **vrf** *vrf-name* ] \| **range** *local-IPv6 local-port-min local-port-max* [ **vrf** *vrf-name* ] { *global-IPv6* \| **interface** *interface-name* } *global-port-min global-port-max* [ **vrf** *vrf-name* ]} | Mandatory<br>By default, do not configure NAT66 UDP protocol internal source address port static translation rule. |

## 21.2.4. Configure NAT66 Internal Source Address Dynamic Translation Rule

The address mapping relationship between NAT66 internal network (inside) and external network (outside) is dynamic. The inside side actively initiates access. If the packet matches the translation rules, the source address will be converted to the address in the specified address pool or the address of the specified interface. It is mainly used in the scene of multiple hosts in the internal network accessing external Internet network.

### Configuration Conditions

Before configuring NAT66 dynamic translation rule, first complete the following task:

- Configure IPv6 ACL.
- Configure NAT66 address pool

### Configure NAT66 Internal Source Address Dynamic Translation Rule

Table 21-7 Configure NAT66 internal source address dynamic translation rules

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 internal source address dynamic translation rules | **nat66 inside source list** *access-list-name* [ **vrf** *vrf-name* ] { **pool** *pool-name* [ **overload**] \| **interface** *interface-name* **overload** } [ **reversible** { **full** \| **restrict** } ] [ **vrf** *vrf-name* ] | Mandatory<br>By default, do not configure NAT66 internal source address dynamic translation rules. |

### Configure NAT66 Address Pool

Table 21-8 Configure the NAT66 address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the NAT66 address pool | **nat66 pool** *pool-name* | Mandatory<br>By default, do not configure the address pool. |

### Configure Available Address Segment of NAT66 Address Pool

Table 21-9 Configure the available address segment of the NAT66 address pool

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the NAT66 address pool configuration mode | **nat66 pool** *pool-name* | Mandatory |

| Step | Command | Description |
|---|---|---|
| Configure the available address segment of the NAT66 address pool | **address** *start-ipv6 end-ipv6* [ **match interface** *interface* ] | Mandatory<br>By default, do not configure the available address segment of the NAT66 address pool. |

### 21.2.5. Configure NAT66 Internal Server Translation Rule

This rule specifies that the packet conforming to the specified ACL rules can use the specified address and port to convert the destination address and port of the packet. It is mainly used in the scenario of the external network actively accessing the internal server.

#### Configuration Conditions

Configure IPv6 ACL.

#### Configure NAT66 Internal Server Translation Rule

Table 21-10 Configure NAT66 internal server translation rule

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 internal server translation rule | **nat66 server list** *access-list-name* **[ vrf** *global-vrf-name* **] { *local-IPv6* \| network** *local-subnet/prefix-length* **} [** *local-port* **] [vrf** *local-vrf-name* **]** | Mandatory<br>By default, do not configure NAT66 internal server translation rule. |

### 21.2.6. Configure NAT66 Flow Log Function

After configuring the new flow log function of NAT66, the corresponding data log will be output when the translation table entry of NAT66 is generated. Configure the end flow log function of NAT66. When the translation table entry of NAT66 is aged normally or is cleared, the corresponding data log will be output.

#### Configuration Conditions

None

#### Configure NAT66 Flow Log Function

Table 21-11 Configure NAT66 flow log function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 flow log function | **nat66 logging security-data { connection-begin \| connection-end } *** | Mandatory<br>By default, do not configure NAT66 flow log function. |

## 21.2.7. **Configure NAT66 ALG Switch**

The NAT66 ALG switch is used to enable the processing function of application layer message. At present, the application layer protocols supported by NAT66 include FTP, TFTP, HTTP and SIP.

### Configuration Conditions

None

### Configure NAT66 ALG Switch

Table 21-12 Configure NAT66 ALG switch

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure NAT66 ALG switch | **nat66 alg** { **ftp \| tftp \| http \| sip \| all** } | Mandatory<br>By default, do not configure NAT66 ALG, and the ALG function is not enabled. |

### 21.2.8. NAT66 Monitoring and Maintaining

Table 21-13 NAT66 monitoring and maintaining

| Command | Description |
|---|---|
| **clear nat66 statistics** | Clear all statistics information of NAT66 |
| **show nat66 statistics [verbose]** | Display the NAT66 global and board statistics information |
| **show nat66 translation lpu** *lpu-unit* [ **proto** { **icmp \| tcp \| udp \| raw-ip** } ] [ **v6src** *ipv6* ] [ **v6dst** *ipv6* ] | Display the NAT66 translation table entry |
| **show running-config nat66** | Display all configuration information of NAT66 |

## 21.3. NAT66 Typical Configuration Example

### 21.3.1. Configure Internal Source NAT66 Static Address/Port Translation

#### Network Requirements

- PC1 and PC2 are internal private network hosts, web server is the web server on the public network, Device is the NAT66 device, gigabitethernet0 is the NAT66 internal interface, gigabitethernet1 is the NAT66 external interface, and internal source NAT66 static address/port translation is configured.
- The internal private network PC1 can successfully access the web server, the source address of the packet is converted from private address 2018::2 to public address 2001:1::5, PC2 can successfully access the web server using source port 1024, the source address of the packet is converted from private address 2018::3 to public address 2001:1::6, and the source port 1024 is converted to 2000.

#### Network Topology

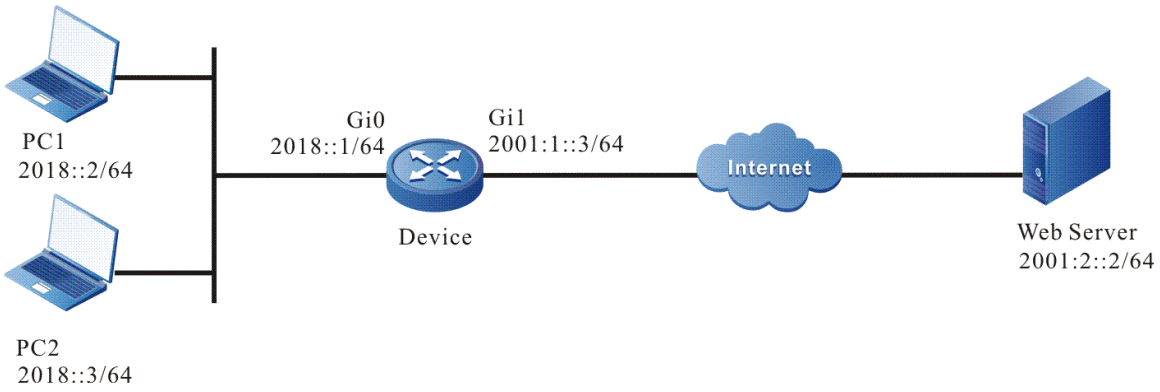



Figure 21-1Networking of configuring internal source NAT66 static address/port translation

### Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT66 interface attributes and NAT66 translation rule of Device.

#Configure interface gigabitethernet0 as the inside interface of NAT66.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat66 inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT66.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat66 outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the internal source nat66 static address / port conversion rules, convert the internal private network address 2018::2 to the public network address 2001:1::5, and convert the internal private network address 2018::3 and port 1024 to the public network address 2001:1::6 and port 2000.

```
Device(config)#nat66 inside source static 2018::2 2001:1::5
Device(config)#nat66 inside source static tcp 2018::3 1024 2001:1::6 2000
```

**Step 3:** Check the result.

#When PC1 of the internal private network accesses the web server, you can see the NAT translation table item with type SAT on Device. When PC2 accesses the web server using source port 1024, you can see the NAT translation table item with type SPT on Device.

```
Device#show nat66 translation
 Type    Pro          Local source:port                    Global source:port
lifetime
                  Local destination:port          Global destination:port
 ----    ---        ----------------------              ----------------------
--------
 SPT    TCP        [2018::3]:1024                [2001:1::6]:2000              3600
                   [2001:2::2]:80                [2001:2::2]: 80
 SAT    TCP        [2018::2]:1024                [2001:1::5]:1024              3600
                   [2001:2::2]: 80               [2001:2::2]: 80


 Valid/Total: 2/2
```

## 21.3.2. Configure Internal Source NAT66 Dynamic Address Translation

### Network Requirements

- PC1 and PC2 are internal private network hosts, web server is the web server on the public network, Device is the NAT66 device, gigabitethernet0 is the NAT66 internal

interface, gigabitethernet1 is the NAT66 external interface, and the internal source NAT66 dynamic address translation is configured.

- The users in the 2018::/64 network segment of the internal private network can successfully access the web server, and the public network addresses used are 2001:1:: 4 and 2001:1:: 5.

## Network Topology

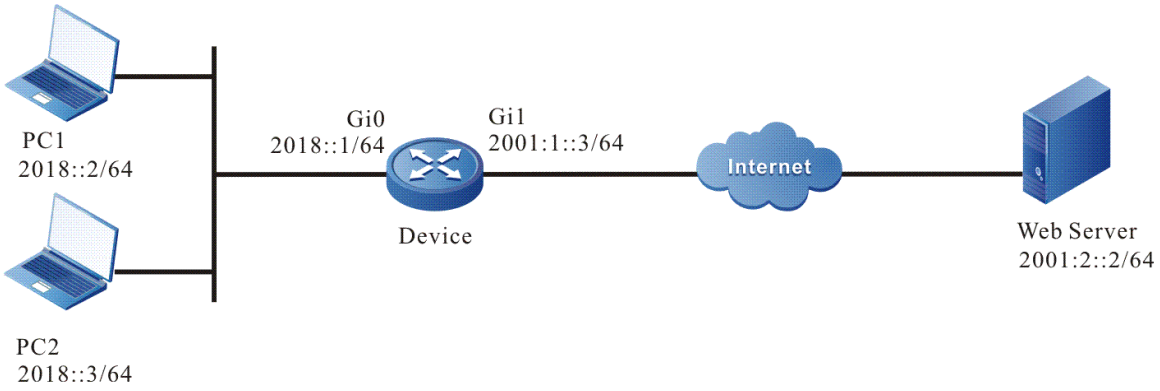



Figure 21-2 Networking of configuring internal source NAT66 dynamic address translation

## Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT66 interface attributes and NAT66 translation rule of Device.

#Configure interface gigabitethernet0 as the inside interface of NAT66.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat66 inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT66.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat66 outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT66 address pool pool1, which contains two public network addresses 2001:1::4 and 2001:1::5.

```
Device(config)#nat66 pool pool1
Device (config-nat66-pool)#address 2001:1::4 2001:1::5
Device (config-nat66-pool)#exit
```

#Configure the access control list 7001, only permitting the PCs in the 2018::/64 network segment of the internal private network to access the web server.

```
Device(config)#ipv6 access-list extended 7001
Device(config-ext-nacl)# permit ipv6 2018::/64 any
Device(config-ext-nacl)#exit
```

#Configure the internal source NAT66 dynamic address translation rules to convert the internal private network addresses 2018::2 and 2018::3 into public network addresses 2001:1::4 and 2001:1::5 respectively.

```
Device(config)#nat66 inside source list 7001 pool pool1
```

**Step 3:** Check the result.

#When PC1 and PC2 of the private network access the web server, you can see the NAT66 translation table entry on Device.

```
Device #show nat66 translation
 Type    Pro         Local source:port                    Global source:port
lifetime
                 Local destination:port              Global destination:port
 ----    ---      ----------------------              ----------------------
--------
 NAT     TCP      [2018::3]:1024                      [2001:1::5]:1024          3600
                  [2001:2::2]:80                      [2001:2::2]:80
 NAT     TCP      [2018::2]:1024                      [2001:1::4]:1024          3600
                  [2001:2::2]:80                      [2001:2::2]:80


 Valid/Total: 2/2
```

## 21.3.3. Configure Internal Source NAT66 Dynamic Port Translation

### Network Requirements

- PC1 and PC2 are internal private network hosts, web server is the web server on the public network, Device is the NAT66 device, gigabitethernet0 is the NAT66 internal interface, gigabitethernet1 is the NAT66 external interface, and the internal source NAT66 dynamic port translation is configured.
- The users in the 2018::/64 network segment of the internal private network can successfully access the web server, and multiplex the same public network address by mapping the TCP port number. The public network address used is 2001:1::4.

### Network Topology

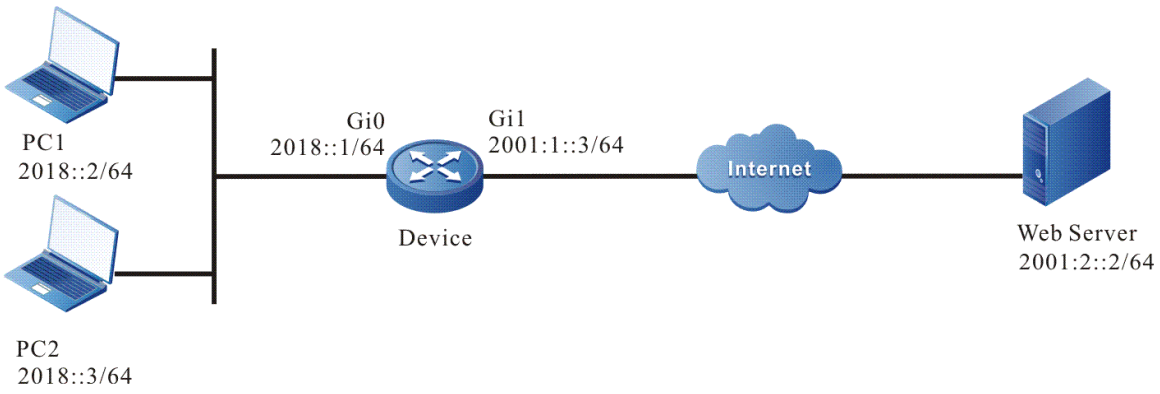



Figure 21-3 Networking of configuring the internal source NAT66 dynamic port translation

## Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT66 interface attributes and NAT66 translation rule of Device.

#Configure interface gigabitethernet0 as the inside interface of NAT66.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat66 inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT66.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat66 outside
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT66 address pool pool1, which only contains one public network address 2001:1::4.

```
Device(config)#nat66 pool pool1
Device (config-nat66-pool)#address 2001:1::4 2001:1::4
Device (config-nat66-pool)#exit
```

#Configure the access control list 7001, only permitting the PCs in the 2018:: / 64 network segment of the internal private network to access the web server.

```
Device(config)#ipv6 access-list extended 7001
Device(config-ext-nacl)# permit ipv6 2018::/64 any
Device(config-ext-nacl)#exit
```

#Configure the internal source NAT66 dynamic port translation rules to convert the internal private network addresses 2018::2 and 2018::3 to the public network address 2001:1::4.

```
Device(config)# nat66 inside source list 7001 pool poo11 overload
```

**Step 3:** Check the result.

#When PC1 and PC2 of the internal private network access the web server, you can see the NAT66 translation table entry on the device.

```
Device #show nat66 translation
 Type    Pro          Local source:port                    Global source:port
lifetime
                Local destination:port              Global destination:port
 ----   ---      ---------------------              ----------------------
--------
 PAT    TCP      [2018::3]:1024                     [2001:1::4]:10001          3598
                 [2001:2::2]:80                     [2001:2::2]:80
 PAT    TCP      [2018::2]:1024                     [2001:1::4]:10002          3598
                 [2001:2::2]:80                     [2001:2::2]:80
```

Valid/Total: 2/2

## 21.3.4. Configure NAT66 Internal Server Translation

### Network Requirements

- PC1 and PC2 are external hosts, Server1 is a web server on the intranet, the server IPv6 address is 2018::2, Device is a NAT66 device connected to gigabitethernet1 and gigabitethernet2, respectively connected to two operator lines of Rostelecom, gigabitethernet0 is configured as NAT66 inside interface, and gigabitethernet1 and gigabitethernet2 are configured as NAT66 outside interface.
- PC1 accesses 2001:1::2 and PC2 accesses 2001:3::3. After nat66 translation, replace the destination address with IPv6 address as 2018::2. Finally, both can access and reach the web server.

### Network Topology

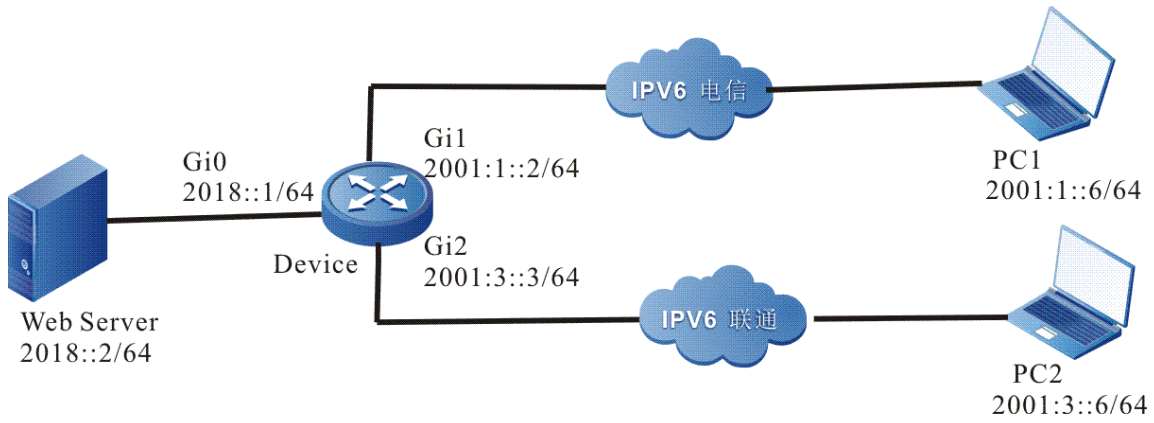



Figure 21-4 Networking of configuring NAT66 internal server translation

### Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure the NAT66 interface attributes and NAT66 translation rule of Device.

#Configure interface gigabitethernet0 as the inside interface of NAT66.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat66 inside
Device(config-if-gigabitethernet0)#exit
```

#Configure interface gigabitethernet1 as the outside interface of NAT66.

```
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#nat66 outside
Device(config-if-gigabitethernet1)#exit
```

#Configure interface gigabitethernet2 as the outside interface of NAT66.

```
Device(config)#interface gigabitethernet 2
Device(config-if-gigabitethernet2)#nat66 outside
```

```
Device(config-if-gigabitethernet2)#exit
```

#Configure the access control list server. The source address is any and the destination address is the interface address of the two exits of Device.

```
Device (config)#ipv6 access-list extended server
Device (config-v6-list)# permit ipv6 any host 2001:1::2
Device (config-v6-list)# permit ipv6 any host 2001:3::3
Device (config-ext-nacl)#exit
```

#Configure NAT66 internal server translation rules to convert the destination public network addresses 2001:1::2 and 2001:3::3 into private server addresses 2018::2.

```
Device(config)# nat66 server list server 2018::2
```

**Step 3:** Check the result.

#The public network users PC1 and PC2 connected to Rostelecom respectively access the web server. The access destination addresses are the addresses 2001:1::2 and 2001:3::3 of the outgoing interfaces of the two operators connected to Device. The NAT66 translation table entry can be seen on the Device.

```
Device #show nat66 translation
 Type    Pro         Local source:port                    Global source:port
lifetime
                 Local destination:port              Global destination:port
 ----   ---       ----------------------              ----------------------
--------
 SERVER TCP       [2018::2]:80                 [2001:1::2]:80              3600
                  [2001:1::6]:1024             [2001:1::6]:1024
 SERVER TCP       [2018::2]:80                 [2001:3::3]:80              3543
                  [2001:3::6]:1024             [2001:3::6]:1024

 Valid/Total: 2/2
```

# 22. BANDWIDTH LIMIT

## 22.1. Overview

Bandwidth limit provides IP-based bandwidth guarantee and constraint for intranet users. When the traffic of intranet users is connected to Internet through NAT devices, the resources provided to the intranet users are usually limited–for example, the bandwidth provided by ISP to users is limited and the address translation quantity provided by NAT devices for users is also limited. With the increasing application of users, more and more bandwidth is needed. Therefore, in case of the limited resources, provide every user with the fair resources as possible, and prevent one or some users from occupying too many resources, which leads to the failure of other users to get reasonable resources. IP-based bandwidth limit mainly provides the bandwidth limit based on IP address for NAT devices, which prevents some users from occupying too much bandwidth, which makes other users unable to allocate bandwidth resources. At the same time, IP-based bandwidth limit can also be used in non-NAT scenarios.

## 22.2. Bandwidth Limit Function Configuration

Table 22-1 Bandwidth limit function configuration list

| Configuration Tasks | |
|---|---|
| Configure uplink traffic bandwidth limit | Configure the total bandwidth of the uplink bandwidth |
| | Configure the bandwidth of the uplink user |
| Configure downlink traffic bandwidth limit | Configure the total bandwidth of the downlink bandwidth |
| | Configure the bandwidth of the downlink user |

### 22.2.1. Configure Uplink Traffic Bandwidth Limit

#### Configuration Conditions

None

### Configure Uplink Traffic Bandwidth Limit Function

Table 22-2 Configure the uplink traffic bandwidth limit function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the total uplink bandwidth of the interface | **bandwidth-limit total ipv4 egress** *bw-value* | Mandatory<br>By default, the interface is not configured with the total uplink bandwidth. |
| Configure the uplink bandwidth of the user | **bandwidth-limit ipv4 address { range** *start-address end-address* **} egress** *cir_value bir_value* | Mandatory<br>By default, the user uplink bandwidth is not configured for the interface<br>The total uplink bandwidth of the interface must be configured before configuring the user's uplink bandwidth. |

## 22.2.2. Configure Downlink Traffic Bandwidth Limit

### Configuration Conditions

None

### Configure Downlink Traffic Bandwidth Function

Table 22-3 Configure the downlink traffic bandwidth function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the total downlink bandwidth of the interface | **bandwidth-limit total ipv4 ingress** *bw-value* | Mandatory<br>By default, the interace is not configured with the total downlink bandwidth. |
| Configure the downlink bandwidth of the user | **bandwidth-limit ipv4 address { range** *start-address end-address* **} ingress** *cir_value bir_value* | Mandatory<br>By default, the user downlink bandwidth is not configured for the interface<br>The total downlink bandwidth of the interface must be configured before configuring the downlink bandwidth of the user. |

## 22.3. Bandwidth Limit Typical Configuration Example

### 22.3.1. Configure IP-based Bandwidth Limit

#### Network Requirements

- Device is a NAT device, and the interface gigabitethernet0 is connected to the Intranet; Interface gigabitethernet1 is connects to the Internet network.
- IP-based bandwidth restriction is applied to the data flow sent and received by interface gigabitethernet1 on Device: limit the 20Mbps total interface bandwidth for the uplink traffic (sending data) of interface gigabitethernet1, and limit the 100Mbps total interface bandwidth for the downlink traffic (receiving data) of interface gigabitethernet1; perform the bandwidth limit for the users of NAT intranet host network segment 1 (source IP address range: 192.168.1.1–192.168.1.10). Ensure 500Kbps uplink traffic and 1Mbps downlink traffic for each IP; perform the bandwidth limit for the users of NAT intranet host network segment 2 (source IP address range: 192.168.2.1–192.168.2.10), and ensure 1Mbps uplink traffic and 2Mbps downlink traffic for each IP; the users of

network segment 1 and network segment 2 can share the free bandwidth of the total bandwidth of the interface.

## Network Topology

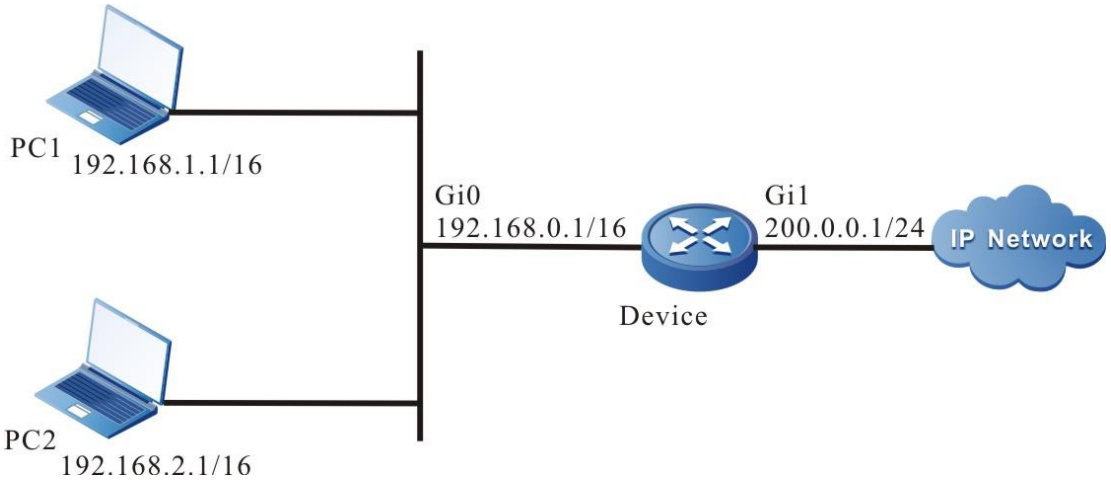



Figure 22-1 Networking of configuring IP-based bandwidth limit

## Configuration Steps

**Step 1:** Configure the IP address and route of the interface (omitted).

**Step 2:** Configure Device NAT.

#Configure the NAT interface attributes and NAT translation rules of Device.

```
Device#configure terminal
Device(config)#interface gigabitethernet0
Device(config-if-gigabitethernet0)#ip nat inside
Device(config-if-gigabitethernet0)#exit
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)#ip nat outside
Device(config-if-gigabitethernet1)#exit
Device(config)#ip nat pool pool1 200.0.0.1 200.0.0.1 netmask 255.255.255.0
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip 192.168.0.0 0.0.255.255 any
Device(config-ext-nacl)#exit
Device(config)#ip nat inside source list 1001 pool pool1 overload
```

**Step 3:** Configure the IP-based bandwidth limit rules of Device.

#Configure the total interface bandwidth of the bandwidth limit.

```
Device#configure terminal
Device(config)#interface gigabitethernet1
Device(config-if-gigabitethernet1)# bandwidth-limit total ipv4 egress 20000
Device(config-if-gigabitethernet1)# bandwidth-limit total ipv4 ingress 100000
```

#Configure the user bandwidth rules of the bandwidth limit.

```
Device(config-if-gigabitethernet1)#bandwidth-limit ipv4 address range 192.168.1.1 192.168.1.10 egress 500 20000
Device(config-if-gigabitethernet1)#bandwidth-limit ipv4 address range 192.168.1.1 192.168.1.10 ingress 1000 100000
Device(config-if-gigabitethernet1)#bandwidth-limit ipv4 address range 192.168.2.1 192.168.2.10 egress 1000 20000
Device(config-if-gigabitethernet1)#bandwidth-limit ipv4 address range 192.168.2.1 192.168.2.10 ingress 2000 100000
Device(config-if-gigabitethernet1)#end
```

**Step 4:** When the Device interface gigabitethernet1 has uplink and downlink traffic of intranet users, the bandwidth limit takes effect. You can view the user traffic statistics on the device.

```
Device#show bandwidth-limit ipv4 user interface gigabitethernet1

bandwidth user entity:
Source address      bw (kbps)              Recv:pps bps | pps bps              Send:pps bps | pps bps
--------------      --------------         ----------------------              ----------------------
192.168.1.1                    500<->20000                                     0:0|0:0 11877:10452053|11761:10350237
192.168.2.1                    1000<->20000                                    0:0|0:0 10874:9569413|10784:9490096
192.168.1.1         1000<->100000          25365:22321786|25373:22328606       0:0|0:0
192.168.2.1         2000<->100000          25365:22321786|25373:22328460       0:0|0:0
```

# 23. CONNECTION LIMIT

## 23.1. Overview

Connection management is a common basic function abstracted to facilitate the implementation of NAT, ASPF, IPFIX and other security related services based on DPI. This function is classified according to transport layer packets (TCP, UDP, ICMP, ICMPv6, etc.), abstracted into connections according to their interaction, and maintains and ages the connection status according to the packet information of the initiator and the responder. It supports multiple business modules to process the same connection. In order to prevent too many connections, it supports limiting the number of connections based on ACL classification.

The working principle of connection management is to analyze the transport layer packet information, extract its source/destination IP address, source/destination port number (ICMP takes the ID value), protocol number and VRF to create a connection, and maintain the current connection status according to whether it is the initiator or responder (additional flag information of the TCP packet).

As a public function, connection management abstracts connections and tracks, ages and limits connections. It must be combined with specific services to play security and other functions, and only supports unicast.

## 23.2. Connection Limit Function Configuration

Table 23-1 Connection limit function configuration list

| **Configuration Tasks** | |
|---|---|
| Configure the aging time function of the connection | Configure the aging time function of the connection |
| Configure connection limit policy function | Configure the connection limit rules |
| Configure the global application connection limit policy function | Configure the global application connection limit policy function |
| Configure the maximum specification function of the connection | Configure the maximum specification function of the connection |
| Configure forwarding action function when failed to get connection | When failed to configure the service module to get the connection, refer to the forwarding action function. |

### 23.2.1. **Configure Connection Aging Time Function**

By default, the connection aging time of the protocol status is:

TCP-SYN: 30s;

TCP-EST: 3600s;

TCP-FIN: 30s;

TCP-TIME-WAIT: 2s;
TCP-CLOSE: 2s;
UDP-OPEN: 30s;
UDP-READY: 60s;
ICMP: 60s;
ICMP-ERROR: 30s;
RAWIP-OPEN: 30s;
RAWIP-READY: 60s;

### Configuration Conditions

None

### Configure Age Time of Connection

Table 23-2 Configure the aging time of the connection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the aging time of the connection | **connection age-timeout { tcp-syn \| tcp-fin \| tcp-est \| tcp-close \| tcp-time-wait \| udp-open \| udp-ready \| rawip-open \| rawip-ready \| icmp \| icmp-error } [** *lifetime* **\| never ]** | Mandatory |

## 23.2.2. Configure Connection Limit Policy Function

A connection restriction policy can define multiple connection restriction rules. Each connection restriction rule formulates a connection restriction user range. The new connections of the user belonging to this range will be restricted by the policies configured by this rule. When the number of the connections of a certain type reaches the upper limit, the new connection of this type will be rejected. The new connection cannot be permitted until the number of connections of this type is lower than the lower limit.

Currently, it only supports limiting the user range according to ACL. The types of the restriction rules include:

per-source: Limit by source IP address
per-service: Limit by the service (same transport layer protocol and port);

By default, limit all users specified by the service ACL.

When matching the connections to be established with the restriction rule, it will be matched according to the size order of the rule serial number (the smaller serial number, the earlier matching). Therefore, if there are special requirements for the matching order, it needs to be considered as a whole.

The current restriction policy only supports global application.

### Configuration Conditions

None

### Configure Connection Restriction Policy

Table 23-3 Configure the connection restriction policy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the connection restriction policy | **connection limit { ipv6-policy \| policy }** *policy-name* | Mandatory |

### Configure Connection Restriction Rule

Table 23-4 Configure the connection restriction rule

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the connection restriction policy configuration mode | **connection limit { ipv6-policy \| policy }** *policy-name* | Mandatory |
| Configure the connection restriction rule | **rule** *rule-id* { **acl** \| **ipv6-acl** } { *acl-number* \| *acl-name* } [ **per-source \| per-service** ] **count** *min-count max-count* | Mandatory |

## 23.2.3. Configure Global Application Connection Restriction Policy Function

### Configuration Conditions

None

### Configure Global Application Connection Restriction Policy

Table 23-5 Configure the global application connection restriction policy

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the global application connection restriction policy | **connection limit apply global { ipv6-policy \| policy }** *policy-name* | Mandatory |

## 23.2.4. Configure the Max. Specification Function of Connection

### Configuration Conditions

None

### Configure Max. Specification of Connection

Table 23-6 Configure the maximum specification of the connection

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the maximum specification of the connection | **connection { ipv4 \| ipv6 } max-entries** *entries-size* | Mandatory |

## 23.2.5. Configure Forarding Action Function When Failed to Get Connection Node

When a service (such as NAT and ASPF) fails to obtain/create a connection, it can further obtain the forwarding behavior policy after the failure.

### Configuration Conditions

None

### Configure Forwarding Action after Failed to Get Connection Node

Table 23-7 Configure the forwarding action after failed to get connection node

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Configure the forwarding action after failed to get connection node | **connection policy create-failure drop** | Mandatory。<br>By default, forward. After configuration, drop. |

## 23.2.6. Connection Restriction Monitoring and Maintaining

Table 23-8 Connection restriction monitoring and maintaining

| Command | Description |
|---|---|
| **clear connection { all \| { ipv4 \| ipv6 } [ proto icmp \| raw-ip \| tcp \| udp ] [ src-ip** *src-ipaddress* **] [ dst-ip** *dst-ipaddress* **] [ src-port** *src-port* **] [ dst-port** *dst-port* **] [ vrf** *vrf-name* **] \| limit statistics }** | Clear the connection table entry/connection restriction statistics, and the 6-tuple filter can be used in any combination |
| **show connection all statistics** | View the statistics information of the connection table |
| **show connection limit** { **ipv6-policy \| policy** } [ *policy name* ] | Display the configuration information of the connection restriction policy |
| **show connection limit statistics** | Display the statistics information of the connection restriction |
| **show connection limit [ table \| ipv6-table ] [ source** *src-addr* **\| service** *port-number* **]** | Display the connection restriction table entry information |
| **show connection { ipv4 \| ipv6 } { table \| expectant }** | Display the ipv4/ipv6 connection table entry information |

## 23.3. Connection Restriction Typical Configuration Example

### 23.3.1. Configure Globally Referencing IPv4 and IPv6 Connection Restriction Policy

#### Network Requirements

- PC1 and PC2 are internal hosts, server is a server on the public network, Device is a NAT64 device, NAT64 translation rules are configured, and the route is reachable.
- In order to prevent the system resources in the application environment from being exhausted by a few users, allocate the system resources reasonably, and configure the connection restriction policies of IPv4 and IPv6 on Device to restrict the connections in the NAT64 service environment. When the number of connections exceeds the maximum upper limit, it is not allowed to create new connections, and when the number of connections falls below the lower limit, it is allowed to create new connections.

#### Network Topology

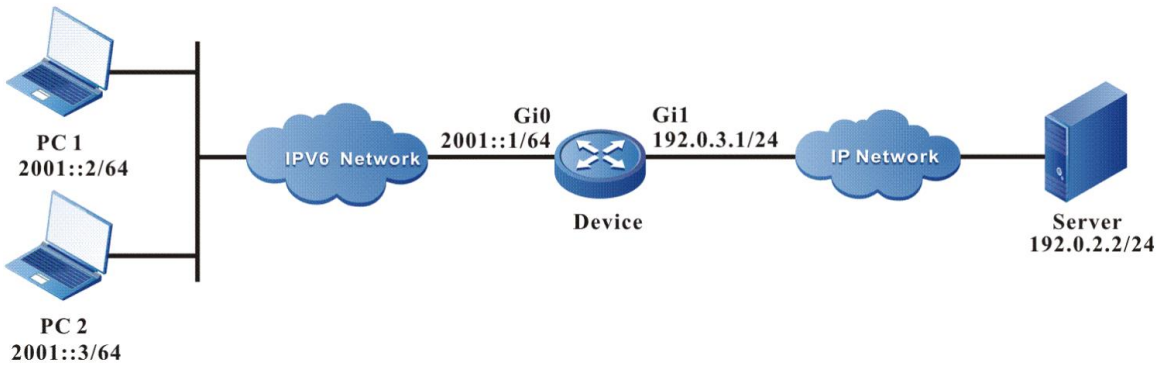



Figure 23-1 Networking of globally referencing IPv4 and IPv6 connection restriction policy

#### Configuration Steps

**Step 1:** Configure the IP address of the interface (omitted).

**Step 2:** Configure the IPv6 route of the host PC1 and PC2.

On PC1 and PC2, configure the static route to the prefix 2001:1::/96.

**Step 3:** Configure the NAT64 prefix, NAT64 interface attributesm and NAT64 translation rules of Device.

#Configure the NAT64 prefix.

```
Device#configure terminal
Device(config)#nat64 prefix 2001:1::/96
```

#Enable the NAT64 function on interface gigabitethernet 0.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#nat64 enable
Device(config-if-gigabitethernet0)#exit
```

#Enable the NAT64 function on interface gigabitethernet 1.

```
Device(config)#interface gigabitethernet 1
```

```
Device(config-if-gigabitethernet1)#nat64 enable
Device(config-if-gigabitethernet1)#exit
```

#Configure the NAT64 static address translation rule.

```
Device(config)#nat64 static 192.0.3.2 2001::2
Device(config)#nat64 static 192.0.3.3 2001::3
```

**Step 4:** Configure the IPv4 and IPv6 connection restriction policy, and reference globally.

#Configure the IPv6 access control list 7001, only permitting the packet with source address 2001::2 to pass.

```
Device(config)#ipv6 access-list extended 7001
Device(config-v6-list)#permit ipv6 host 2001::2 any
Device(config-v6-list)#exit
```

#Configure the IPv6 connection restriction policy. The matching method of connection restriction rules specified in the policy is per-service. The maximum value of restriction is 100 and the minimum value is 80. The policy is referenced globally.

```
Device(config)#connection limit ipv6-policy testv6
Device(config-connlmt-ipv6-policy)#rule 1 ipv6-acl 7001 per-service count 80 100
Device(config-connlmt-ipv6-policy)#exit
Device(config)#connection limit apply global ipv6-policy testv6
```

#Configure IPv4 access control list 1001, only permitting the packet with source address 192.0.2.2 and destination address 192.0.3.3 to pass.

```
Device(config)#ip access-list extended 1001
Device(config-ext-nacl)#permit ip host 192.0.2.2 host 192.0.3.3
Device(config-ext-nacl)#exit
```

#Configure the IPv4 connection restriction policy. The matching method of the connection restriction rules specified in the policy is per-source. The maximum value of the restriction is 200 and the minimum value is 150. The policy is referenced globally.

```
Device(config)#connection limit policy testv4
Device(config-connlmt-policy)#rule 1 acl 1001 per-source count 150 200
Device(config-connlmt-policy)#exit
Device(config)#connection limit apply global policy testv4
```

**Step 5:** Check the result.

#On Device, execute the command **show connection limit policy** and **show connection limit ipv6-policy** to view the configuration information of the connection restriction policy, as follows:

```
Device#show connection limit policy
No. Policy                    Rule Rule-Type LoThres   HiThres   Acl
--- ------------------------------ ---- --------- --------- --------- ---------
----------------------
1   testv4                    1    Src      150    200       1001
```

```
Device#show connection limit ipv6-policy
No. Policy                         Rule Rule-Type LoThres   HiThres   Acl
--- ------------------------------ ---- --------- --------- --------- ---------
----------------------
1   testv6                         1    Dst-Port  80       100       7001
```

#PC1 initiates TCP and UDP IPv6 connections to the extranet server. The source address is 2001::2 and the destination address is 2001:1::C000:0202. The source port starts from 10001 and increases by 110 in steps of 1. The TCP connection destination port is 80 and the UDP connection destination port is 53. Execute the command **show connection limit ipv6-table** on Device to view the IPv6 connection restriction table item.

```
Device#show connection limit ipv6-table
connection limit table
Rule Rule-Type No.    Src-Addr/Dst-Port                    Count     New flag
---- --------- ------- ------------------------------------ ---------- ----
----
1    Dst-Port  1      tcp/80                        100       Deny
               2      udp/53                        100       Deny
```

Print the log information of IPv6 connection limit reaching the upper limit on Device.

```
Sep  23  2019  08:05:14  Device  %CONN-LIMIT6_EXCEED-5:Dst-Port[udp/53]
connections more than HiThres 100, cann't create new connections.
Sep  23  2019  08:05:14  Device  %CONN-LIMIT6_EXCEED-5:Dst-Port[tcp/80]
connections more than HiThres 100, cann't create new connections.
```

When the number of IPv6 connections on the device ages below 80, print the log information that the connection limit is restored to the lower limit. At this time, the device can continue to create new IPv6 connections.

```
Sep  23  2019  08:12:27  Device  %CONN-LIMIT6_RESUME-5:Dst-Port[tcp/80]
connections less than LoThres 80.
Sep  23  2019  08:12:27  Device  %CONN-LIMIT6_RESUME-5:Dst-Port[udp/53]
connections less than LoThres 80.
```

#The extranet server initiates an IPv4 connection to the intranet host PC, the source address is 192.0.2.2, the destination address is 192.0.3.3, the source port starts from 10001, increases by 210 in steps of 1, and the destination port is 1024. Execute the command **show connection limit table** to view the IPv4 connection limit table entries on the device.

```
Device#show connection limit table
connection limit table
Rule Rule-Type No.    Src-Addr/Dst-Port                    Count     New flag
---- --------- ------- ------------------------------------ ---------- ----
----
1    Src       1       192.0.2.2                     200       Deny
```

Print the log information that the IPv4 connection limit has reached the upper limit on the device.

```
Sep  23  2019  08:22:52  Device  %CONN-LIMIT4_EXCEED-5:Source[192.0.2.2]
connections more than HiThres 200, cann't create new connections.
```

When the number of IPv4 connections on the device ages below 150, print the log information that the connection limit is restored to the lower limit. At this time, the device can continue to create new IPv4 connections.

```
Sep 23 2019 08:28:27 Device %CONN-LIMIT4_RESUME-5:Source[192.0.2.2]
connections less than LoThres 150.
```

# 24. TPC

## 24.1. Overview

Transport Payload Compression (TPC) is a packet compression technology developed by Qtech. For the transport layer packet to be sent, compress the payload part, and the length of the payload part is reduced after compression, so as to save bandwidth. This function is mainly deployed on the limited WAN port bandwidth, so as to realize more data transmission per unit time and improve the WAN port bandwidth utilization, so as to accelerate the WAN communication.

The TPC function is based on the interface, and the enable/disable is divided into three modes:

1. The transmitter compression and receiver decompression are enabled/disabled at the same time;
2. Only enable/disable the sender compression;
3. Only enable/disable the receiver decompression.

During compression, the minimum length of the transport layer load can be set. Compression processing can be carried out only when the length of the transport layer load is greater than the set minimum length, which can provide an adjustable means for communication acceleration and device load balance. In addition, ACL matching is also supported during compression. Only transport layer packets that match the set ACL rules will compress their load.

## 24.2. TPC Function Configuration

Table 24-1 TPC function configuration list

| Configuration Tasks | |
|---|---|
| Configure the TPC basic functions | Enable the transport layer payload compression function |
| Configure TPC extended functions | Configure the minimum length of the transport layer load to compress |
| | Configure the ACL matching the transport layer packet to be compressed |
| | Configure the recalculation function of transport layer packet checksum |

### 24.2.1. Configure TPC Basic Functions

#### Configuration Conditions

None

### Enable Transport Layer Payload Compression Function

Table 24-2 Enable the transport layer payload compression function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the transport layer payload compression function | **tpc enable** [**compress \| decompress**] | Mandatory<br>By default, the interface does not enable the transport layer payload compression function. |

**Note:**

- You can only enable the compression function or only enable the decompression function through parameters.

## 24.2.2. Configure TPC Extended Function

### Configuration Conditions

Before configuring the TPC extended function, first complete the following task:

- Enable the transport layer payload compression function on the interface

### Configure Minimum Length of Transport Layer Payload to Be Compressed

Table 24-3 Configure the minimum length of the transport layer payload to be compressed

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the minimum length of the transport layer payload to be compressed | **tpc compress minimum length** *length* | Optional<br>By default, the minimum length of the transport layer payload to be compressed is 256 bytes. |

### Configure Matching ACL of Transport Layer Packet to Be Compressed

Table 24-4 Configure the matching ACL of the transport layer packet to be compressed

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the matching ACL of the transport layer packet to be compressed | **tpc ip access-list** {*acl-num* \| *acl-name*} | Optional<br>By default, do not configure the matching ACL of the transport layer packet to be compressed |

**Note:**

- ACL only supports matching the IPv4 packet.

### Configure the Re-calculation Function of Transport Layer Packet Checksum

Table 24-5 Configure the re-calculation function of the transport layer packet checksum

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the re-calculation function of the transport layer packet checksum | **tpc rechecksum** | Optional<br>By default, do not enable the re-calculation function of the transport layer packet checksum. |

### 24.2.3. **TPC Monitoring and Maintaining**

Table 24-6 TPC monitoring and maintaining

| **Command** | **Description** |
|---|---|
| **show tpc interface** {*interface-name*} | Display the related information of the transport layer payload compression on the specified interface |
| **clear tpc statistics** {**global** \| **interface** *interface-name*} | Clear the transport layer payload compression statistics information globally or of the specified interface |

# 25. FEC

## 25.1. Overview

Forward Error Correcting (FEC) is a technology that provides packet loss guarantee for real-time audio and video network flow (UDP flow). By adding a small amount of redundant packets at the sending end, in case of line packet loss, the original data can also be completely recovered through the remaining original packets and redundant packets to realize the lossless transmission.

The FEC function is based on the interface, and the enable/disable is divided into three modes:

1. The coding at the sending end and the decoding at the receiving end are enabled/disabled at the same time;
2. Only enable/disable coding at the sending end;
3. Only enable/disable decoding at the receiver.

ACL shall be set for filtering during coding to avoid code redundancy of irrelevant packets and waste of bandwidth. The decoding end also supports ACL matching. Only packets that match the set ACL rules will be decoded.

## 25.2. FEC Function Configuration

Table 25-1 FEC function configuration list

| Configuration Tasks | |
|---|---|
| Configure the FEC basic function | Enable the FEC function |
| Configure the FEC extended function | Configure the ACL to be matched by the coding the packet |
| | Configure the ACL to be matched by the decoding packet |

### 25.2.1. Configure FEC Basic Function

#### Configuration Conditions

None

### Enable the FEC Function

Table 25-2 Enable the FEC function

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Enable the FEC function | **fec enable** [**code \| decode**] | Mandatory<br>By default, the interface does not enable the FEC function. |

**Note:**

- You can enable only the encoding function or the decoding function through parameters.
- The encoding function will not take effect until the ACL required for encoding is configured.

## 25.2.2. Configure FEC Extended Function

### Configuration Conditions

Before configuring the FEC extended function, first complete the following task:

- Enable the FEC function on the interface

### Configure ACL to Be Matched by the Coding Packet

Table 25-3 Configure the ACL to be matched by the coding packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |

| Step | Command | Description |
|---|---|---|
| Configure the ACL to be matched by the coding packet | **fec code ip access-list** {*acl-num* \| *acl-name*} | Mandatory<br>By default, do not configure the ACL to be matched by the coding packet. |

**Note:**

- ACL only supports matching the IPv4 packet.
- The decoding function takes effect only after the ACL to be matched by the coding packet is configured.

### Configure ACL to Be Matched by the decoding Packet

Table 25-4 Configure the ACL to be matched by the decoding packet

| Step | Command | Description |
|---|---|---|
| Enter the global configuration mode | **configure terminal** | - |
| Enter the interface configuration mode | **interface** *interface-name* | - |
| Configure the ACL to be matched by the decoding packet | **fec decode ip access-list** {*acl-num* \| *acl-name*} | Optional<br>By default, do not configure the ACL to be matched by the decoding packet. |

## 25.2.3. FEC Monitoring and Maintaining

Table 25-5 FEC monitoring and maintaining

| Command | Description |
|---|---|
| **show fec interface** {*interface-name*} | Display the FEC information on the specified interface |
| **clear fec statistics** {**global** \| **interface** *interface-name*} | Clear the FEC statistics information globally or of the specified interface |

# 26. ОБЩАЯ ИНФОРМАЦИЯ

## 26.1. Замечания и предложения

Мы всегда стремимся улучшить нашу документацию и помочь вам работать лучше, поэтому мы хотим услышать вас. Мы всегда рады обратной связи, в особенности:

- ошибки в содержании, непонятные или противоречащие места в тексте;
- идеи по улучшению документации, чтобы находить информацию быстрее;
- неработающие ссылки и замечания к навигации по документу.

Если вы хотите написать нам по поводу данного документа, то используйте, пожалуйста, форму обратной связи на qtech.ru.

## 26.2. Гарантия и сервис

Процедура и необходимые действия по вопросам гарантии описаны на сайте QTECH в разделе «Поддержка» −> «Гарантийное обслуживание».

Ознакомиться с информацией по вопросам тестирования оборудования можно на сайте QTECH в разделе «Поддержка» −> «Взять оборудование на тест».

Вы можете написать напрямую в службу сервиса по электронной почте sc@qtech.ru.

## 26.3. Техническая поддержка

Если вам необходимо содействие в вопросах, касающихся нашего оборудования, то можете воспользоваться нашей автоматизированной системой запросов технического сервис-центра helpdesk.qtech.ru.

Телефон Технической поддержки +7 (495) 797-33-11 доб. 0