Contents

Configuring GRE

About GRE

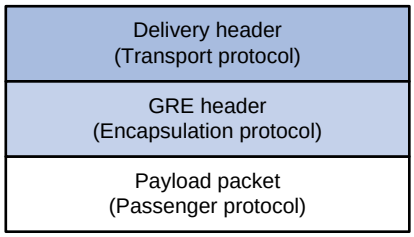
Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a protocol (such as IP, MPLS, or Ethernet) into a virtual point-to-point tunnel over a network (such as an IP network). Packets are encapsulated at one tunnel end and de-encapsulated at the other tunnel end. The network layer protocol of the packets before encapsulation and after encapsulation can be the same or different.

GRE encapsulation format

As shown in Figure 1, a GRE-tunneled packet includes the following parts:

- **Payload packet**—Original packet. The protocol type of the payload packet is called the passenger protocol. The passenger protocol can be any network layer protocol.
- **GRE header**—Header that is added to the payload packet to change the payload packet to a GRE packet. A GRE header includes the number of encapsulations, version, passenger protocol type, checksum, and key. GRE is called the encapsulation protocol.
- **Delivery header**—Header that is added to the GRE packet to deliver it to the tunnel end. The transport protocol (or delivery protocol) is the network layer protocol that transfers GRE packets.

Figure 1 GRE encapsulation format



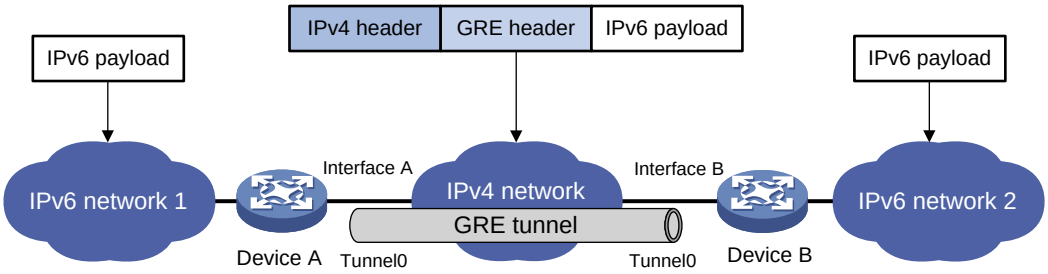
GRE tunnel operating principle

As shown in Figure 2, an IPv6 protocol packet traverses an IPv4 network through a GRE tunnel as follows:

1. After receiving an IPv6 packet from the interface connected to IPv6 network 1, Device A processes the packet as follows:
   a. Looks up the routing table to identify the outgoing interface for the IPv6 packet.
   b. Submits the IPv6 packet to the outgoing interface—the GRE tunnel interface Tunnel 0.
2. Upon receiving the packet, the tunnel interface encapsulates the packet with GRE and then with IPv4. In the IPv4 header:
   - The source address is the tunnel's source address (the IP address of Interface A of Device A).
   - The destination address is the tunnel's destination address (the IP address of Interface B of Device B).
3. Device A looks up the routing table according to the destination address in the IPv4 header, and forwards the IPv4 packet out of the physical interface (Interface A) of the GRE tunnel.

4. When the IPv4 arrives at the GRE tunnel destination Device B, Device B checks the destination address. Because the destination is Device B itself and the protocol number in the IP header is 47 (the protocol number for GRE), Device B submits the packet to GRE for de-encapsulation.
5. GRE first removes the IPv4 header, and then checks the packet sequence number. After GRE finishes the checking, it removes the GRE header, and submits the payload to the IPv6 protocol for forwarding.

Figure 2 IPv6 networks interconnected through a GRE tunnel



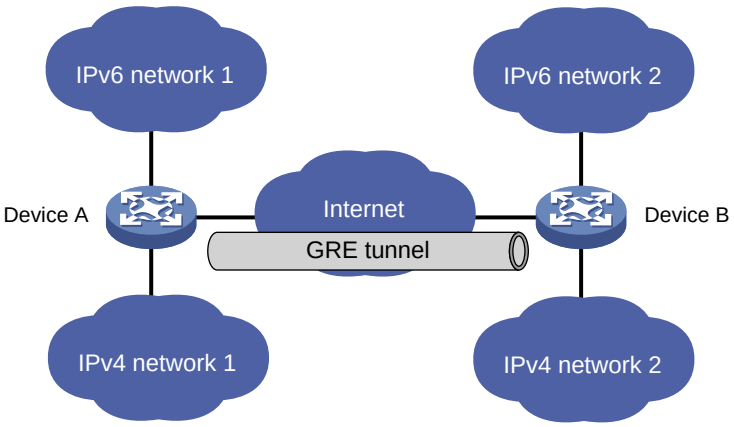
GRE application scenarios

The following shows typical GRE application scenarios:

Connecting networks running different protocols over a single backbone

As shown in Figure 3, IPv6 network 1 and IPv6 network 2 are IPv6 networks, and IPv4 network 1 and IPv4 network 2 are IPv4 networks. Through the GRE tunnel between Device A and Device B, IPv6 network 1 can communicate with IPv6 network 2 and IPv4 network 1 can communicate with IPv4 network 2, without affecting each other.

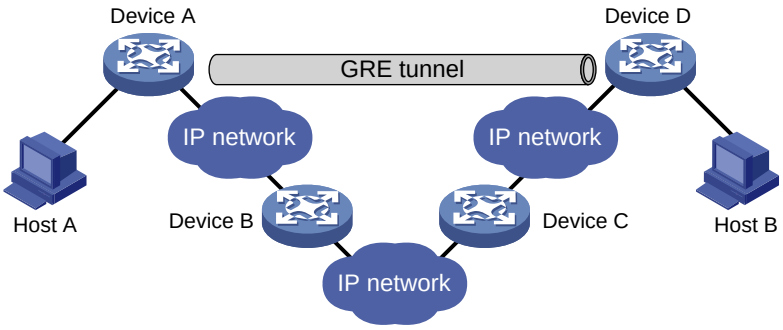
Figure 3 Network diagram



Enlarging network scope

In an IP network, the maximum TTL value of a packet is 255. If two devices have more than 255 hops in between, they cannot communicate with each other. By using a GRE tunnel, you can hide some hops to enlarge the network scope. As shown in Figure 4, only the tunnel-end devices (Device A and Device D) of the GRE tunnel are counted in hop count calculation. Therefore, there are only three hops between Host A and Host B.

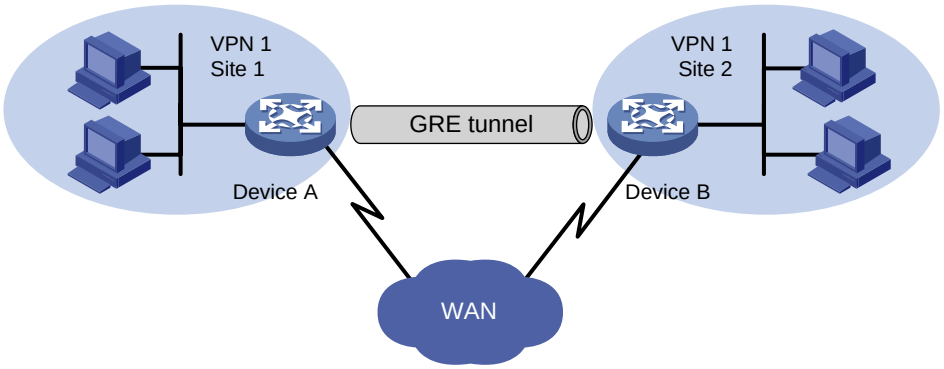
Figure 4 Network diagram



Constructing VPN

As shown in Figure 5, Site 1 and Site 2 both belong to VPN 1 and are located in different cities. Using a GRE tunnel can connect the two VPN sites across the WAN.

Figure 5 Network diagram



Protocols and standards

- RFC 1701, *Generic Routing Encapsulation (GRE)*
- RFC 1702, *Generic Routing Encapsulation over IPv4 networks*
- RFC 2784, *Generic Routing Encapsulation (GRE)*
- RFC 2890, *Key and Sequence Number Extensions to GRE*

Restrictions and guidelines: GRE configuration

When you configure a GRE tunnel, follow these restrictions and guidelines:

- You must configure the tunnel source address and destination address at both ends of a tunnel. The tunnel source or destination address at one end must be the tunnel destination or source address at the other end.
- As a best practice, do not configure the same tunnel source and destination addresses for local tunnel interfaces that use the same tunnel mode.
- To ensure correct packet forwarding, identify whether the destination network of packets and the IP address of the local tunnel interface are on the same subnet. If they are not, configure a route reaching the destination network through the tunnel interface. You can configure the route by using one of the following methods:

- Configure a static route, using the local tunnel interface as the outgoing interface of the route.
- Enable a dynamic routing protocol on both the tunnel interface and the interface connecting the private network. This allows the dynamic routing protocol to establish a routing entry with the tunnel interface as the outgoing interface.

- GRE encapsulation and de-encapsulation can decrease the forwarding efficiency of tunnel-end devices.

Configuring a GRE/IPv4 tunnel

Restrictions and guidelines

This task describes only GRE/IPv4 tunnel required tunnel interface commands (the **interface tunnel**, **source**, and **destination** commands). For more tunnel interface commands, see "Configuring tunneling."

Procedure

1. Enter system view.

   system-view

2. Create a GRE tunnel interface, and specify the tunnel mode as GRE/IPv4.

   interface tunnel *number* **mode gre**

   You must configure the same tunnel mode on both ends of a tunnel. Otherwise, packet delivery might fail.

3. Configure an IP address for the tunnel interface based on the passenger protocol.

   IPv4:

   For information about how to assign an IPv4 address to an interface, see "Configuring IP addressing."

   IPv6:

   For information about how to assign an IPv6 address to an interface, see "Configuring basic IPv6 settings."

   By default, no IP address is configured for a tunnel interface.

4. Configure a source address or source interface for the tunnel interface.

   source { *ip-address* | *interface-type interface-number* }

   By default, no source address or interface is configured for a tunnel interface.

   If you configure a source address for a tunnel interface, the tunnel interface uses the source address as the source address of the encapsulated packets.

   If you configure a source interface for a tunnel interface, the tunnel interface uses the primary IP address of the source interface as the source address of the encapsulated packets.

5. Configure a destination address for the tunnel interface.

   destination *ip-address*

   By default, no destination address is configured for a tunnel interface.

   The destination address is the address of the physical interface that the tunnel remote end uses to receive packets from the GRE tunnel.

   The tunnel local end uses this address as the destination address of the encapsulated packets.

   The tunnel destination address and the IP address of the tunnel interface must be in different subnets.

6. (Optional.) Enable GRE keepalive, and set the keepalive interval and keepalive number.

   keepalive [*interval* [*times*]]

By default, GRE keepalive is disabled.

7. (Optional.) Set the DF bit for encapsulated packets.

   tunnel dfbit enable

   By default, the DF bit is not set, allowing encapsulated packets to be fragmented.

Enabling dropping IPv6 packets that use IPv4-compatible IPv6 addresses

About dropping IPv6 packets that use IPv4-compatible IPv6 addresses

This feature enables the device to check the source and destination IPv6 addresses of the de-encapsulated IPv6 packets from a tunnel. If a packet uses an IPv4-compatible IPv6 address as the source or destination address, the device discards the packet.

Procedure

1. Enter system view.

   system-view

2. Configure the device to discard IPv6 packets with IPv4-compatible IPv6 addresses.

   tunnel discard ipv4-compatible-packet

   By default, the device does not discard such IPv6 packets.

   For more information about this command, see *Layer 3—IP Services Command Reference*.

Display and maintenance commands for GRE

Execute **display** commands in any view and **reset** commands in user view.

Task	Command	Remarks
Display information about tunnel interfaces.	**display interface** [**tunnel** [*number*]] [**brief** [**description** | **down**]]	For more information about the commands, see *Layer 3—IP Services Command Reference*.
Display IPv6 information about tunnel interface.	**display ipv6 interface** [**tunnel** [*number*]] [**brief**]	For more information about this command, see *Layer 3—IP Services Command Reference*.
Clear tunnel interface statistics.	**reset counters interface** [**tunnel** [*number*]]	For more information about this command, see *Layer 3—IP Services Command Reference*.
Clear IPv6 statistics on tunnel interfaces.	**reset ipv6 statistics** [**slot** *slot-number*]	For more information about this command, see IPv6 basics in *Layer 3—IP Services Command Reference*.

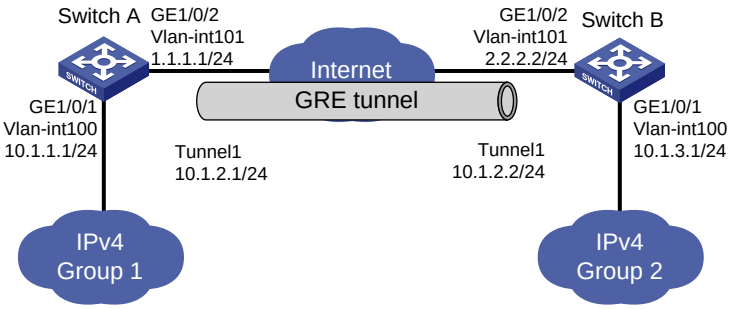
GRE configuration examples

Example: Configuring an IPv4 over IPv4 GRE tunnel

Network configuration

As shown in Figure 6, Group 1 and Group 2 are two private IPv4 networks. The two networks both use private network addresses and belong to the same VPN. Establish a GRE tunnel between Switch A and Switch B to interconnect the two private IPv4 networks Group 1 and Group 2.

Figure 6 Network diagram



Procedure

Before performing the following configuration, configure an IP address for each interface, and make sure Switch A and Switch B can reach each other.

1. Configure Switch A:

Enter system view.

```
<SwitchA> system-view
```

Create a tunnel interface Tunnel 1, and specify the tunnel mode as GRE/IPv4.

```
[SwitchA] interface tunnel 1 mode gre
```

Configure an IP address for the tunnel interface.

```
[SwitchA-Tunnel1] ip address 10.1.2.1 255.255.255.0
```

Configure the source address of tunnel interface as the IP address of VLAN-interface 101 on Switch A.

```
[SwitchA-Tunnel1] source 1.1.1.1
```

Configure the destination address of the tunnel interface as the IP address of VLAN-interface 101 on Switch B.

```
[SwitchA-Tunnel1] destination 2.2.2.2
[SwitchA-Tunnel1] quit
```

Configure a static route from Switch A through the tunnel interface to Group 2.

```
[SwitchA] ip route-static 10.1.3.0 255.255.255.0 tunnel 1
```

2. Configure Switch B:

Enter system view.

```
<SwitchB> system-view
```

Create a tunnel interface Tunnel 1, and specify the tunnel mode as GRE/IPv4.

```
[SwitchB] interface tunnel 1 mode gre
```

Configure an IP address for the tunnel interface.

```
[SwitchB-Tunnel1] ip address 10.1.2.2 255.255.255.0
```

Configure the source address of tunnel interface as the IP address of VLAN-interface 101 on Switch B.

```
[SwitchB-Tunnel1] source 2.2.2.2
```

Configure the destination address of the tunnel interface as the IP address of VLAN-interface 101 on Switch A.

```
[SwitchB-Tunnel1] destination 1.1.1.1
[SwitchB-Tunnel1] quit
```

Configure a static route from Switch B through the tunnel interface to Group 1.

```
[SwitchB] ip route-static 10.1.1.0 255.255.255.0 Tunnel 1
```

Verifying the configuration

Display tunnel interface information on Switch A.

```
[SwitchA] display interface tunnel 1
Tunnel1
Current state: UP
Line protocol state: UP
Description: Tunnel1 Interface
Bandwidth: 64kbps
Maximum transmission unit: 1476
Internet address: 10.1.2.1/24 (primary)
Tunnel source 1.1.1.1, destination 2.2.2.2
Tunnel keepalive disabled
Tunnel TTL 255
Tunnel protocol/transport GRE/IP
    GRE key disabled
    Checksumming of GRE packets disabled
Last clearing of counters: Never
Last 300 seconds input rate: 0 bytes/sec, 0 bits/sec, 0 packets/sec
Last 300 seconds output rate: 0 bytes/sec, 0 bits/sec, 0 packets/sec
Input: 0 packets, 0 bytes, 0 drops
Output: 0 packets, 0 bytes, 0 drops
```

Display tunnel interface information on Switch B.

```
[SwitchB] display interface tunnel 1
Tunnel1
Current state: UP
Line protocol state: UP
Description: Tunnel1 Interface
Bandwidth: 64kbps
Maximum transmission unit: 1476
Internet address: 10.1.2.2/24 (primary)
Tunnel source 2.2.2.2, destination 1.1.1.1
Tunnel keepalive disabled
Tunnel TTL 255
Tunnel protocol/transport GRE/IP
    GRE key disabled
    Checksumming of GRE packets disabled
Last clearing of counters: Never
Last 300 seconds input rate: 0 bytes/sec, 0 bits/sec, 0 packets/sec
Last 300 seconds output rate: 0 bytes/sec, 0 bits/sec, 0 packets/sec
```

```
Input: 0 packets, 0 bytes, 0 drops
Output: 0 packets, 0 bytes, 0 drops
```

From Switch B, ping the IP address of VLAN-interface 100 on Switch A.

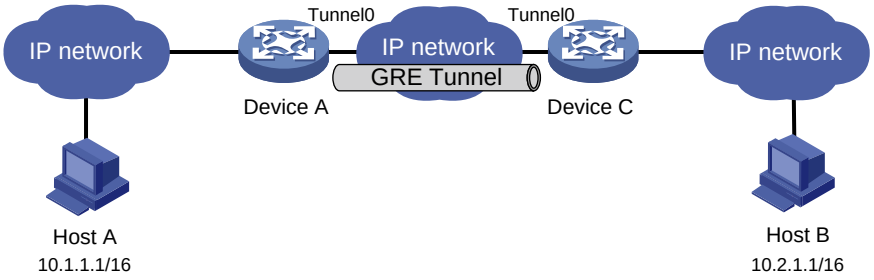
```
[SwitchB] ping -a 10.1.3.1 10.1.1.1
Ping 10.1.1.1 (10.1.1.1) from 10.1.3.1: 56 data bytes, press CTRL_C to break
56 bytes from 10.1.1.1: icmp_seq=0 ttl=255 time=11.000 ms
56 bytes from 10.1.1.1: icmp_seq=1 ttl=255 time=1.000 ms
56 bytes from 10.1.1.1: icmp_seq=2 ttl=255 time=0.000 ms
56 bytes from 10.1.1.1: icmp_seq=3 ttl=255 time=0.000 ms
56 bytes from 10.1.1.1: icmp_seq=4 ttl=255 time=0.000 ms

--- Ping statistics for 10.1.1.1 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.000/2.400/11.000/4.317 ms
```

The output shows that Switch B can successfully ping Switch A.

Troubleshooting GRE

The key to configuring GRE is to keep the configuration consistent. Most faults can be located by using the **debugging gre** or **debugging tunnel** command. This section analyzes one type of fault for illustration, with the scenario shown in Figure 7.

Figure 7 Network diagram



Hosts at both ends of a GRE tunnel cannot ping each other

Symptom

The interfaces at both ends of the tunnel are configured correctly and can ping each other, but Host A and Host B cannot ping each other.

Solution

To resolve the issue:

1. Execute the **display ip routing-table** command on Device A and Device C to view whether Device A has a route over tunnel 0 to 10.2.0.0/16 and whether Device C has a route over tunnel 0 to 10.1.0.0/16.
2. If such a route does not exist, execute the **ip route-static** command in system view to add the route. Take Device A as an example:

   ```
   [DeviceA] ip route-static 10.2.0.0 255.255.0.0 tunnel 0
   ```

3. If the issue persists, contact H3C Support.