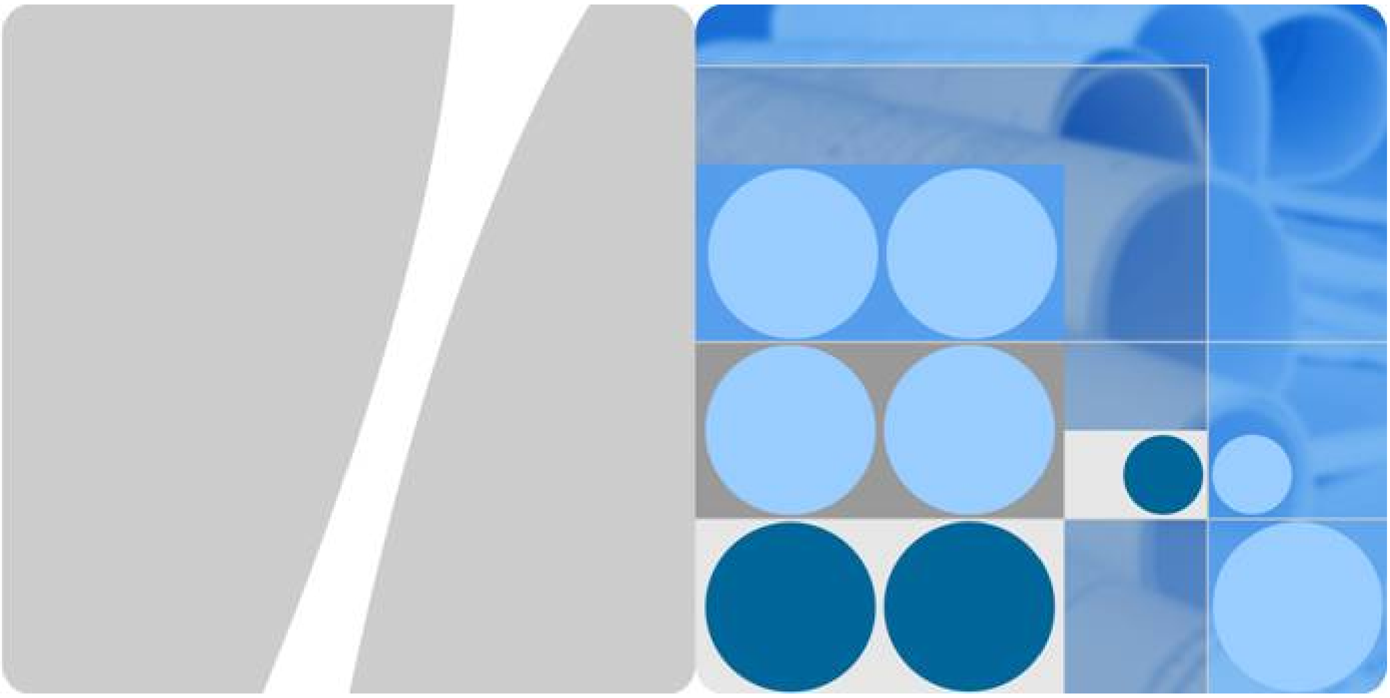


HUAWEI NetEngine5000E Core Router

V800R002C01

Configuration Guide - IP Routing

Issue	**01**
Date	**2011-10-15**

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Huawei Technologies Co., Ltd.

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

Website: http://www.huawei.com

Email: support@huawei.com

About This Document

Intended Audience

This document provides the basic concepts, configuration procedures, and configuration examples in different application scenarios of the IP Routing feature supported by the NE5000E device.

This document describes how to configure the Basic Configurations feature.

This document is intended for:

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

Related Versions (Optional)

The following table lists the product versions related to this document.

Product Name	Version
HUAWEI NetEngine5000E Core Router	V800R002C01

Symbol Conventions

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

Symbol	Description
⚠ DANGER	Indicates a hazard with a high level of risk, which if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazard with a medium or low level of risk, which if not avoided, could result in minor or moderate injury.

Symbol	Description
⚠ CAUTION	Indicates a potentially hazardous situation, which if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
TIP	Indicates a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points of the main text.

Command Conventions (Optional)

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

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

Change History

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

Changes in Issue 01 (2011-10-15)

The initial commercial release.

Contents

1 IP Routing Overview

About This Chapter

IP routing is the basic element of data communication networks.

1.1 Routing Management

To forward data, routers must be capable of establishing and refreshing routing tables and forwarding datagrams according to routing tables.

Routers provide a mechanism joining heterogeneous networks, transmitting datagrams from one network to another. A route contains the information about the path along which IP datagrams are forwarded.

1.1.1 Router

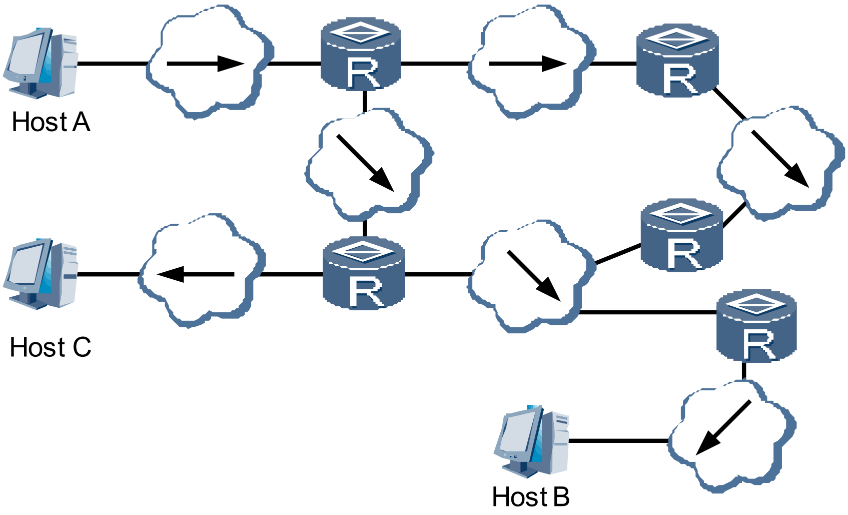
The major function of routers is to interconnect dissimilar kinds of networks so that data can be transmitted across the interconnected network.

On the Internet, network connection devices are used to control network traffic and ensure the quality of data transmission on the network. Common network connection devices include hubs, bridges, switches, and routers.

As a typical network connection device, a router performs route selection and packet forwarding. Upon receiving a packet, a router selects a preferred path, which has one or multiple hops, to send the packet to the next router according to the destination address in the packet. The last router is responsible for sending the packet to the destination host. The router is able to determine which path is the optimal one to transmit data.

For example, in **Figure 1-1**, traffic from Host A to Host C needs to pass through three networks and two routers. The number of hops from a router to a directly connected network is 0. The number of hops from a router to the network which the router can reach through another router is 1, and so on. If a router is connected to another router through a network, a network segment exists between the two routers. The two routers are considered adjacent to each other on the Internet. The bold arrows in the figure indicate network segments. The routers do not have to be concerned with which physical links compose each network segment.

Figure 1-1 Network segment and number of hops



The sizes of networks may vary greatly, and the actual lengths of network segments are also different. Therefore, for different networks, you can set a weighted coefficient for their network segments, and then measure the cost of a route according to the number of network segments.

A route with the minimum number of network segments is not necessarily the ideal route. For example, a route passing through three high-speed Local Area Network (LAN) network segments may be of a much higher rate than a route passing through two low-speed Wide Area Network (WAN) network segments.

1.1.2 Routing Table

The routing table is the key factor for a router to forward packets.

A router selects routes from the routing table, and each router maintains at least one routing table.

The routing table stores the routes discovered by various routing protocols. Based on the sources, routes in the routing table are classified into the following types:

- Routes discovered by link layer protocols, which are also called interface routes or direct routes
- Static routes that are manually configured by the network administrator
- Dynamic routes that are discovered by dynamic routing protocols

Routing Table

Each router maintains a protocol routing table for each protocol and a local core routing table (or routing management table).

- Protocol routing table

  A protocol routing table stores the routing information discovered by the protocol.

  A routing protocol can import and advertise the routes generated by other routing protocols. For example, if the router that runs the Open Shortest Path First (OSPF) protocol needs to use OSPF to advertise direct routes, static routes, or Intermediate System to Intermediate System (IS-IS) routes, the router needs to import these routes into the OSPF routing table.
- Local core routing table

  A router uses the local core routing table to store protocol routes and preferred routes. The router then delivers the preferred routes for packet forwarding. Routes in the routing table are selected according to the priorities of protocols and costs. You can run the **display ip routing-table** command to view the local core routing table of a router.

  NOTE

  The router that supports the Layer 3 Virtual Private Network (L3VPN) maintains a management routing table, namely, a local core routing table, for each VPN instance.

Contents in the Routing Table

In the NE5000E, by running the **display ip routing-table** command, you can view brief information about the routing table of a router.

```
<HUAWEI> display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table: _public_
         Destinations : 8        Routes : 8
```

```
Destination/Mask   Proto  Pre  Cost    Flags NextHop        Interface

   0.0.0.0/0       Static 60   0       D     1.1.4.2        Pos1/0/0
  1.1.1.0/24       Direct 0    0       D     1.1.1.1        GigabitEthernet2/0/0
  1.1.1.1/32       Direct 0    0       D     127.0.0.1      GigabitEthernet2/0/0
  1.1.4.0/30       OSPF   10   0       D     1.1.4.1        Pos1/0/0
  1.1.4.1/32       Direct 0    0       D     127.0.0.1      Pos1/0/0
  1.1.4.2/32       OSPF   10   0       D     1.1.4.2        Pos1/0/0
127.0.0.0/8        Direct 0    0       D     127.0.0.1      InLoopBack0
127.0.0.1/32       Direct 0    0       D     127.0.0.1      InLoopBack0
```

A routing table contains the following key entries:

- Destination: indicates the destination address. The destination address identifies the destination IP address or the destination network address of an IP packet.
- Mask: indicates the network mask. The network mask and the destination address are used together to identify the address of the network segment where the destination host or router resides.
  - The address of the network segment where the destination host or router resides can be calculated based on the combined destination address and network mask. For example, if the destination address is 1.1.1.1 and the mask is 255.255.255.0, the address of the network segment where the host or the router resides is 1.1.1.0.
  - The mask, which consists of several consecutive 1s, can be expressed either in dotted decimal notation or by the number of consecutive 1s in the mask. For example, the length of the mask 255.255.255.0 is 24, and therefore, it can also be expressed as 24.
- Pre: indicates the preference of a route that is added to the IP routing table. There may be multiple routes to the same destination, which have different next hops or outbound interfaces. These routes may be discovered by different routing protocols or manually configured. The route with the highest preference (the smallest value) is selected as the optimal route. For details about the route preference of each routing protocol, see **Table 1-1**.
- Cost: indicates the route cost. When multiple routes to the same destination have the same preference, the route with the smallest cost is selected as the optimal route.

  NOTE

  The preference is used to compare the preferences of the routes discovered by different routing protocols, whereas the cost is used to compare the preferences of the routes discovered by the same routing protocol.
- Nexthop: indicates the IP address of the next hop. The next hop refers to the next router that an IP packet passes through.
- Interface: indicates the outbound interface. It is the interface through which an IP packet is forwarded.

Based on the destination addresses, routes can be classified into the following types:

- Network segment route: The destination is a network segment.
- Host route: The destination is a host.

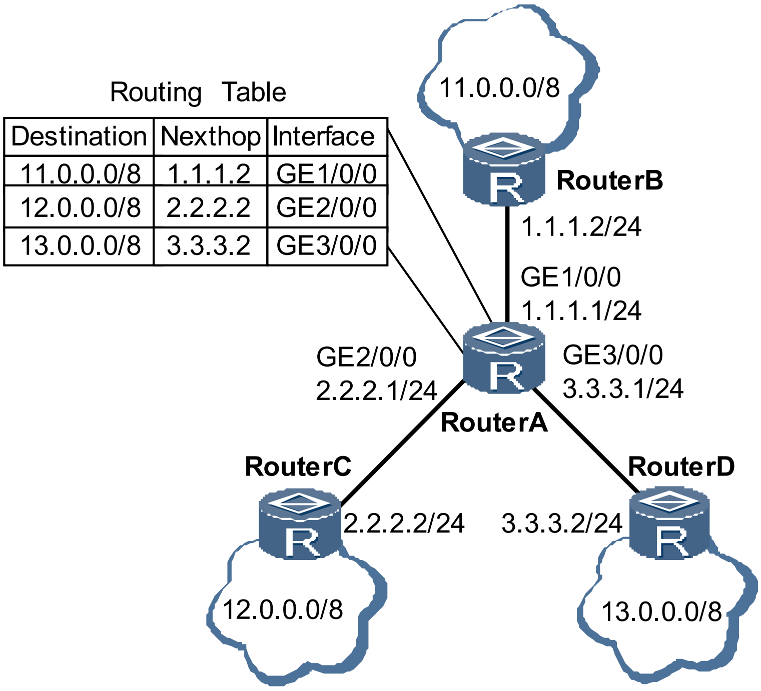
In addition, based on whether the destination is directly connected to the router, routes can be classified into the following types:

- Direct route: The router is directly connected to the network where the destination resides.
- Indirect route: The router is indirectly connected to the network where the destination resides.

You can set a default route to reduce the number of routing entries in the routing table. When a router fails to find a preferred route in the routing table, it sends packets through the default route. In the preceding routing table, the route whose destination address is 0.0.0.0/0 is a default route.

As shown in **Figure 1-2**, Router A is connected to three networks, and therefore it has three IP addresses and three outbound interfaces. **Figure 1-2** shows the routing table of Router A.

Figure 1-2 Schematic diagram of the routing table



Routing Table

Destination	Nexthop	Interface
11.0.0.0/8	1.1.1.2	GE1/0/0
12.0.0.0/8	2.2.2.2	GE2/0/0
13.0.0.0/8	3.3.3.2	GE3/0/0

1.2 IP Routing Features Supported by the NE5000E

The features that are involved in the configuration of IP routes include route classification, route preference, load balancing, route backup, routing policy, Bidirectional Forwarding Detection (BFD), IPv4 Fast Reroute (FRR), IPv6 Fast Reroute, IPv4 Multi-topology, IPv6 Multi-topology, ISSU, NSR, GR and Techniques for Routing Security.

1.2.1 Route Classification

There are three major types of route: direct route, static route, and dynamic route.

Besides direct routes and static routes, the NE5000E supports routes of dynamic routing protocols, such as RIP, OSPF, IS-IS, and the Border Gateway Protocol (BGP).

Static routes can be easily configured and have low requirements on the system. They are applicable to simple, stable, and small-scale networks. The disadvantage of static routes is that they cannot automatically adapt to changes in the network topology. Therefore, static routes require subsequent maintenance.

Routers rely on routing protocols to select routes and forward packets. As the rules used by routers to maintain routing tables, routing protocols are used to discover routes, generate routing tables, and guide packet forwarding. Routing protocols are classified into link-state protocols and distance-vector protocols.

Dynamic routing protocols have their routing algorithms and can automatically adapt to changes of the network topology. They are applicable to the network equipped with certain Layer 3 devices. The configurations of dynamic routes are complex. Dynamic routes have higher requirements on the system than static ones and consume network resources.

Dynamic routing protocols can be classified according to any of the following conditions.

Application Scope

According to the application scope, routing protocols can be classified into the following types:

- Interior Gateway Protocol (IGP): runs inside an Autonomous System (AS), including RIP, OSPF, and IS-IS.
- Exterior Gateway Protocol (EGP): runs between different ASs. At present, BGP is the most widely used EGP.

Routing Algorithm

According to the routing algorithm, routing protocols can be classified into the following types:

- Distance-vector routing protocol: includes RIP and BGP. BGP is also called the path-vector protocol.
- Link-state routing protocol: includes OSPF and IS-IS.

The preceding routing algorithms mainly differ in the methods of discovering routes and calculating routes.

Type of the Destination Address

According to the type of the destination address, routing protocols can be classified into the following types:

- Unicast routing protocols: include RIP, OSPF, BGP, and IS-IS.
- Multicast routing protocols: include Protocol Independent Multicast-Sparse Mode (PIM-SM) and Protocol Independent Multicast-Dense Mode (PIM-DM).

NOTE

This manual mainly describes unicast routing protocols. For details on multicast routing protocols, refer to the *HUAWEI NetEngine5000E Core Router Feature Description -IP Multicast*.

Static routes and dynamic routes discovered by routing protocols are centrally managed on the router. Routing protocols can import routes from each other to implement routing information advertisement.

1.2.2 Route Preference

If a routing table has multiple routes, the route with the highest preference is the optimal route. By adjusting the route convergence priority, you can ensure faster convergence of routes for key services.

Route Preference

Different routing protocols, including the static routing protocol may discover different routes to the same destination, but not all these routes are optimal. At a certain moment, only one routing protocol determines the preferred route to a certain destination. To select the optimal route, routes of routing protocols including static routes are configured with preferences. When there are multiple routing information sources, the route learned by the routing protocol with the highest preference becomes the optimal route. The smaller the value, the higher the preference. **Table 1-1** lists routing protocols and the default preferences of the routes learned by these protocols.

In **Table 1-1**, 0 indicates the direct route and 255 indicates any route learnt from unreliable sources. The smaller the value, the higher the preference.

Table 1-1 Routing protocol and default route preference

Routing Protocol or Route Type	Route Preference
DIRECT	0
OSPF	10
IS-IS	15
STATIC	60
RIP	100
OSPF ASE	150
OSPF NSSA	150
IBGP	255
EBGP	255

Except for direct routes, the preferences of routing protocols can be manually configured. In addition, the preference of each static route can be different.

The NE5000E defines the external preference and internal preference. The external preference refers to the preference set by users for each routing protocol. **Table 1-1** shows the default external preferences.

When different routing protocols are configured with the same preference, the system determines which route discovered by these routing protocols becomes the optimal route according to the internal preference. For the internal preference of each routing protocol, see **Table 1-2**.

Table 1-2 Internal preference of each routing protocol

Routing Protocol or Route Type	Route Preference
DIRECT	0
OSPF	10
IS-IS Level-1	15

Routing Protocol or Route Type	Route Preference
IS-IS Level-2	18
STATIC	60
RIP	100
OSPF ASE	150
OSPF NSSA	150
IBGP	200
EBGP	20

Presume that there are two routes: an OSPF route and a static route. Both routes can reach the destination 10.1.1.0/24, and the preferences of the two routes are set to 5. In this case, the NE5000E determines the optimal route according to the internal preferences listed in **Table 1-2**. The internal preference (the value is 10) of the OSPF route is higher than the internal preference (the value is 60) of the static route. Therefore, the system selects the route discovered by OSPF as the optimal route.

Route Convergence Priority

As an important technology for improving network reliability, priority-based route convergence provides faster convergence of routes for key services. For example, to shorten the interruption period of key services in case of network faults, Video on Demand (VoD) services require routes to multicast sources to be fast converged, whereas the Multiprotocol Label Switching (MPLS) VPN bearer network requires routes between PEs to be fast converged.

Priorities in route convergence are critical, high, medium, and low, which are listed in descending order. **Table 1-3** lists the default convergence priorities of public network routes. You can set convergence priorities for OSPF routes as required.

Table 1-3 Default convergence priorities of public network routes

Routing Protocol or Route Type	Convergence Priority
DIRECT	Critical
STATIC	medium
32-bit host routes (routes with 32-bit masks) of OSPF and IS-IS	medium
OSPF routes (excluding 32-bit host routes)	low
IS-IS routes (excluding 32-bit host routes)	low

Routing Protocol or Route Type	Convergence Priority
RIP	low
BGP	low

NOTE

For private network routes, direct routes are identified as critical, 32-bit host routes of OSPF and IS-IS are identified as medium, and the other routes are identifies as low.

1.2.3 Load Balancing and Route Backup

Configuring load balancing and route backup helps allocate network resources more reasonably, improving the reliability of the network.

Load Balancing

The NE5000E supports the multi-route model. That is, you can configure multiple routes that have the same destination and same preference. If the destinations and costs of the multiple routes discovered by one routing protocol are the same, load balancing can be performed among these routes. In the view of each routing protocol, you can run the **maximum load-balancing** *number* command to implement load balancing. Load balancing can be classified into the following types:

- Packet-based load balancing

  When packet-based load balancing is configured, a router at the network layer forwards packets to the same destination through each equal-cost route. That is, the router always chooses the next-hop address that is different from the last one to send packets. In this manner, packet-based load balancing is implemented.

- Flow-based load balancing

  When flow-based load balancing is configured, the router forwards packets according to five factors, namely, the source addresses, destination addresses, source ports, destination ports, and protocols in the packets. When the five factors are the same, the router always chooses the next-hop address that is the same as the last one to send packets.

In the current implementation, RIP, OSPF, BGP, and IS-IS support load balancing, and static routes also support load balancing.

NOTE

The number of equal-cost routes for load balancing varies with products.

Route Backup

The NE5000E supports route backup to improve network reliability. You can configure multiple routes to the same destination as required. The route with the highest preference functions as the primary route, and the other routes with lower preferences function as backup routes.

Normally, the router uses the primary route to forward packets. When the link fails, the primary route becomes inactive. The router then selects a backup route with the highest preference to forward packets. In this manner, the primary route is switched to the backup route. When the previous primary route recovers, the router restores the corresponding route and reselects the optimal route. Because the previous primary route has the highest preference, the router selects this route to send packets. In this manner, the backup route is switched to the primary route.

1.2.4 Routing Policy

By using routing policies, you can control the selection, sending, and receiving of routes.

Routing policies are applied to routing information to change the path through which network traffic passes. Routing policies can change the path through which network traffic passes by applying route attributes (including reachability).

When advertising or receiving routes, a router uses certain routing policies to filter routes by applying route attributes. Currently, the main method of implementing routing policies is to filter routing information.

NOTE

For details about how to configure routing policies, refer to the chapter "Routing Policy Configuration".

1.2.5 BFD Link Detection

BFD sessions provide a fast link detection mechanism for IP routes.

Bidirectional Forwarding Detection (BFD) is a unified detection mechanism used to rapidly detect and track the connectivity of the network links or IP routes. For better performance, two adjacent systems must be capable of fast detecting communications faults to switch traffic to a normal tunnel for service recovery.

NOTE

For details on how to configure BFD, refer to the chapter "BFD Configuration" in *HUAWEI NetEngine5000E Core Router Configuration Guide - Reliability*.

BFD for Static Routes

Static routes do not have a detection mechanism. When a fault occurs on a network, administrator interference is required.

With the feature of BFD for static routes, the BFD session can be used to detect the status of the static IPv4 routes of the public network. The routing management (RM) module determines whether static routes are available according to the BFD session status.

NOTE

For details on how to configure BFD for static routes, refer to the chapter "IP Static Route Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Routing*.

BFD for Dynamic Routing Protocols

BFD uses the local discriminator and remote discriminator to differentiate multiple BFD sessions between a pair of systems. IS-IS can both dynamically and statically create BFD sessions; BGP and OSPF can dynamically create BFD sessions.

The BFD session dynamically triggered by a routing protocol is implemented as follows:

- Dynamically allocating the local discriminator
- Self-learning the remote discriminator

When the neighbor relationship is set up between devices running the same routing protocol, the routing protocol notifies BFD through the RM module to establish BFD sessions. The neighbor relationship then can be rapidly detected. The detection parameters of BFD sessions are negotiated by both ends through the routing protocol.

When a BFD session detects a fault, the BFD session becomes Down, triggering route convergence.

NOTE

Usually, a routing protocol is only capable of second-level detection by using the Keepalive mechanism that is based on Hello messages. BFD, however, implements millisecond-level detection. It uses a 10ms detection period and a detection interval that is three times the detection period. Therefore, BFD can report a protocol fault within 50ms, improving the route convergence speed to a large extent.

When the neighbor becomes unreachable, the routing protocol instructs BFD to delete the session.

NOTE

For details on how to configure BFD for dynamic routing protocols, refer to the related configurations of dynamic routing protocols in the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Routing*.

1.2.6 IP FRR

IP FRR is applicable to IP services that are sensitive to delay and packet loss.

IP FRR Overview

IP FRR is applicable to the services that are very sensitive to packet loss and delay. After FRR is configured, when a fault is detected at the lower layer (physical layer or link layer), the fault is reported to the upper layer routing system. Meanwhile, packets are forwarded through a backup link. In this manner, the impact of link faults on services is minimized.

The IP FRR function of the NE5000E can back up direct routes, static routes, and dynamic RIP routes, OSPF routes, IS-IS routes, and BGP routes.

Background of IP FRR

In traditional IP networks, after a forwarding device such as a router detects a fault on the link of the lower layer, it takes the system several seconds to complete the route convergence. A second-level convergence may seriously affect services that are extremely sensitive to delay or packet loss, leading to service interruption. In the case of the VoIP service, the maximum convergence time tolerable is about 50ms when network interruption occurs.

Therefore, to prevent services from being seriously affected by faults on the link, the forwarding system must be able to fast detect and rectify faults, and to restore the affected services as soon as possible. IP FRR ensures that the forwarding system rapidly detects such a fault and then takes measures to restore services as soon as possible.

Classification and Implementation of IP FRR

Designed for routes on IP networks, IP FRR are classified into public network IP FRR and VPN IP FRR.

- Public network IP FRR: protects routers on the public network.

- VPN IP FRR: protects Customer Edges (CEs).

NOTE

For detailed configuration of VPN IP FRR, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN*.

The major means of implementing IP FRR are described as follows:

- IP FRR is enabled or disabled through commands.
- When optimal routes are selected from the routes discovered by routing protocols, a backup link is selected for each preferred primary link according to the protocol priority, and then the forwarding information of primary and backup links is provided for the forwarding engine.

1.2.7 ISSU

When a single device is being upgraded, ISSU ensures non-stop traffic forwarding.

In-Service Software Upgrade (ISSU) is a mechanism that ensures non-stop traffic forwarding when the system software of the forwarding device is being upgraded or rolled back.

ISSU can shorten the service interruption period during the planned software upgrade, greatly improving the reliability of devices. ISSU also minimizes the impact of upgrade failure on the system by means of the rollback mechanism.

ISSU applies to direct routes, static routes, IS-IS routes, OSPF routes, and BGP routes.

NOTE

For details on how to configure ISSU, refer to the chapter "ISSU Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - Reliability*.

1.2.8 NSR

Non-Stop Routing (NSR) ensures that services are not affected when a fault occurs on the forwarding device, or minimizes the impact on ongoing services.

As networks develop at a fast pace, operators are having increasingly higher requirements for reliability of IP networks. NSR, as a high availability (HA) solution, is introduced to ensure that services transmitted by a device are not affected when a hardware or software failure occurs on the device.

NSR ensures that the control plane of a neighbor does not sense the fault on the control plane of a router that has a backup control plane. In this process, the neighbor relationships set up by using specific routing protocols, MPLS, and other protocols that carry services are not interrupted.

As an HA solution, NSR ensures that user services are not affected or least affected in case of device failures.

Currently, direct routes, static routes, RIP, OSPF, IS-IS and BGP running on the HUAWEI NetEngine5000E implement NSR.

NOTE

For details on how to configure NSR, refer to the chapter "NSR Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - Reliability*.

1.2.9 GR

GR ensures non-stop forwarding when the routing protocol is restarted in case of master-slave switchover of the system.

Graceful Restart (GR) is a mechanism to prevent services from being affected in case of protocol reconvergence on a router or master-slave switchover of the router. With GR, the neighbor does not immediately delete the routes learnt from the router. Instead, the neighbor waits for the re-establishment of the neighbor relationship. Then, the two exchange routing information again. According to the GR specifications, the router that is restarted must adopt the architecture with the control plane and forwarding plane being independent of each other. This means that restart of the control plane does not affect packet forwarding.

IETF extends the protocols related to IP/MPLS forwarding, such as OSPF, IS-IS, BGP, LDP, and Resource Reservation Protocol (RSVP) to ensure that the forwarding is not interrupted by protocol restart. This reduces protocol flapping at the control plane when the system performs the master-slave switchover. These IETF-proposed standards are generally referred to as these protocols' Graceful Restart extension, or GR in short. Currently, GR has been widely applied to the master-slave switchover and system upgrade.

Currently, OSPF, IS-IS, and BGP running on the HUAWEI NetEngine5000E implement GR.

NOTE

- For details on how to configure GR, refer to the chapter "GR Configuration" in *HUAWEI NetEngine5000E Core Router Configuration Guide - Reliability*.
- For details on how to configure OSPF GR, IS-IS GR, and BGP GR, refer to the chapters "OSPF Configuration", "IS-IS Configuration", and "BGP Configuration" respectively.

1.2.10 Techniques for Routing Security

Proper use of routing security techniques improves the security and reliability of networks.

Currently, the HUAWEI NetEngine5000E implements the following techniques for routing security.

- Restricting the number of routes

  This technique restricts the number of routes in the routing table. After the number of routes in the routing table exceeds the configured upper limit, routes dynamically learnt through routing protocols can no longer be added to the routing table. Static routes or direct routes, however, can still be added to the routing table.

  NOTE

  When the number of routes in the routing table exceeds the configured upper limit, a log is generated. For details on the log, refer to the *HUAWEI NetEngine5000E Core Router Log Reference*.

- GTSM

  GTSM is short for Generalized TTL Security Mechanism (GTSM), a mechanism that protects the service over IP layer by checking whether the TTL value in the IP packet header is within a pre-defined range.

  The main purpose of GTSM is to protect TCP/IP-based control plane from Denial of Service (DoS) attacks. For example, an attacker may keep sending packets to a router by simulating a routing protocol. Such attack packets are received and processed by the router as valid packets, and as a result, the system becomes extremely busy, and the CPU usage remains at an exceedingly high level. To avoid CPU overload, GTSM can be deployed to protect IP-forwarded services by checking whether the TTL value in the IP header of the IP packet is within a pre-defined range.

NOTE

For details on how to configure OSPF GTSM and BGP GTSM, refer to the chapters "OSPF Configuration" and "BGP Configuration" respectively.

- MD5 authentication

  The NE5000E supports the MD5 authentication mode:

  - MD5 authentication: You only need to set MD5 authentication passwords for TCP connections. The authentication is performed by TCP. If the authentication fails, the TCP connection cannot be established.

1.3 Configuring the Router ID

By configuring the router ID, you can uniquely identify a device in an AS.

Applicable Environment

A router ID is a 32-bit IP address that uniquely identifies a router in an Autonomous System (AS). A router ID can be generated as follows:

- Manually configured on the public network or private network
- Protocol-configured
- Automatically selected

NOTE

In a network, if the ID of a router is the IP address of the physical interface of the router, when the IP address is changed, route flapping may occur. To enhance the stability of the network, it is recommended that you configure the IP address of the loopback interface as the router ID.

Pre-configuration Tasks

None

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
router id router-id
```

The Router ID is specified.

By default, if a router is not configured with any interface, the router ID is 0.0.0.0.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Using the **display router id** command, you can view the set router ID.

# Run the **display router id** command to view the ID of the router.

```
<HUAWEI> display router id
RouterID:1.1.1.1
```

1.4 Configuring IPv4 FRR

IPv4 FRR is suitable for IPv4 services that are very sensitive to packet loss and delay.

Applicable Environment

Public network IPv4 FRR is applicable to the services that are sensitive to packet loss or delay on the public network.

After FRR is configured, when a fault is detected at the lower layer, the fault is reported to the upper-layer routing system. Then, packets are forwarded through a backup link. In this manner, the impact of link faults on the ongoing services is minimized.

⚠ **CAUTION**

IPv4 FRR enables different protocol routes to serve as a backup to each other, and therefore may result in loops. You should configure IPv4 FRR with caution.

Pre-configuration Task

Before configuring IPv4 FRR, complete the following task:

- Configuring parameters of the link layer protocol and IPv4 addresses for interfaces and ensuring that the status of the link layer protocol on the interfaces is Up
- Configuring routes destined for the same destination address but discovered by different routing protocols.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip frr
```

IPv4 FRR is enabled.

NOTE

If FRR is configured both in the system view or for routing protocols, the FRR that is configured for routing protocols preferentially takes effect.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Using the **display ip routing-table verbose** command, you can view information about a backup outbound interface and the backup next hop in the routing table.
- Using the **display ip routing-table** *ip-address* [*mask* | *mask-length*] [**longer-match**] **verbose** command, you can view information about a backup outbound interface and the backup next hop in the routing table.
- Using the **display ip routing-table** *ip-address1* { *mask1* | *mask-length1* } *ip-address2* { *mask2* | *mask-length2* } **verbose** command, you can view information about a backup outbound interface and the backup next hop in the routing table.

# Run the **display ip routing-table verbose** command to check information about route backup.

```
<HUAWEI> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 172.17.1.0/24
     Protocol: OSPF            Process ID: 1
   Preference: 10                    Cost: 3
      NextHop: 192.168.10.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h06m49s
          Tag: 0                 Priority: low
        Label: NULL               QoSInfo: 0x0
   IndirectID: 0x0
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet2/0/0
     TunnelID: 0x0                  Flags:  D
    BkNextHop: 192.168.20.2   BkInterface: GigabitEthernet3/0/0
      BkLabel: NULL           SecTunnelID: 0x0
 BkPETunnelID: 0x0        BkPESecTunnelID: 0x0
 BkIndirectID: 0x0
```

Related Tasks

1.7.1 Example for Configuring IPv4 FRR of the Public Network

1.5 Configuring IPv6 FRR

IPv6 FRR is applicable to services that are sensitive to the delay and packet loss on an IPv6 network.

Applicable Environment

IPv6 FRR of the public network is applicable to services that are sensitive to the delay and packet loss on an IPv6 public network.

After IPv6 FRR is configured, if a link fault is detected at a lower layer (physical layer or link layer), the fault is reported to the upper layer routing system. Meanwhile, packets are forwarded by using a backup link. The impact of the link fault on services is thus minimized.

CAUTION

IPv6 FRR enables routes generated by different routing protocols to back up each other, which may cause a loop. Therefore, exercise caution when configuring IPv6 FRR.

Pre-configuration Tasks

Before configuring IPv6 FRR, complete the following tasks:

- Configuring link layer protocol parameters and assigning IPv6 addresses to interfaces to ensure that the status of the link layer protocol on the interfaces is Up
- Configuring routes destined for the same destination address but discovered by different routing protocols.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 frr
```

IPv6 FRR is enabled.

NOTE

When IPv6 FRR is configured in both the system view and the routing protocol view, the IPv6 FRR configuration in the routing protocol view takes precedence over that in the system view.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display ipv6 routing-table verbose** command to check detailed information about backup outbound interfaces and backup next hops of routes in the routing table.

- Run the **display ipv6 routing-table** *ip-address* [*mask* | *mask-length*] [**longer-match**] **verbose** command to check information about the backup outbound interfaces and backup next hops of routes in the routing table.
- Run the **display ipv6 routing-table** *ip-address1* { *mask1* | *mask-length1* } *ip-address2* { *mask2* | *mask-length2* } **verbose** command to check information about the backup outbound interfaces and backup next hops of routes in the routing table.

# Run the **display ipv6 routing-table verbose** command, and you can view information about route backup. For example:

```
<HUAWEI> display ipv6 routing-table 20::1 64 verbose
Routing Table : _public_
Summary Count : 2

Destination  : 20::                                    PrefixLength : 64
NextHop      : 200::2                                  Preference   : 10
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol    : OSPFv3
State        : Active Adv                              Cost         : 3
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0x69                                    Age          : 553sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 2/0/0                   Flags        : D
BkNextHop    : 100::2                                BkInterface : gigabitethernet
3/0/0
BkLabel      : NULL                                    BkTunnelID   : 0x0
BkPETunnelID : 0x0                                     BkIndirectID : 0xb5

Destination  : 20::                                    PrefixLength : 64
NextHop      : 100::2                                  Preference   : 15
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol     : ISIS
State        : Inactive Adv                            Cost         : 30
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0xb5                                    Age          : 485sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 3/0/0                  Flags        : 0
```

Related Tasks

1.7.2 Example for Configuring IPv6 FRR for a Public Network

1.6 Maintaining IP Routes

Maintaining IP routes involves displaying the routing table and routing management module, and debugging the routing management module.

1.6.1 Displaying the Routing Table

Viewing information about the routing table helps locate faults on the network.

Context

It is necessary to check the information in the routing table in order to locate the routing problems. The following shows the common commands for displaying routing information.

The **display** commands can be used in all views.

Procedure

- Using the **display ip routing-table** command, you can view brief information about activated routes in the IP routing table.
- Using the **display ip routing-table verbose** command, you can view detailed information about the IP routing table.
- Using the **display ip routing-table** *ip-address* [*mask* | *mask-length*] [**longer-match**] [**verbose**] command, you can view the route to the specified destination IP address.
- Using the **display ip routing-table** *ip-address1* { *mask1* | *mask-length1* } *ip-address2* { *mask2* | *mask-length2* } [**verbose**] command, you can view the routes to the specified range of destination IP addresses.
- Using the **display ip routing-table protocol***protocol* [**inactive** | **verbose**] command, you can view the routes that are learnt by the specified protocol.
- Using the **display ip routing-table statistics** command, you can view general information about the routing table.
- Using the **display ip routing-table vpn-instance** *vpn-instance-name* command, you can view brief information about the VPN routing table.
- Using the **display ip routing-table vpn-instance** *vpn-instance-name* **verbose** command, you can view detailed information about the VPN routing table.
- Using the **display ip routing-table vpn-instance** *vpn-instance-name ip-address* [*mask* | *mask-length*] [**longer-match**] [**verbose**] command, you can view detailed information about the route to the specified destination IP address in the VPN routing table.
- Using the **display ip routing-table vpn-instance** *vpn-instance-name ip-address1* { *mask1* | *mask-length1* } *ip-address2* { *mask2* | *mask-length2* } [**verbose**] command, you can view detailed information about the routes to the specified network segment in the VPN routing table.

----End

1.7 Configuration Examples

IP routing configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

1.7.1 Example for Configuring IPv4 FRR of the Public Network

By configuring IPv4 FRR of the public network, you can enable traffic to be rapidly switched to the backup link when the primary link fails.

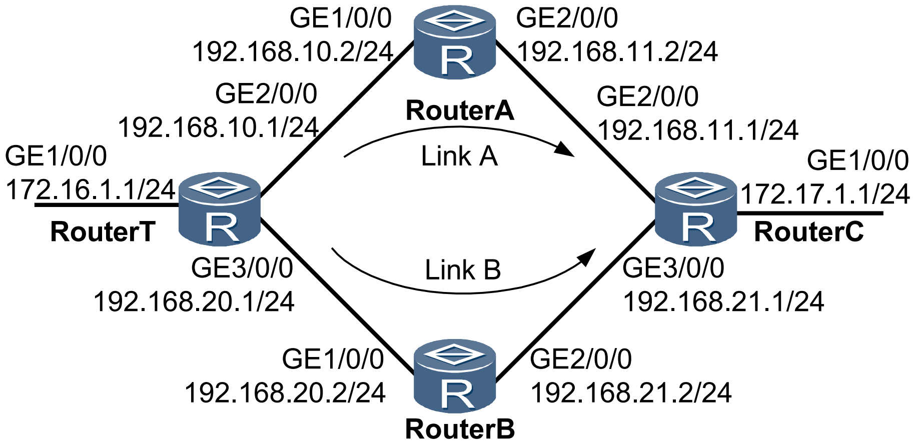
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 1-3**, it is required that the backup outbound interface and backup next hop be configured on Router T to ensure that Link B functions as the backup of Link A. When Link A fails, traffic is rapidly switched to the backup link, namely, Link B.

Figure 1-3 Networking diagram of configuring IPv4 FRR of the public network



Configuration Notes

When configuring IPv4 FRR of the public network, pay attention to the following:

- Ensure that at least two routes destined for the same destination address but discovered by different routing protocols.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on Router T, Router A, and Router C.
2. Configure basic IS-IS functions on Router T, Router B, and Router C.
3. Enable IPv4 FRR of the public network on Router T, and check the backup outbound interface and the backup next hop.
4. Disable IPv4 FRR, and check the backup outbound interface and the backup next hop.

Data Preparation

To complete the configuration, you need the following data:

- OSPF process IDs of Router T, Router A, and Router C
- IS-IS area addresses of Router T, Router B, and Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure OSPF on Router T, Router A, and Router C. The configuration details are not mentioned here.

Step 3 Configure IS-IS on Router T, Router B, and Router C. The configuration details are not mentioned here.

Step 4 Check routing information.

# Check the routes to the destination 172.17.1.0 on Router T.

```
<RouterT> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 1        Routes : 2

Destination: 172.17.1.0/24
     Protocol: OSPF            Process ID: 1
   Preference: 10                    Cost: 3
      NextHop: 192.168.10.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h00m07s
          Tag: 0                 Priority: low
        Label: NULL               QoSInfo: 0xa98ac7
   IndirectID: 0x40000041
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet2/0/0
     TunnelID: 0x0                  Flags: D

Destination: 172.17.1.0/24
     Protocol: ISIS            Process ID: 1
   Preference: 15                    Cost: 30
      NextHop: 192.168.20.2     Neighbour: 0.0.0.0
        State: Inactive Adv           Age: 00h01m26s
          Tag: 0                 Priority: high
        Label: NULL               QoSInfo: 0xa98ac7
   IndirectID: 0x80000081
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet3/0/0
     TunnelID: 0x0                  Flags: 0
```

In the routing table, you can view that there are two routes to the destination 172.17.1.0/24. The route with the next hop being 192.168.10.2 is the optimal route because the priority of an OSPF route is higher than the priority of an IS-IS route.

Step 5 Enable IPv4 FRR of the public network.

# Enable IPv4 FRR on Router T.

```
[~RouterT] ip frr
[~RouterT] commit
```

# Check the backup outbound interface and the backup next hop on Router T.

```
<RouterT> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 1        Routes : 2

Destination: 172.17.1.0/24
     Protocol: OSPF            Process ID: 1
   Preference: 10                    Cost: 3
      NextHop: 192.168.10.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h01m36s
          Tag: 0                 Priority: low
        Label: NULL               QoSInfo: 0xa98ac7
   IndirectID: 0x40000041
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet2/0/0
     TunnelID: 0x0                  Flags: D
    BkNextHop: 192.168.20.2   BkInterface: GigabitEthernet3/0/0
      BkLabel: NULL           SecTunnelID: 0x0
 BkPETunnelID: 0x0        BkPESecTunnelID: 0x0
 BkIndirectID: 0x80000081
```

```
Destination: 172.17.1.0/24
     Protocol: ISIS            Process ID: 1
   Preference: 15                    Cost: 30
      NextHop: 192.168.20.2     Neighbour: 0.0.0.0
        State: Inactive Adv           Age: 00h02m55s
          Tag: 0                 Priority: high
        Label: NULL               QoSInfo: 0xa98ac7
   IndirectID: 0x80000081
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet3/0/0
     TunnelID: 0x0                  Flags: 0
```

In the routing table, you can view that the route to the destination 172.17.1.0/24 has the backup outbound interface and the backup next hop, and the IS-IS route becomes the backup route.

Step 6 Verify the configuration.

# Simulate a link fault on Router T.

```
[~RouterT] interface gigabitethernet 2/0/0
[~RouterT-GigabitEthernet2/0/0] shutdown
[~RouterT-GigabitEthernet2/0/0] commit
[~RouterT-GigabitEthernet2/0/0] quit
```

# Check the routes to the destination 172.17.1.0/24 on Router T.

```
<RouterT> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 1        Routes : 1

Destination: 172.17.1.0/24
     Protocol: ISIS            Process ID: 1
   Preference: 15                    Cost: 30
      NextHop: 192.168.20.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h57m30s
          Tag: 0                 Priority: high
        Label: NULL               QoSInfo: 0xa98ac7
   IndirectID: 0x80000081
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet3/0/0
     TunnelID: 0x0                  Flags: D
```

When Link A fails, traffic is rapidly switched to the backup link, namely, Link B.

----End

Configuration Files

- Configuration file of Router T

```
#
sysname RouterT
#
ip frr
#
isis 1
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.10.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 192.168.20.1 255.255.255.0
 isis enable 1
#
ospf 1
 area 0.0.0.0
  network 192.168.10.0 0.0.0.255
 area 0.0.0.1
  network 172.16.1.0 0.0.0.255
#
return
```

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.10.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.11.2 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.10.0 0.0.0.255
  network 192.168.11.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.20.2 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.21.2 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
sysname RouterC
#
isis 1
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.11.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 192.168.21.1 255.255.255.0
 isis enable 1
#
ospf 1
```

```
 area 0.0.0.0
  network 192.168.11.0 0.0.0.255
  network 192.168.21.0 0.0.0.255
 area 0.0.0.2
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

1.4 Configuring IPv4 FRR

1.7.2 Example for Configuring IPv6 FRR for a Public Network

After IPv6 FRR is configured for a public network, traffic can be quickly switched to the backup link if the primary link fails.

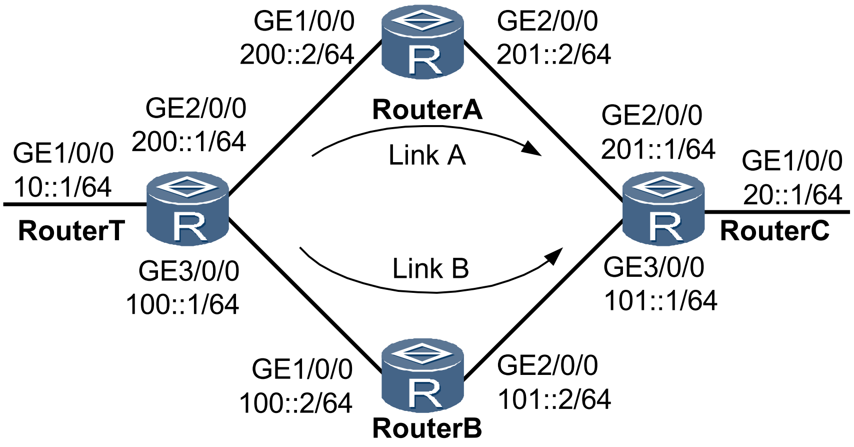
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 1-4**, the backup outbound interface and backup next hop must be configured on Router T so that link B functions as the backup of link A. In this manner, if link A becomes faulty, traffic can be switched quickly to link B, thus improving network reliability.

Figure 1-4 Networking diagram for configuring IPv6 FRR for the public network



Configuration Notes

When configuring IPv6 FRR, note the following point:

- Ensure that at least two routes destined for the same destination address but discovered by different routing protocols.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable OSPFv3 on Router T, Router A, and Router C.
2. Enable IPv6 IS-IS on Router T, Router B, and Router C.
3. Enable IPv6 FRR of the public network on Router T, and then check information about the backup outbound interface and backup next hop.
4. Disable IPv6 FRR, and then check information about the backup outbound interface and backup next hop.

Data Preparation

To complete the configuration, you need the following data:

- OSPFv3 process IDs of Router T, Router A, and Router C
- IPv6 IS-IS area addresses of Router T, Router B, and Router C

Procedure

Step 1 Configure an IPv6 address for each interface. The configuration details are not provided here.

Step 2 Configure OSPFv3 on Router T, Router A, and Router C. The configuration details are not provided here.

Step 3 Configure IPv6 IS-IS on Router T, Router B, and Router C. The configuration details are not provided here.

Step 4 Check routing information.

# On Router T, check the routes to 20::1/64.

```
<RouterT> display ipv6 routing-table 20::1 64 verbose
Routing Table : _public_
Summary Count : 2

Destination  : 20::                                   PrefixLength : 64
NextHop      : 200::2                                  Preference   : 10
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol     : OSPFv3
State        : Active Adv                              Cost         : 3
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0x69                                    Age          : 269sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 2/0/0                   Flags        : D

Destination  : 20::                                   PrefixLength : 64
NextHop      : 100::2                                  Preference   : 15
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol     : ISIS
State        : Inactive Adv                            Cost         : 30
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0xb5                                    Age          : 201sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 3/0/0                   Flags        : 0
```

The preceding command output shows that there are two routes to 20::1/64 and that the route with the next hop being 200::2 is preferred (because the preference of the OSPFv3 route is higher than that of the IPv6 IS-IS route).

Step 5 Enable IPv6 FRR of the public network.

# Enable IPv6 FRR on Router T.

```
[~RouterT] ipv6 frr
[~RouterT] commit
```

# Check information about the backup outbound interface and backup next hop on Router T.

```
<RouterT> display ipv6 routing-table 20::1 64 verbose
Routing Table : _public_
Summary Count : 2

Destination  : 20::                                    PrefixLength : 64
NextHop      : 200::2                                  Preference   : 10
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol     : OSPFv3
State        : Active Adv                              Cost         : 3
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0x69                                    Age          : 553sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 2/0/0                   Flags        : D
BkNextHop   : 100::2                                BkInterface : gigabitethernet
3/0/0
BkLabel      : NULL                                    BkTunnelID   : 0x0
BkPETunnelID : 0x0                                     BkIndirectID : 0xb5

Destination  : 20::                                    PrefixLength : 64
NextHop      : 100::2                                  Preference   : 15
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol     : ISIS
State        : Inactive Adv                            Cost         : 30
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0xb5                                    Age          : 485sec
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : gigabitethernet 3/0/0                  Flags        : 0
```

The preceding command output shows that the route to 20::1/64 has a backup outbound interface and a backup next hop (that is, the IPv6 IS-IS route becomes the backup route).

Step 6 Verify the configuration.

# Simulate a link fault on Router T.

```
[~RouterT] interface gigabitethernet 2/0/0
[~RouterT-GigabitEthernet2/0/0] shutdown
[~RouterT-GigabitEthernet2/0/0] commit
[~RouterT-GigabitEthernet2/0/0] quit
```

# On Router T, check the routes to 20::1/64.

```
<RouterT> display ipv6 routing-table 20::1 64 verbose
Routing Table : _public_
Summary Count : 1

Destination  : 20::                                    PrefixLength : 64
NextHop      : 100:2                                   Preference   : 15
Neighbour    : ::                                      ProcessID    : 1
Label        : NULL                                    Protocol    : ISIS
State        : Active Adv                              Cost         : 30
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
```

```
IndirectID   : 0xb5                                     Age          : 1279sec
RelayNextHop : ::                                       TunnelID     : 0x0
Interface    : gigabitethernet 3/0/0                    Flags        : D
```

The preceding command output shows that when link A becomes faulty, traffic is quickly switched to link B.

----End

Configuration Files

- Configuration file of Router T

```
#
sysname RouterT
#
ipv6 frr
#
isis 1
 is-level level-1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0001.00
#
ospfv3 1
 router-id 1.1.1.1
 area 0.0.0.0
 area 0.0.0.1
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 10::1/64
 ospfv3 1 area 0.0.0.1
 isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 200::1/64
 ospfv3 1 area 0.0.0.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 100::1/64
 isis enable 1
 isis ipv6 enable 1
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.0
#
return
```

- Configuration file of Router A

```
#
sysname RouterA
#
ospfv3 1
 router-id 2.2.2.2
 area 0.0.0.0
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 200::2/64
 ospfv3 1 area 0.0.0.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 201::2/64
 ospfv3 1 area 0.0.0.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.0
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 is-level level-1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 100::2/64
 isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 101::2/64
 isis ipv6 enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
 is-level level-1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0003.00
#
ospfv3 1
 router-id 1.1.1.1
 area 0.0.0.0
 area 0.0.0.2
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 20::1/64
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 201::1/64
 ospfv3 1 area 0.0.0.2
#
interface GigabitEthernet3/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 101::1/64
 isis enable 1
 isis ipv6 enable 1
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.0
#
return
```

Related Tasks

1.5 Configuring IPv6 FRR

2 Static Route Configuration

About This Chapter

Static routes are applicable to networks with simple structures. Proper configuration and usage of static routes improves the network performance and helps ensure the bandwidth requirement of important applications.

2.10 Configuring FRR for IPv6 Static Routes

FRR is applicable to IP services that are sensitive to delay and packet loss. FRR can be configured for IPv6 static routes to implement link protection.

2.11 Configuration Examples

This section provides configuration examples of static routes. Configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

2.1 Overview of Static Routes

By manually configuring static routes, you can implement the interworking of simple networks.

Static routes are a special kind of routes that need to be manually configured.

On a simple network, you only need to configure static routes so that the network can run properly. When a router cannot run dynamic routing protocols or is unable to generate routes to the destination network, you can configure static routes on the router. Proper configuration and usage of static routes improve the network performance and help ensure bandwidth for important services.

Route selection can be accurately controlled by configuring static routes. The disadvantage of static routes is that each time a fault occurs on the network or the network topology changes, you must configure the static routes again.

When configuring static routes, make sure that:

- Outbound interfaces are configured with IP addresses and are working properly.
- Next-hop IP addresses are reachable.

2.2 Static Route Features Supported by the NE5000E

The NE5000E supports various static routes, including IPv4 static routes, IPv6 static routes, default routes, as well as the Bidirectional Forwarding Detection (BFD) for static routes feature.

IPv4 Static Routes

The NE5000E supports common static routes and the static routes that are associated with VPN instances. Common static routes are used to implement the interworking of simple networks, and the static routes that are associated with VPN instances are used to manage VPN routes.

For details on VPN instances, refer to the *HUAWEI NetEngine5000E Core Router Feature Description - VPN.*

IPv6 Static Routes

Similar to IPv4 static routes, IPv6 static routes need to be manually configured by the administrator. IPv6 static routes are applicable to simple IPv6 networks.

The major differences between IPv6 static routes and IPv4 static routes are their destination addresses and next-hop addresses. A static IPv6 route uses an IPv6 address as the next hop, whereas a static IPv4 route use an IPv4 address as the next hop.

If the destination address of a static IPv6 route is ::/0 (the mask length being 0), this static IPv6 route is a default IPv6 route. If the destination address of an IPv6 packet fails to match any entry in the routing table, the router selects the default IPv6 route to forward the IPv6 packet.

Default Static Routes

Default routes are a special kind of routes. Generally, a default route can be manually configured.

The default route is used only when no entry is matched in the routing table. In a routing table, both the destination address and the mask of the default route are 0.0.0.0. You can check whether the default route is configured by running the **display ip routing-table** command.

If the destination address of a packet does not match any entry in the routing table, the router selects the default route to forward this packet. If no default route exists and the destination address of the packet does not match any entry in the routing table, the packet is discarded. An Internet Control Message Protocol (ICMP) packet is then sent, informing the originating host that the destination host or network is unreachable.

Floating Static Routes

Floating static routes have a lower priority than other static routes in the routing table. A floating static route is activated in the routing table only when the preferred route fails.

Load Balancing Among Static Routes

Load balancing among static routes is a feature that allows the router to transmit traffic by using all available static routes. Static routes do not have metrics, and are used for equal load balancing only, that is, the same proportion of traffic is load balanced on multiple static routes.

If both load-balancing and IP FRR are configured, load-balancing rather than IP FRR takes effect.

FRR on Static Routes

When routes are delivered to the routing management (RM) module, the optimal route is delivered along with a backup route. This ensures that when the optimal route fails, the Forwarding Information Base (FIB) is immediately switched to the backup route. As a result, traffic is fast switched from the failed optimal route to the backup route, reducing packet drop to a large extent.

You need to configure two routes with the same prefix but different preferences to implement FRR. The route with the higher preference is the primary route, and the route with the lower preference is the backup route. FRR is implemented only on static routes that are manually configured. That is, FRR is not implemented on iterated next hops.

BFD for Static Routes

Unlike dynamic routing protocols, static routes do not have a detection mechanism. When a fault occurs on a network, administrator interference is required. BFD for static routes is used to bind BFD sessions to IPv4 static routes on the public network. The BFD session is used to detect the status of the link of a static route.

After BFD for static routes is configured, each static route can be bound to a BFD session.

- If the BFD session on the link of a static route detects that the link changes from Up to Down, BFD reports the fault and then sets the route to inactive. The route becomes unavailable and is deleted from the IP routing table.
- When a BFD session is established on the link of a static route (the link changes from Down to Up), BFD reports the success and sets the route to active. The route becomes available and is added to the IP routing table.

2.3 Configuring IPv4 Static Routes

On a network, you can accurately control route selection by configuring IPv4 Static Routes.

Applicable Environment

IPv4 Static Routes can be configured for a network with simple structure to achieve connectivity.

The NE5000E supports ordinary static routes and the static routes associated with VPN instances. The static routes associated with VPN instances are used to manage VPN routes. For details on VPN instances, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN*.

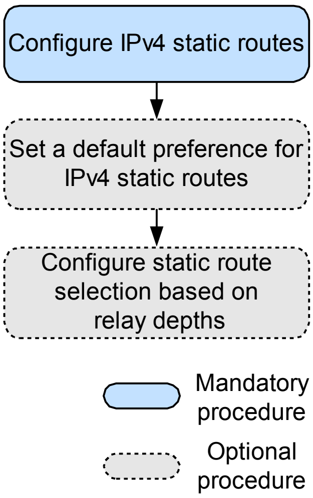
Pre-configuration Task

Before configuring IPv4 Static Routes, complete the following task:

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

Configuration Procedures

Figure 2-1 Flowchart of configuring IPv4 Static Routes


Configure IPv4 static routes
Set a default preference for IPv4 static routes
Configure static route selection based on relay depths
Mandatory procedure
Optional procedure

Related Tasks

2.11.1 Example for Configuring IPv4 Static Routes

2.3.1 Configuring IPv4 Static Routes

When configuring a IPv4 static route, you need to correctly configure its destination address, outbound interface, and next hop.

Context

When configuring a IPv4 static route, you need the following information:

- Destination address and mask

  In the **ip route-static** command, the IPv4 address is expressed in dotted decimal notation, and the mask is expressed either in dotted decimal notation or represented by the mask length.

- Outbound interface and next-hop address

  When configuring a static route, you can specify an outbound interface, a next-hop address, or both the outbound interface and the next hop-address, depending on actual requirements.

  Actually, every routing entry requires a next-hop address. When sending a packet, the device first searches for the matched route in the routing table according to the destination address. The link layer cannot find the associated link layer address to forward the packet unless the next-hop address of the packet is specified.

  When specifying an outbound interface, note the following:

  - For a Point-to-Point (P2P) interface, the next-hop address is specified while the outbound interface is specified. That is, the address of the remote interface (interface on the peer device) connected to this interface is the next-hop address. For example, when a POS interface is encapsulated with the Point-to-Point Protocol (PPP) and obtains the remote IP address (IP address of the interface on the peer device) through PPP negotiation, you need to specify only the outbound interface. The next-hop address is not required.
  - Non-Broadcast Multiple-Access (NBMA) interfaces (such as an ATM interface) are applicable to Point-to-Multipoint (P2MP) networks. Therefore, you need to configure IP routes and the mappings between IP addresses and link layer addresses. In this case, next-hop addresses need to be configured.
  - When configuring static routes, it is not recommended to specify the Ethernet interface or the virtual-template (VT) interface as the outbound interface. This is because the Ethernet interface is a broadcast interface whereas the VT interface can be associated with several virtual access (VA) interfaces. Therefore, if the Ethernet interface or the VT interface is specified as the outbound interface, multiple next hops exist and the system cannot decide which next hop is to be used. In actual applications, when specifying a broadcast interface (such as an Ethernet interface) or a VT interface as the outbound interface, you are recommended to specify the associated next-hop address.

- Other attributes

  Different static routes can be configured with different preferences so that routing management policies can be flexibly applied. For example, when configuring multiple routes to the same destination address, you can set the same preference for these routes to implement load balancing. You can also set different preferences to implement routing redundancy.

  When the **ip route-static** command is run to configure a static route, if the destination address and the mask are set to all 0s (0.0.0.0 0.0.0.0), it indicates that a default route is configured.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip route-static ip-address { mask | mask-length } { nexthop-address | interface-type interface-number [ nexthop-address ] | vpn-instance vpn-instance-name nexthop-address } [ preference preference ] [ tag tag ] [ description text ]
```

A IPv4 static route is configured.

By default, no IPv4 static route is configured.

NOTE

If the outbound interface of a static route is a broadcast interface or an NBMA interface, the next hop of the outbound interface must be specified.

In the scenario where static routes carry out load balancing, if a static route has an Ethernet interface as its outbound interface but does not have a next hop address, this static route cannot carry out load balancing with static routes that have next hop addresses. To enable this static route to carry out load balancing with other static routes that have next hop addresses, specify a next hop address for this route.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

2.3.2 (Optional) Setting the Default Preference for IPv4 Static Routes

By setting the default preference for IPv4 static routes, you can change the preference of the static routes.

Context

When an IPv4 static route is configured, the default preference is used if the preference of the static route is not specified. After the default preference is re-set, the new default preference is valid for both existing IPv4 static routes that adopt the default preference and new IPv4 static routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip route-static default-preference preference
```

The default preference of static routes is set.

By default, the preference of static routes is 60.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

2.3.3 (Optional) Configuring Static Route Selection Based on Relay Depths

To enable the system to select the static route with the smallest relay depth as the active route, you need to configure static route selection based on relay depths.

Context

Among static routes with the same prefix but different relay depths, static routes with lower relay depths are more stable. After static route selection based on relay depths is configured, the system selects the static route with the smallest relay depth as the active route and delivers it to the FIB table. The other routes become inactive.

To prevent a loop when GGSN services are forwarded between boards, you need to use the **ip route-static selection-rule relay-depth** command to configure static routes to be selected according to iteration depths.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip route-static selection-rule relay-depth
```

Static route selection that is based on relay depths is configured.

By default, static routes are not selected according to relay depths.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

2.3.4 Checking the Configuration

After IPv4 static routes are configured, you can check detailed information about the configured IPv4 static routes.

Prerequisite

The configurations of IPv4 static routes are complete.

Procedure

- Run the **display ip routing-table** command to check brief information about the IPv4 routing table.
- Run the **display ip routing-table verbose** command to check detailed information about the IPv4 routing table.

----End

Example

# Run the **display ip routing-table** command to check the preferences, next hops, and outbound interfaces of static routes.

```
<HUAWEI> display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 8        Routes : 8
Destination/Mask    Proto  Pre  Cost    Flags    NextHop         Interface
        0.0.0.0/0   Static 60   0         RD       1.1.4.2         Pos1/0/0
       1.1.1.0/24  Direct 0    0         D       1.1.1.1         GigabitEthernet2/0/0
        1.1.1.1/32  Direct 0    0          D       127.0.0.1       InLoopBack0
        1.1.4.0/30  Direct 0    0          D       1.1.4.1         Pos1/0/0
        1.1.4.1/32  Direct 0    0          D       127.0.0.1       InLoopBack0
        1.1.4.2/32  Direct 0    0          D       1.1.4.2         Pos1/0/0
      127.0.0.0/8   Direct 0    0          D       127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0          D       127.0.0.1       InLoopBack0
```

2.4 Configuring IPv6 Static Routes

On a network, you can accurately control route selection by configuring IPv6 static routes.

Applicable Environment

On a small IPv6 network, you can achieve network connectivity by configuring IPv6 static routes. Compared with the use of dynamic routing protocols, configuring static routes saves bandwidth.

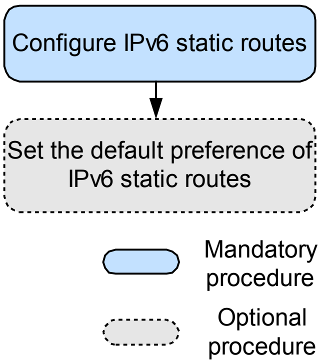
Pre-configuration Task

Before configuring an IPv6 static route, complete the following task:

- Configuring link layer protocol parameters and assigning IPv6 addresses to interfaces to ensure that the status of the link layer protocol of the interface is Up

Configuration Procedures

Figure 2-2 Flowchart of configuring IPv6 static routes


Configure IPv6 static routes
Set the default preference of IPv6 static routes
Mandatory procedure
Optional procedure

Related Tasks

2.11.7 Example for Configuring IPv6 Static Routes

2.4.1 Configuring IPv6 Static Routes

When configuring an IPv6 static route, configure the destination IP address, outbound interface, and next hop correctly.

Context

When configuring a static route, you need to specify either the outbound interface or the next-hop address according to the actual situation. If the outbound interface is a PPP interface, you can simply specify the outbound interface. If the outbound interface is a broadcast interface or an NBMA interface, you must also specify the next hop address in addition to specifying the outbound interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 route-static dest-ipv6-address prefix-length { interface-type interface-
number [ nexthop-ipv6-address ] | vpn-instance vpn-destination-name [ nexthop-ipv6-
address ] | nexthop-ipv6-address } [ preference preference | tag tag ]*
[ description text ]
```

A static IPv6 route is configured.

By default, no IPv6 static route is configured.

If the parameter **preference** is not specified, the default route preference is 60.

NOTE

In the scenario where static routes carry out load balancing, if a static route has an Ethernet interface as its outbound interface but does not have a next hop address, this static route cannot carry out load balancing with static routes that have next hop addresses. To enable this static route to carry out load balancing with other static routes that have next hop addresses, specify a next hop address for this route.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

2.4.2 (Optional) Setting the Default Preference for IPv6 Static Routes

By setting the default preference for IPv6 static routes, you can change the preference of the static routes.

Context

When an IPv6 static route is configured, the default preference is used if the preference of the static route is not specified. After the default preference is re-set, the new default preference is valid for both existing IPv6 static routes that adopt the default preference and new IPv6 static routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 route-static default-preference preference
```

The default preference of IPv6 static routes is set.

By default, the default preference of IPv6 static routes is 60.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

2.4.3 Checking the Configuration

After an IPv6 static route is configured, you can check detailed information about the route.

Prerequisite

The configurations of IPv6 static routes are complete.

Procedure

- Run the **display ipv6 routing-table** command to check brief information about the IPv6 routing table.
- Run the **display ipv6 routing-table verbose** command to check detailed information about the IPv6 routing table.

----End

Example

# Run the **display ipv6 routing-table** command to check the preferences, next hops, and outbound interfaces of static routes.

```
<HUAWEI> display ipv6 routing-table
Routing Table :
         Destinations : 5        Routes : 5

 Destination  : ::                                           PrefixLength : 0
 NextHop      : FE80::510A:0:8D7:1                           Preference   : 60
 Interface    : Pos1/0/0                                     Protocol     : Static
 State        : Active Adv                                   Cost         : 0
 Tunnel ID    : 0x0                                          Label        : NULL
 Age          : 685270sec

 Destination  : ::1                                          PrefixLength : 128
 NextHop      : ::1                                          Preference   : 0
 Interface    : InLoopBack0                                  Protocol     : Direct
 State        : Active NoAdv                                 Cost         : 0
 Tunnel ID    : 0x0                                          Label        : NULL
 Age          : 523sec
```

```
Destination  : 1::                                     PrefixLength : 64
NextHop      : 1::1                                    Preference   : 0
Interface    : GigabitEthernet2/0/0                    Protocol     : Direct
State        : Active Adv                              Cost         : 0
Tunnel ID    : 0x0                                     Label        : NULL
Age          : 523sec

Destination  : 1::1                                    PrefixLength : 128
NextHop      : ::1                                     Preference   : 0
Interface    : InLoopBack0                             Protocol     : Direct
State        : Active NoAdv                            Cost         : 0
Tunnel ID    : 0x0                                     Label        : NULL
Age          : 357sec

Destination  : FE80::                                  PrefixLength : 10
NextHop      : ::                                      Preference   : 0
Interface    : NULL0                                   Protocol     : Direct
State        : Active NoAdv                            Cost         : 0
Tunnel ID    : 0x0                                     Label        : NULL
Age          : 407sec
```

2.5 Configuring Dynamic BFD to detect IPv4 Static Routes

By configuring dynamic BFD to detect IPv4 static routes, you can enable devices to fast detect link changes and thus improve network reliability.

Applicable Environment

To use BFD sessions to provide link detection for IPv4 static routes in the public network, you can bind static routes to BFD sessions. One static route can be bound to one BFD session.

Preferred static routes are delivered to the forwarding table for packet forwarding. But a static route is incapable of detecting whether the next-hop link is working properly. You can bind static routes to BFD sessions. A BFD session can fast detect changes over a link and inform the routing management system of the changes. When a BFD session detects that a link is interrupted, the routing management system immediately removes the static route that is bound to the BFD session from the forwarding table and recalculates another active route. In this manner, fast route convergence is implemented.

Pre-configuration Task

Before configuring dynamic BFD to detect static routes, complete the following task:

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

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

Global BFD is enabled on the node.

Step 3 Run:

```
quit
```

Back to the system view.

Step 4 (Optional) Run:

```
ip route-static default-bfd [ min-rx-interval min-rx-interval ] [ min-tx-interval
min-tx-interval ] [ detect-multiplier multiplier ]
```

Global BFD parameters are configured for static routes.

By default, the values of the global BFD parameters **min-rx-interval**, **min-tx-interval**, and **detect-multiplier** are respectively 10ms, 10ms, and 3.

Step 5 Run:

```
ip route-static bfd [ interface-type interface-number ] nexthop-address [ local-
address address ] [ min-rx-interval min-rx-interval ] [ min-tx-interval min-tx-
interval ] [ detect-multiplier multiplier ]
```

The BFD parameters of an IPv4 static route are set.

NOTE

If *interface-type interface-number* is not set, **local-address** *address* must be specified.

If none of min-rx-interval, min-tx-interval, and detect-multiplier is specified, the global default values of the BFD parameters are used.

Step 6 Run:

```
ip route-static ip-address { mask | mask-length } { nexthop-address | interface-
type interface-number [ nexthop-address ] } [ preference preference ] bfd enable
[ description text ]
```

A public network static IPv4 route is bound to a BFD session.

Step 7 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display bfd session** { **all** | **discriminator** *discr-value* } [**verbose**] command to check information about the BFD session.
- Run the **display current-configuration** | **include bfd** command to check the configuration of BFD for static routes.

  You can check the information about a BFD session only after parameters of the BFD session are set and the BFD session is established.

  If the BFD session negotiation succeeds, you can view that the status of the BFD session is Up. Run the **display current-configuration** | **include bfd** command in the system view, and you can view that the BFD session is bound to a static route.

# Run the **display bfd session** command to check information about BFD sessions.

```
<HUAWEI> display bfd session all
--------------------------------------------------------------------------------
Local  Remote PeerIpAddr      State     Type      InterfaceName
--------------------------------------------------------------------------------
10     20     1.1.1.2         Up        S_IP_PEER   -
```

```
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
```

Related Tasks

2.11.2 Example for Configuring Dynamic BFD to Detect IPv4 Static Routes

2.6 Configuring Dynamic BFD to detect IPv6 Static Routes

Dynamic BFD for IPv6 static routes can quickly detect link changes, thus improving network reliability.

Applicable Environment

To use BFD sessions to detect links to which IPv6 static routes on the public network correspond, you can bind static routes to BFD sessions. An IPv6 static route can be bound to only one BFD session.

Preferred IPv6 static routes are delivered to the forwarding table for packet forwarding. An IPv6 static route, however, is incapable of detecting whether or not the link to the next hop is working properly. Binding the IPv6 static route to a BFD session can address this problem, because a BFD session is capable of detecting link changes and informing the routing management module of the changes. If a BFD session detects that a link is interrupted, the routing management module immediately withdraws the IPv6 static route that is bound to the BFD session from the forwarding table and recalculates another active route. In this manner, fast route convergence is implemented.

Pre-configuration Tasks

Before configuring dynamic BFD for IPv6 static routes, complete the following task:

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

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

bfd

Global BFD is enabled on the node.

Step 3 Run:

quit

Back to the system view.

Step 4 (Optional) Run:

ipv6 route-static default-bfd [**detect-multiplier** *multiplier*] [**min-rx-interval** *min-rx-interval*] [**min-tx-interval** *min-tx-interval*]

The global BFD parameters of the IPv6 static routes on the device are set.

The default values of the global BFD parameters *min-rx-interval*, *min-tx-interval*, and *multiplier* are 10 ms, 10 ms, and 3 respectively.

Step 5 Run:

```
ipv6 route-static bfd [ interface-type interface-number ] nexthop-address local-
address address [ detect-multiplier multiplier ] [ min-rx-interval min-rx-
interval ] [ min-tx-interval min-tx-interval ]
```

The BFD parameters of an IPv6 static route are set.

NOTE

If *interface-type interface-number* is not set, **local-address** *address* must be specified.

If none of min-rx-interval, min-tx-interval, and detect-multiplier is specified, the global default values of the BFD parameters are used.

Step 6 Run:

```
ipv6 route-static dest-ipv6-address prefix-length { interface-type interface-
number [ nexthop-ipv6-address ] | nexthop-ipv6-address } [ preference preference ]
[ tag tag ] bfd enable [ description text ]
```

A BFD session is bound to an IPv6 static route on the public network.

Step 7 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display bfd session** { **all** | **discriminator** *discr-value* } [**verbose**] command to check information about BFD sessions.
- Run the **display current-configuration** | **include bfd** command to check configurations of BFD for IPv6 static routes.

  Information about a BFD session can be viewed only after the parameters of the BFD session are set and the BFD session is established.

  If a BFD session is established, you can see that the status of the BFD session is Up. Run the **display current-configuration** | **include bfd** command in the system view, and you can see that the BFD session has been bound to a static route.

# Run the **display bfd session** command, and you can see that the BFD session is in the Up state. For example:

```
<HUAWEI> display bfd session all
--------------------------------------------------------------------------------
Local  Remote PeerIpAddr      State     Type      InterfaceName
--------------------------------------------------------------------------------
10     20      1::2           Up    S_IP_PEER  -
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
```

Related Tasks

2.11.3 Example for Configuring Dynamic BFD for IPv6 Static Routes

2.7 Configuring Static BFD to detect IPv4 Static Routes

Static BFD for IPv4 static routes enables a device to rapidly detect changes of a link to a destination address, improving network reliability.

Applicable Environment

To use BFD sessions to provide link detection for IPv4 static routes in the public network, you can bind static routes to BFD sessions. One static route can be bound to one BFD session.

Preferred static routes are delivered to the forwarding table for packet forwarding. But a static route is incapable of detecting whether each link to a destination address is working properly. You need to bind static routes to BFD sessions. A BFD session can fast detect changes over a link and inform the routing management module of the changes. When a BFD session detects that a link is interrupted, the routing management module immediately withdraws the static route that is bound to the BFD session from the forwarding table and recalculates another active route. This implements fast route convergence.

Pre-configuration Task

Before configuring static BFD to detect static routes, complete the following task:

- Configuring parameters of the link layer protocol and IP addresses for interfaces to ensure that the link layer protocol on the interfaces is Up
- Configuring a BFD Session, See the *HUAWEI NetEngine5000E Core Router Configuration Guide - Network Reliability*.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip route-static ip-address { mask | mask-length } { nexthop-address | interface-type interface-number [ nexthop-address ] } [ preference preference ] track bfd-session cfg-name [ description text ]
```

A public network static IPv4 route is bound to a BFD session.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display bfd session** { **all** | **discriminator** *discr-value* } [**verbose**] command to check information about the BFD session.

- Run the **display current-configuration | include bfd** command to check the configuration of BFD for static routes.

  You can check the information about a BFD session only after parameters of the BFD session are set and the BFD session is established.

  If the BFD session negotiation succeeds, you can view that the status of the BFD session is Up. Run the **display current-configuration | include bfd** command in the system view, and you can view that the BFD session is bound to a static route.

# Run the **display bfd session** command to check information about BFD sessions.

```
<HUAWEI> display bfd session all
--------------------------------------------------------------------------------
Local  Remote PeerIpAddr      State     Type      InterfaceName
--------------------------------------------------------------------------------
10     20     1.1.1.2         Up        S_IP_PEER   -
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
```

Related Tasks

2.11.4 Example for Configuring Static BFD for IPv4 Static Routes

2.8 Configuring Static BFD to detect IPv6 Static Routes

Static BFD for IPv6 static routes enables a device to rapidly detect changes of a link to a destination address, improving network reliability.

Applicable Environment

To use BFD sessions to detect links to which IPv6 static routes on the public network correspond, you can bind static routes to BFD sessions. An IPv6 static route can be bound to only one BFD session.

Preferred IPv6 static routes are delivered to the forwarding table for packet forwarding. But a IPv6 static route is incapable of detecting whether each link to a destination address is working properly. You need to bind IPv6 static routes to BFD sessions. A BFD session can fast detect changes over a link and inform the routing management module of the changes. When a BFD session detects that a link is interrupted, the routing management module immediately withdraws the IPv6 static route that is bound to the BFD session from the forwarding table and recalculates another active route. This implements fast route convergence.

Pre-configuration Tasks

Before configuring static BFD for IPv6 static routes, complete the following task:

- Configuring link layer protocol parameters and IP addresses for interfaces to ensure that the link layer protocol on the interfaces is Up
- Configuring a BFD Session, See the *HUAWEI NetEngine5000E Core Router Configuration Guide - Network Reliability*.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ipv6 route-static dest-ipv6-address prefix-length { interface-type interface-number [ nexthop-ipv6-address ] | vpn-instance vpn-destination-name [ nexthop-ipv6-address ] | nexthop-ipv6-address } [ preference preference ] [ tag tag ] track bfd-session cfg-name [ description text ]
```

A BFD session is bound to an IPv6 static route on the public network.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display bfd session** { **all** | **discriminator** *discr-value* } [**verbose**] command to check information about BFD sessions.
- Run the **display current-configuration** | **include bfd** command to check configurations of BFD for IPv6 static routes.

  Information about a BFD session can be viewed only after the parameters of the BFD session are set and the BFD session is established.

  If a BFD session is established, you can see that the status of the BFD session is Up. Run the **display current-configuration** | **include bfd** command in the system view, and you can see that the BFD session has been bound to a static route.

# Run the **display bfd session** command, and you can see that the BFD session is in the Up state. For example:

```
<HUAWEI> display bfd session all
--------------------------------------------------------------------------------
Local  Remote PeerIpAddr      State     Type      InterfaceName
--------------------------------------------------------------------------------
10    20      1::2            Up     S_IP_PEER  -
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
```

Related Tasks

2.11.5 Example for Configuring Static BFD for IPv6 Static Routes

2.9 Configuring FRR for IPv4 Static Routes

FRR is applicable to IP services that are sensitive to packet delay and packet loss. FRR can be configured for IPv4 static routes to implement traffic protection by use of a backup link.

Applicable Environment

FRR is applicable to IP services that are sensitive to delay and packet loss. FRR minimizes the impact of link faults on services.

Different static routes can be configured with different preferences to implement FRR flexibly on static routes. By specifying different preferences for multiple routes to the same destination, route backup is implemented.

Pre-configuration Task

Before configuring FRR for IPv4 static routes, complete the following task:

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

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip route-static frr
```

FRR is enabled for public network IPv4 static routes.

NOTE

To implement route backup by configuring FRR for static routes, you need to specify different preferences for these static routes.

In the scenario where static route FRR and BFD are used, if a static route has an Ethernet interface as its outbound interface but does not have a next hop address, this static route cannot implement FRR with static routes that have next hop addresses. To enable this static route to implement FRR with other static routes that have next hop addresses, specify a next hop address for this route.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display ip routing-table verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.
- Run the **display ip routing-table** *ip-address* [*mask* | *mask-length*] [**longer-match**] **verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.
- Run the **display ip routing-table** *ip-address1* { *mask1* | *mask-length1* } *ip-address2* { *mask2* | *mask-length2* } **verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.

# Run the **display ip routing-table verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.

```
<HUAWEI>display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
```

```
Routing Table : _public_
Summary Count : 1

Destination: 172.17.1.0/24
     Protocol: OSPF          Process ID: 1
   Preference: 60                    Cost: 0
      NextHop: 192.168.10.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h00m06s
          Tag: 0                 Priority: 0
        Label: NULL               QoSInfo: 0x0
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet2/0/0
     TunnelID: 0x0                  Flags:  D
    BkNextHop: 192.168.20.2   BkInterface: GigabitEthernet3/0/0
      BkLabel: 0              SecTunnelID: 0x0
 BkPETunnelID: 0x0        BkPESecTunnelID: 0x0
```

Related Tasks

2.11.6 Example for Configuring FRR for IPv4 Static Routes on the Public Network

2.10 Configuring FRR for IPv6 Static Routes

FRR is applicable to IP services that are sensitive to delay and packet loss. FRR can be configured for IPv6 static routes to implement link protection.

Applicable Environment

FRR is applicable to IP services that are sensitive to delay and packet loss. FRR minimizes the impact of link faults on services.

Different static routes can be configured with different preferences to implement FRR flexibly on static routes. By specifying different preferences for multiple routes to the same destination, route backup is implemented.

Pre-configuration Task

Before configuring FRR for IPv6 static routess, complete the following task:

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

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

ipv6 route-static frr

FRR is enabled for public network IPv6 static routess.

NOTE

To implement route backup by configuring FRR for static routes, you need to specify different preferences for these static routes.

Step 3 Run:

commit

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display ipv6 routing-table verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.
- Run the **display ipv6 routing-table** *ipv6-address* [*prefix-length*] [**longer-match**] **verbose** command to check information about the backup outbound interface and the backup next hop in the routing table.

# Run the **display ipv6 routing-table** command to check information about the backup outbound interface and the backup next hop in the routing table.

```
<HUAWEI> display ipv6 routing-table 10::1 verbose
Routing Table : _public_
Summary Count : 1

Destination  : 9::                                     PrefixLength : 64
NextHop      : 10::2                                   Preference   : 40
Neighbour    : ::                                      ProcessID    : 0
Label        : NULL                                    Protocol     : Static
State        : Active Adv Relied                       Cost         : 0
Entry ID     : 0                                       EntryFlags   : 0x00000000
Reference Cnt: 0                                       Tag          : 0
IndirectID   : 0x2c                                    Age          : 41sec
RelayNextHop : 10::2                                   TunnelID     : 0x0
Interface   : Pos1/0/0                                Flags        : RD
BkNextHop   : 20::2                                   BkInterface : Pos2/0/0
BkLabel      : NULL                                    BkTunnelID   : 0x0
BkPETunnelID : 0x0                                     BkIndirectID : 0x2d
```

2.11 Configuration Examples

This section provides configuration examples of static routes. Configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

2.11.1 Example for Configuring IPv4 Static Routes

You can configure IPv4 static routes to interconnect any two devices on an IPv4 network.

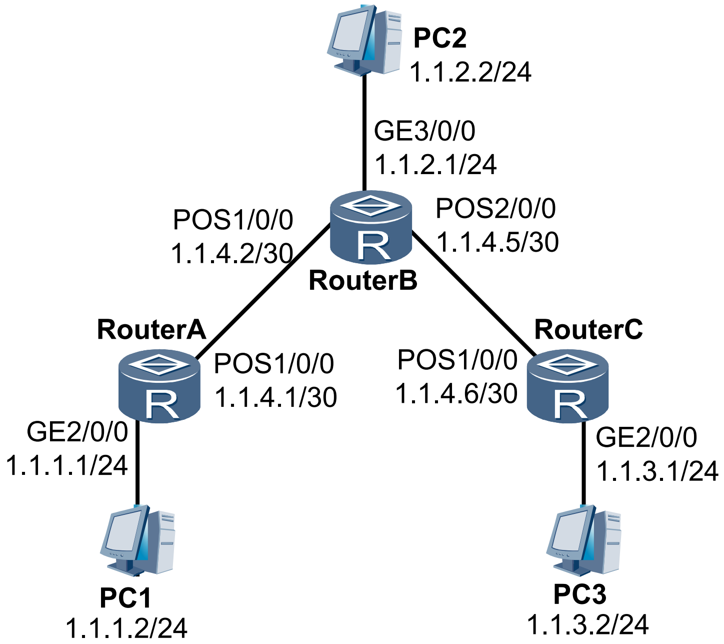
Networking Requirements

⚠ **CAUTION**

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

Figure 2-3 shows the IP addresses and masks of interfaces and hosts on routers. It is required that any two hosts in Figure 2-3 communicate through static routes.

Figure 2-3 Networking diagram of configuring IPv4 static routes



Configuration Notes

When configuring IPv4 static routes, pay attention to the following:

- If the outbound interface is a broadcast interface, specify the next-hop address.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IPv4 addresses for interfaces on each router for interworking.
2. Configure the IPv4 static route and the default route to the destination address on each router.
3. Configure the IPv4 default gateway on each host to make any two hosts communicate.

Data Preparation

To complete the configuration, you need the following data:

- Default route with the next hop being 1.1.4.2 on Router A
- Static route with the destination address being 1.1.1.0 and next hop being 1.1.4.1 on Router B
- Static route with the destination address being 1.1.3.0 and next hop being 1.1.4.6 on Router B

- Default route with the next hop being 1.1.4.5 on Router C
- Default gateways of PC1, PC2, and PC3

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure a static route.

# Configure an IPv4 default route on Router A.

```
[~RouterA] ip route-static 0.0.0.0 0.0.0.0 1.1.4.2
[~RouterA] commit
```

# Configure two IPv4 static routes on Router B.

```
[~RouterB] ip route-static 1.1.1.0 255.255.255.0 1.1.4.1
[~RouterB] ip route-static 1.1.3.0 255.255.255.0 1.1.4.6
[~RouterB] commit
```

# Configure an IPv4 default route on Router C.

```
[~RouterC] ip route-static 0.0.0.0 0.0.0.0 1.1.4.5
[~RouterC] commit
```

Step 3 Configure hosts.

Configure the default gateways of PC1, PC2, and PC3 as 1.1.1.1, 1.1.2.1, and 1.1.3.1 respectively.

Step 4 Verify the configuration.

# Check the IP routing table of Router A.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 8        Routes : 8
Destination/Mask    Proto  Pre  Cost   Flags    NextHop         Interface
        0.0.0.0/0   Static 60   0        RD       1.1.4.2         Pos1/0/0
       1.1.1.0/24  Direct 0    0        D      1.1.1.1        GigabitEthernet2/0/0
        1.1.1.1/32  Direct 0    0         D       127.0.0.1       InLoopBack0
        1.1.4.0/30  Direct 0    0         D       1.1.4.1         Pos1/0/0
        1.1.4.1/32  Direct 0    0         D       127.0.0.1       InLoopBack0
        1.1.4.2/32  Direct 0    0         D       1.1.4.2         Pos1/0/0
      127.0.0.0/8   Direct 0    0         D       127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0         D       127.0.0.1       InLoopBack0
```

# Run the **ping** command to verify the connectivity.

```
[~RouterA] ping 1.1.3.1
  PING 1.1.3.1: 56  data bytes, press CTRL_C to break
    Reply from 1.1.3.1: bytes=56 Sequence=1 ttl=254 time=62 ms
    Reply from 1.1.3.1: bytes=56 Sequence=2 ttl=254 time=63 ms
    Reply from 1.1.3.1: bytes=56 Sequence=3 ttl=254 time=63 ms
    Reply from 1.1.3.1: bytes=56 Sequence=4 ttl=254 time=62 ms
    Reply from 1.1.3.1: bytes=56 Sequence=5 ttl=254 time=62 ms
  --- 1.1.3.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 62/62/63 ms
```

# Run the **tracert** command to verify the connectivity.

```
[~RouterA] tracert 1.1.3.1
```

```
traceroute to  1.1.3.1(1.1.3.1), max hops: 30 ,packet length: 40
1 1.1.4.2 31 ms  32 ms  31 ms
2 1.1.4.6 62 ms  63 ms  62 ms
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 1.1.1.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 1.1.4.1 255.255.255.252
#
ip route-static 0.0.0.0 0.0.0.0 1.1.4.2
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 1.1.2.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 1.1.4.2 255.255.255.252
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 1.1.4.5 255.255.255.252
#
ip route-static 1.1.1.0 255.255.255.0 1.1.4.1
ip route-static 1.1.3.0 255.255.255.0 1.1.4.6
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 1.1.3.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 1.1.4.6 255.255.255.252
#
ip route-static 0.0.0.0 0.0.0.0 1.1.4.5
#
return
```

Related Tasks

2.3 Configuring IPv4 Static Routes

2.11.2 Example for Configuring Dynamic BFD to Detect IPv4 Static Routes

Configuring dynamic BFD to detect IPv4 static routes can ensure that link failures are detected rapidly.

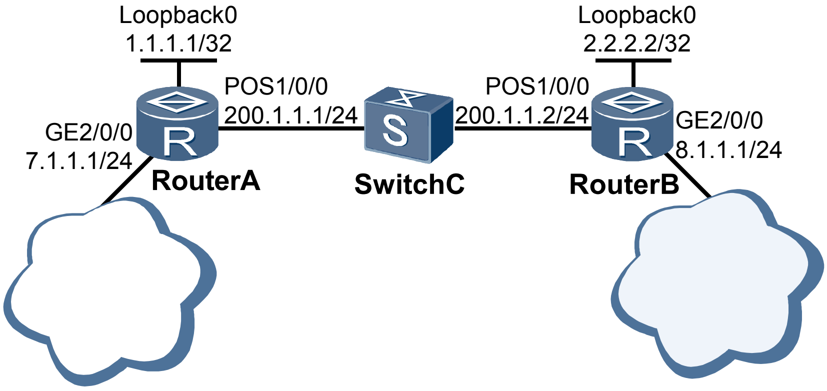
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-4**, Router A is connected to Router B through Switch C. It is required to configure the static default route on Router A to ensure that Router A can communicate with other Routers. A BFD session needs to be established between Router A and Router B to detect link failures.

Figure 2-4 Networking diagram of configuring dynamic BFD to detect static routes



Configuration Notes

When configuring dynamic BFD to detect static routes, pay attention to the following:

- Before configuring dynamic BFD to detect static routes, enable BFD globally.
- When configuring dynamic BFD to detect static routes, ensure that the parameters configured on the two ends of a BFD session are consistent.

Configuration Roadmap

The configuration roadmap is as follows:

1. On Router A, configure IPv4 static routes to Router B.
2. Configure dynamic BFD for static routes.

Data Preparation

To complete the configuration, you need the following data:

- Peer IP address to be detected by BFD
- Default values of the minimum interval for sending BFD Control packets, the minimum interval for receiving BFD Control packets, and the local detection multiplier

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure static routes.

# On Router A, configure a static route to 2.2.2.2/32.

```
[~RouterA] ip route-static 2.2.2.2 32 200.1.1.2
[~RouterA] commit
```

# Check the IP routing table of Router A, and you can view that the static route exists in the routing table.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 6        Routes : 6

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  Direct 0    0             D  127.0.0.1       LoopBack0
        2.2.2.2/32  Static 60   0             RD 200.1.1.2       Pos1/0/0
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
      200.1.1.0/24  Direct 0    0             D  200.1.1.1       Pos1/0/0
      200.1.1.1/32  Direct 0    0             D  127.0.0.1       Pos1/0/0
      200.1.1.2/32  Direct 0    0             D  200.1.1.2       Pos1/0/0
```

# On Router B, configure a static route to 1.1.1.1/32.

```
[~RouterB] ip route-static 1.1.1.1 32 200.1.1.1
[~RouterB] commit
```

Step 3 Configure dynamic BFD to detect static routes.

# On Router A, bind the static route to the BFD session.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] ip route-static bfd 200.1.1.2 local-address 200.1.1.1
[~RouterA] ip route-static 2.2.2.2 32 200.1.1.2 bfd enable
[~RouterA] commit
```

# On Router B, bind the static route to the BFD session.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] ip route-static bfd 200.1.1.1 local-address 200.1.1.2
[~RouterB] ip route-static 1.1.1.1 32 200.1.1.1 bfd enable
[~RouterB] commit
```

Step 4 Verify the configuration.

# After the preceding configuration, on Router A and Router B, you can view that the BFD session has been established and is in the Up state, and the static routes have been bound to the BFD session.

Take the display on Router A as an example.

```
[~RouterA] display bfd session all verbose
------------------------------------------------------------------------------
  (Multi Hop) State : Up                    Name : dyn_8193
------------------------------------------------------------------------------
  Local Discriminator    : 8193             Remote Discriminator   : 8193
  Session Detect Mode    : Asynchronous Mode Without Echo Function
  BFD Bind Type          : Peer IP Address
  Bind Session Type      : Dynamic
  Bind Peer IP Address   : 200.1.1.2
  Bind Interface         : -
  Bind Source IP Address : 200.1.1.1
  FSM Board Id           : 0                TOS-EXP                : 6
  Min Tx Interval (ms)   : 10               Min Rx Interval (ms)   : 10
  Actual Tx Interval (ms): 10               Actual Rx Interval (ms): 10
  Local Detect Multi     : 3                Detect Interval (ms)   : 30
  Echo Passive           : Disable          Acl Number             : -
  Destination Port       : 4784             TTL                    : 253
  Proc Interface Status  : Disable          Process PST            : Disable
  WTR Interval (ms)      : 0                Local Demand Mode      : Disable
  Active Multi           : 3
  Last Local Diagnostic  : No Diagnostic
  Bind Application       : STATICRT
  Session TX TmrID       : 0                Session Detect TmrID   : 0
  Session Init TmrID     : -                Session WTR TmrID      : -
  Session Echo Tx TmrID  : -
  Session Description    : -
------------------------------------------------------------------------------

     Total UP/DOWN Session Number : 1/0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 7.1.1.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
 interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
ip route-static bfd 200.1.1.2 local-address 200.1.1.1
ip route-static 2.2.2.2 32 200.1.1.2 bfd enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
bfd
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 8.1.1.1 255.255.255.0
```

```
#
interface Pos1/0/0
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
 interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
ip route-static bfd 200.1.1.1 local-address 200.1.1.2
ip route-static 1.1.1.1 32 200.1.1.1 bfd enable
#
return
```

Related Tasks

2.5 Configuring Dynamic BFD to detect IPv4 Static Routes

2.11.3 Example for Configuring Dynamic BFD for IPv6 Static Routes

Dynamic BFD for IPv6 static routes can quickly detect link faults.

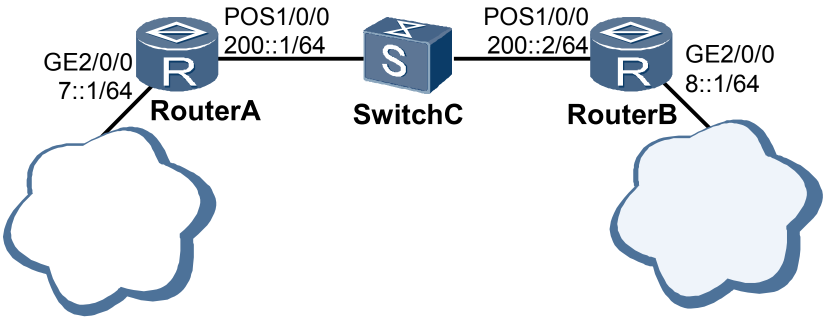
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-5**, Router A is connected to Router B through Switch C. After being configured with static default routes, Router A can communicate with other devices. A BFD session is set up between Router A and Router B to detect any fault in the link between the two devices.

Figure 2-5 Networking diagram for configuring dynamic BFD for IPv6 static routes



Configuration Notes

When configuring dynamic BFD for IPv6 static routes, note the following points:

- Before enabling dynamic BFD for IPv6 static routes, enable BFD globally.
- When enabling dynamic BFD for IPv6 static routes, ensure that the parameters configured on the two ends of a BFD session are consistent.

Configuration Roadmap

The configuration roadmap is as follows:

1. On Router A, configure IPv6 static routes to Router B.
2. Configure dynamic BFD for IPv6 static routes.

Data Preparation

To complete the configuration, you need the following data:

- IPv6 address of the remote end of a BFD session
- Default values of the minimum interval for sending BFD control packets, the minimum interval for receiving BFD control packets, and the local time multiplier

Procedure

Step 1 Configure an IPv6 address for each interface. The configuration details are not provided here.

Step 2 Configure IPv6 static routes.

# On Router A, configure the static routes to 8::1/64.

```
[~RouterA] ipv6 route-static 8:: 64 200::2
[~RouterA] commit
```

# On Router A, check the IPv6 routing table. You can see that static routes exist in the IPv6 routing table.

```
[~RouterA] display ipv6 routing-table
Routing Table : _public_
         Destinations : 6       Routes : 6

Destination  : 7::                                     PrefixLength : 64
NextHop      : 7::1                                    Preference   : 0
Cost         : 0                                       Protocol     : Direct
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : GigabitEthernet2/0/0                    Flags        : D

Destination  : 7::1                                    PrefixLength : 128
NextHop      : ::1                                     Preference   : 0
Cost         : 0                                       Protocol     : Direct
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : GigabitEthernet2/0/0                    Flags        : D

Destination  : 8::                                    PrefixLength : 64
NextHop      : 200::2                                  Preference   : 60
Cost         : 0                                       Protocol     : Static
RelayNextHop : 200::2                                  TunnelID     : 0x0
Interface    : Pos1/0/0                                Flags        : RD

Destination  : 200::                                   PrefixLength : 64
NextHop      : 200::1                                  Preference   : 0
Cost         : 0                                       Protocol     : Direct
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : Pos1/0/0                                Flags        : D

Destination  : 200::1                                  PrefixLength : 128
```

```
NextHop      : ::1                                    Preference   : 0
Cost         : 0                                      Protocol     : Direct
RelayNextHop : ::                                     TunnelID     : 0x0
Interface    : Pos1/0/0                               Flags        : D

Destination  : FE80::                                 PrefixLength : 10
NextHop      : ::                                     Preference   : 0
Cost         : 0                                      Protocol     : Direct
RelayNextHop : ::                                     TunnelID     : 0x0
Interface    : NULL0                                  Flags        : D
```

# On Router B, configure the static routes to 7::1/64.

```
[~RouterB] ipv6 route-static 7:: 64 200::1
[~RouterB] commit
```

Step 3 Configure dynamic BFD for static routes.

# On Router A, bind a static route to a BFD session.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] ipv6 route-static bfd 200::2 local-address 200::1
[~RouterA] ipv6 route-static 8:: 64 200::2 bfd enable
[~RouterA] commit
```

# On Router B, bind a static route to a BFD session.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] ipv6 route-static bfd 200::1 local-address 200::2
[~RouterB] ipv6 route-static 7:: 64 200::1 bfd enable
[~RouterB] commit
```

Step 4 Verify the configuration.

# After the configuration is complete, you can see that a BFD session has been set up between Router A and Router B, its status is Up, and a static route is bound to it.

Use the display on Router A as an example.

```
[~RouterA] display bfd session all verbose
------------------------------------------------------------------------------
  (Multi Hop) State : Up                  Name : dyn_16385
------------------------------------------------------------------------------
  Local Discriminator    : 16385            Remote Discriminator   : 16385
  Session Detect Mode    : Asynchronous Mode Without Echo Function
  BFD Bind Type          : Peer IP Address
  Bind Session Type      : Dynamic
  Bind Peer IP Address  : 200::2
  Bind Interface         : -
  Bind Source IP Address : 200::1
  FSM Board Id           : 3                TOS-EXP                : 6
  Min Tx Interval (ms) :10              Min Rx Interval (ms)  :10
  Actual Tx Interval (ms): 10               Actual Rx Interval (ms): 10
  Local Detect Multi   : 3             Detect Interval (ms)   : 30
  Echo Passive           : Disable          Acl Number             : -
  Destination Port       : 4784             TTL                    : 253
  Proc Interface Status  : Disable          Process PST            : Disable
  WTR Interval (ms)      : 0                Local Demand Mode      : Disable
  Active Multi           : 3
  Last Local Diagnostic  : No Diagnostic
  Bind Application       : STATICRTV6
  Session TX TmrID       : 0                Session Detect TmrID   : 0
  Session Init TmrID     : -                Session WTR TmrID      : -
  Session Echo Tx TmrID  : -
  Session Description    : -
------------------------------------------------------------------------------
```

```
    Total UP/DOWN Session Number : 1/0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 address 7::1/64
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 address 200::1/64
#
ipv6 route-static bfd 200::2 local-address 200::1
ipv6 route-static 8:: 64 200::2 bfd enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
bfd
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 address 8::1/64
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 address 200::2/64
#
ipv6 route-static bfd 200::1 local-address 200::2
ipv6 route-static 7:: 64 200::1 bfd enable
#
return
```

Related Tasks

2.6 Configuring Dynamic BFD to detect IPv6 Static Routes

2.11.4 Example for Configuring Static BFD for IPv4 Static Routes

To improve network reliability, you can configure static BFD for static route to rapidly detect link faults and speed up route convergence.

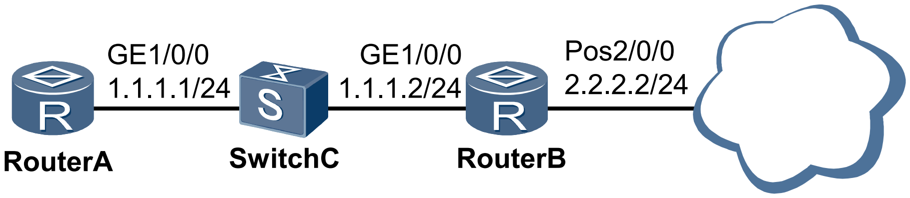
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-6**, Router A is connected to Router B through Switch C. A default static route is configured on Router A so that Router A can communicate with other routers. Additionally, a BFD session is configured between Router A and Router B to detect the link between the two devices.

Figure 2-6 Networking diagram for configuring static BFD for IPv4 static routes



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a BFD session between Router A and Router B to detect the link between the two devices.
2. Configure a default static route from Router A to the external network and bind the default static route to the BFD session.

Data Preparation

To complete the configuration, you need the following data:

- Peer IP address to be detected by BFD
- Local discriminator and remote discriminator of a BFD session
- Default values of BFD parameters, including minimum intervals for sending and receiving BFD control packets and local detection multiplier

Procedure

Step 1 Configure an IP address for each interface.

The configuration details are not described here.

Step 2 Configure a BFD session between Router A and Router B.

# On Router A, configure a BFD session with Router B.

```
<RouterA> system-view
```

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] bfd aa bind peer-ip 1.1.1.2
[~RouterA-bfd-session-aa] discriminator local 10
[~RouterA-bfd-session-aa] discriminator remote 20
[~RouterA-bfd-session-aa] commit
[~RouterA-bfd-session-aa] quit
```

# On Router B, configure a BFD session with Router A.

```
<RouterB> system-view
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] bfd bb bind peer-ip 1.1.1.1
[~RouterB-bfd-session-bb] discriminator local 20
[~RouterB-bfd-session-bb] discriminator remote 10
[~RouterB-bfd-session-bb] commit
[~RouterB-bfd-session-bb] quit
```

Step 3 Configure a default static route and bind it to a BFD session.

# On Router A, configure a default static route to the external network and bind it to a BFD session named **aa**.

```
[~RouterA] ip route-static 0.0.0.0 0 1.1.1.2 track bfd-session aa
```

Step 4 Verify the configuration.

# After the configuration, run the **display bfd session all** command on Router A and Router B. The command output shows that a BFD session has been established and is in the Up state. Then, run the **display current-configuration | include bfd** command in the system view. The command output shows that the default static route has been bound to the BFD session.

Take the display on Router A as an example.

```
[~RouterA] display bfd session all
--------------------------------------------------------------------------------
Local  Remote PeerIpAddr      State     Type      InterfaceName
--------------------------------------------------------------------------------
10    20     1.1.1.2          Up       S_IP_PEER   -
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
[~RouterA] display current-configuration | include bfd
 bfd
bfd aa bind peer-ip 1.1.1.2
 ip route-static 0.0.0.0 0.0.0.0 1.1.1.2 track bfd-session aa
```

# Check the IP routing table of Router A. The command output shows that the static route exists in the routing table.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 3        Routes : 3
Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface
        0.0.0.0/0   Static 60   0          RD  1.1.1.2        GigabitEthernet1/0/0
        1.1.1.0/24  Direct 0    0           D  1.1.1.1        GigabitEthernet1/0/0
        1.1.1.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
```

# Run the **shutdown** command on GE 1/0/0 of Router B to simulate a link fault.

```
[~RouterB] interface GigabitEthernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] shutdown
```

# Check the IP routing table of Router A. The command output shows that default route 0.0.0.0/0 does not exist. This is because the default static route is bound to a BFD session. When BFD detects a link fault, BFD rapidly notifies that the bound static route becomes unavailable.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 2        Routes : 2
Destination/Mask    Proto  Pre  Cost      Flags NextHop         Interface
        1.1.1.0/24  Direct 0    0           D  1.1.1.1         GigabitEthernet1/0/0
        1.1.1.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.1 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 1.1.1.2 track bfd-session aa
#
bfd aa bind peer-ip 1.1.1.2
 discriminator local 10
 discriminator remote 20
#
return
```

- Configuration file of Router B

```
sysname RouterB
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 2.2.2.2 255.255.255.0
#
bfd bb bind peer-ip 1.1.1.1
 discriminator local 20
 discriminator remote 10
#
return
```

Related Tasks

2.7 Configuring Static BFD to detect IPv4 Static Routes

2.11.5 Example for Configuring Static BFD for IPv6 Static Routes

To improve IPv6 network reliability, you can configure static BFD for IPv6 static route to rapidly detect link faults and speed up route convergence.

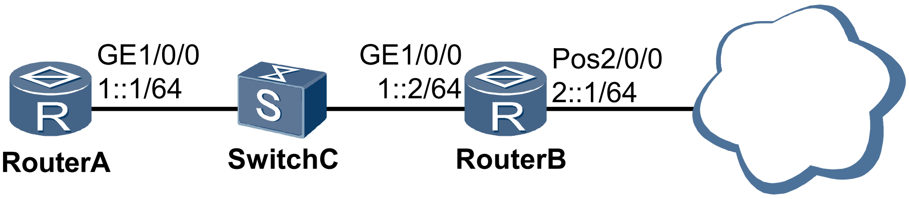
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-7**, Router A is connected to Router B through Switch C. A default static route is configured on Router A so that Router A can communicate with other routers. Additionally, a BFD session is configured between Router A and Router B to detect the link between the two devices.

Figure 2-7 Networking diagram for configuring static BFD for IPv6 static routes



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a BFD session on Router A and Router B to detect the link between the two devices.
2. Configure a default static route from Router A to the external network and bind the default static route to a BFD session.

Data Preparation

To complete the configuration, you need the following data:

- Peer IPv6 address to be detected by BFD
- Local discriminator and remote discriminator of a BFD session
- Default values of BFD parameters, including minimum intervals for sending and receiving BFD control packets and local detection multiplier

Procedure

Step 1 Configure an IPv6 address for each interface.

The configuration details are not described here.

Step 2 Configure a BFD session between Router A and Router B.

# On Router A, configure a BFD session with Router B.

```
<RouterA> system-view
```

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] bfd aa bind peer-ipv6 1::2
[~RouterA-bfd-session-aa] discriminator local 10
[~RouterA-bfd-session-aa] discriminator remote 20
[~RouterA-bfd-session-aa] commit
[~RouterA-bfd-session-aa] quit
```

# On Router B, configure a BFD session with Router A.

```
<RouterB> system-view
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] bfd bb bind peer-ipv6 1::1
[~RouterB-bfd-session-bb] discriminator local 20
[~RouterB-bfd-session-bb] discriminator remote 10
[~RouterB-bfd-session-bb] commit
[~RouterB-bfd-session-bb] quit
```

Step 3 Configure a default static route and bind it to a BFD session.

# On Router A, configure a default static route to the external network and bind it to a BFD session named **aa**.

```
[~RouterA] ipv6 route-static 0::0 0 1::2 track bfd-session aa
```

Step 4 Verify the configuration.

# After the configuration, run the **display bfd session all** command on Router A and Router B. The command output shows that a BFD session has been established and is in the Up state. Then, run the **display current-configuration | include bfd** command in the system view. The command output shows that the default static route has been bound to the BFD session.

Take the display on Router A as an example.

```
[~RouterA] display bfd session all
--------------------------------------------------------------------------------
Local Remote PeerIpAddr      State     Type        InterfaceName
--------------------------------------------------------------------------------
10    20     1::2            Up       S_IP_PEER   -
--------------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
[~RouterA] display current-configuration | include bfd
 bfd
bfd aa bind peer-ipv6 1::2
 ipv6 route-static :: 0 1::2 track bfd-session aa
```

# Check the IP routing table of Router A. The command output shows that the static route exists in the routing table.

```
[~RouterA] display ipv6 routing-table
Routing Table : _public_
         Destinations : 5        Routes : 5

 Destination  : ::                               PrefixLength : 0
 NextHop      : 1::2                             Preference   : 60
 Cost         : 0                                Protocol     : Static
 RelayNextHop : ::                               TunnelID     : 0x0
 Interface    : GigabitEthernet 1/0/0            Flags        : RD

 Destination  : ::1                              PrefixLength : 128
 NextHop      : ::1                              Preference   : 0
 Cost         : 0                                Protocol     : Direct
 RelayNextHop : ::                               TunnelID     : 0x0
 Interface    : InLoopBack0                      Flags        : D

 Destination  : 1::                              PrefixLength : 64
 NextHop      : 1::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : GigabitEthernet 1/0/0           Flags        : D

 Destination  : 1::1                            PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : FE80::                          PrefixLength : 10
 NextHop      : ::                              Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : NULL0                           Flags        : D
```

# Run the **shutdown** command on GE 1/0/0 of Router B to simulate a link fault.

```
[~RouterB] interface GigabitEthernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] shutdown
```

# Check the IP routing table of Router A. The command output shows that default route 0::0/0 does not exist. This is because the default static route is bound to a BFD session. When BFD detects a link fault, BFD rapidly notifies that the bound static route becomes unavailable.

```
[~RouterA] display ipv6 routing-table
Routing Table : _public_
         Destinations : 1        Routes : 1

 Destination  : ::1                             PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
bfd
#
interface GigabitEthernet 1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1::1/64
#
ipv6 route-static :: 0 1::2 track bfd-session aa
#
bfd aa bind peer-ipv6 1::2
 discriminator local 10
 discriminator remote 20
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
bfd
#
```

```
interface GigabitEthernet 1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1::2/64
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 2::1/64
#
bfd bb bind peer-ipv6 1::1
 discriminator local 20
 discriminator remote 10
#
return
```

Related Tasks

2.8 Configuring Static BFD to detect IPv6 Static Routes

2.11.6 Example for Configuring FRR for IPv4 Static Routes on the Public Network

By configuring FRR for IPv4 static routes on the public network, you can fast detect link failures.

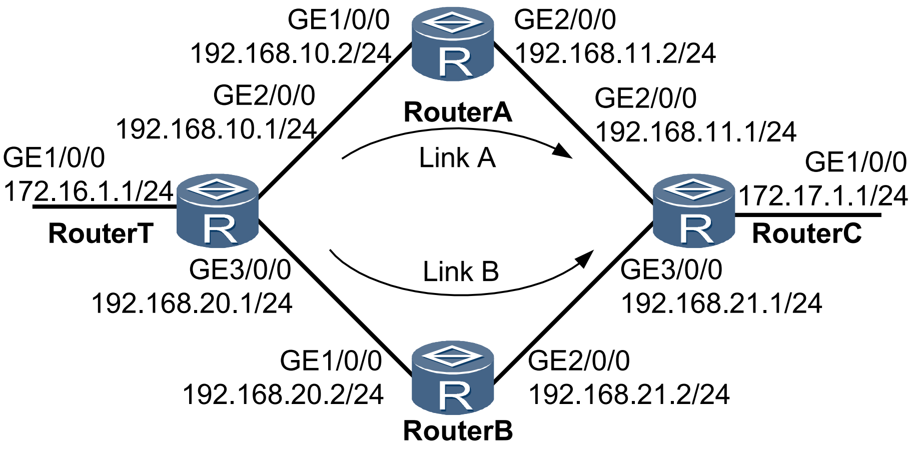
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-8**, it is required to configure two static routes with the next hops being Router A and Router B respectively on Router T to ensure that Link B functions as the backup of Link A. When Link A fails, traffic is rapidly switched to the backup link, namely, Link B.

Figure 2-8 Networking diagram of configuring FRR for IPv4 static routes on the public network



Configuration Notes

When configuring FRR for IPv4 static routes on the public network, pay attention to the following:

- Ensure that there are at least two static routes to the same destination address.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure two static routes with the next hops being Router A and Router B respectively on Router T.
2. On Router T, set a higher preference for Link A to ensure that Link A becomes the primary link.
3. Enable FRR for static route on Router T, and check the backup outbound interface and the backup next hop.
4. Disable FRR for static route, and check the backup outbound interface and the backup next hop.

Data Preparation

To complete the configuration, you need the following data:

- Preferences of static routes

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure static routes.

# On Router A, configure static routes.

```
[~RouterA] ip route-static 172.16.1.0 24 GigabitEthernet1/0/0 192.168.10.1
[~RouterA] ip route-static 172.17.1.0 24 GigabitEthernet2/0/0 192.168.11.1
[~RouterA] commit
```

# On Router B, configure static routes.

```
[~RouterB] ip route-static 172.16.1.0 24 GigabitEthernet1/0/0 192.168.20.1
[~RouterB] ip route-static 172.17.1.0 24 GigabitEthernet2/0/0 192.168.10.1
[~RouterB] commit
```

# On Router C, configure static routes.

```
[~RouterC] ip route-static 172.16.1.0 24 GigabitEthernet2/0/0 192.168.11.2
[~RouterC] ip route-static 172.16.1.0 24 GigabitEthernet3/0/0 192.168.21.2
[~RouterC] commit
```

# On Router T, configure static routes.

```
[~RouterT] ip route-static 172.17.1.1 24 GigabitEthernet2/0/0 192.168.10.2
[~RouterT] ip route-static 172.17.1.1 24 GigabitEthernet3/0/0 192.168.20.2
[~RouterT] commit
```

# Check the IP routing table of Router T, and you can view that load balancing is performed on the two static routes.

```
[~RouterT] display ip routing-table
```

```
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 10       Routes : 11

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  Direct 0    0             D  127.0.0.1       LoopBack0
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
     172.17.1.0/24  Static 60   0             D  192.168.10.2
GigabitEthernet2/0/0
                    Static 60   0             D  192.168.20.2
GigabitEthernet3/0/0
   192.168.10.0/24  Direct 0    0             D  192.168.10.1
GigabitEthernet2/0/0
   192.168.10.1/32  Direct 0    0             D  127.0.0.1
GigabitEthernet2/0/0
 192.168.10.255/32  Direct 0    0             D  127.0.0.1
GigabitEthernet2/0/0
   192.168.20.0/24  Direct 0    0             D  192.168.20.1
GigabitEthernet3/0/0
   192.168.20.1/32  Direct 0    0             D  127.0.0.1
GigabitEthernet3/0/0
 192.168.20.255/32  Direct 0    0            D  127.0.0.1      GigabitEthernet3/0/0
```

Step 3 Change the preferences of the IPv4 static routes.

# Change the preferences of static routes on Router T.

```
[~RouterT] ip route-static 172.17.1.1 24 GigabitEthernet2/0/0 192.168.10.2
preference 40
[~RouterT] commit
```

# Check the IP routing table of Router T, and you can view that the preferences of static routes are changed.

```
[~RouterT] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
         Destinations : 10       Routes : 10

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  Direct 0    0             D  127.0.0.1       LoopBack0
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
     172.17.1.0/24  Static 40   0             D  192.168.10.2
GigabitEthernet2/0/0
   192.168.10.0/24  Direct 0    0             D  192.168.10.1
GigabitEthernet2/0/0
   192.168.10.1/32  Direct 0    0             D  127.0.0.1
GigabitEthernet2/0/0
 192.168.10.255/32  Direct 0    0             D  127.0.0.1
GigabitEthernet2/0/0
   192.168.20.0/24  Direct 0    0             D  192.168.20.1
GigabitEthernet3/0/0
   192.168.20.1/32  Direct 0    0             D  127.0.0.1
GigabitEthernet3/0/0
 192.168.20.255/32  Direct 0    0            D  127.0.0.1      GigabitEthernet3/0/0
```

Step 4 Enable FRR for IPv4 static routes.

# Enable FRR for static route on Router T.

```
[~RouterT] ip route-static frr
[~RouterT] commit
```

# Check the backup outbound interface and the backup next hop on Router T.

```
<RouterT> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 172.17.1.0/24
     Protocol: Static          Process ID: 0
   Preference: 40                    Cost: 0
      NextHop: 192.168.10.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h00m03s
          Tag: 0                 Priority: medium
        Label: NULL               QoSInfo: 0x0
   IndirectID: 0x31000032
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet2/0/0
     TunnelID: 0x0                  Flags: D
    BkNextHop: 192.168.20.2   BkInterface: GigabitEthernet3/0/0
      BkLabel: NULL           SecTunnelID: 0x0
 BkPETunnelID: 0x0        BkPESecTunnelID: 0x0
 BkIndirectID: 0x32000033
```

Step 5 When Link A fails, traffic is rapidly switched to the backup link, namely, Link B.

```
[~RouterT] interface gigabitethernet 2/0/0
[~RouterT-GigabitEthernet2/0/0] shutdown
[~RouterT-GigabitEthernet2/0/0] commit
[~RouterT-GigabitEthernet2/0/0] quit
```

# Check the routes to the destination 172.17.1.0/24 on Router T.

```
<RouterT> display ip routing-table 172.17.1.0 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 172.17.1.0/24
     Protocol: Static          Process ID: 0
   Preference: 60                    Cost: 0
      NextHop: 192.168.20.2     Neighbour: 0.0.0.0
        State: Active Adv             Age: 00h00m07s
          Tag: 0                 Priority: medium
        Label: NULL               QoSInfo: 0x0
   IndirectID: 0x32000033
 RelayNextHop: 0.0.0.0          Interface: GigabitEthernet3/0/0
     TunnelID: 0x0                  Flags: D
```

----End

Configuration Files

- Configuration file of Router T

```
#
sysname RouterT
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.10.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 192.168.20.1 255.255.255.0
 ospf cost 100
#
ip route-static frr
ip route-static 172.17.1.0 24 GigabitEthernet2/0/0 192.168.10.2 preference 40
ip route-static 172.17.1.0 24 GigabitEthernet3/0/0 192.168.20.2
#
return
```

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.10.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.11.2 255.255.255.0
#
ip route-static 172.16.1.0 24 GigabitEthernet1/0/0 192.168.10.1
ip route-static 172.17.1.0 24 GigabitEthernet2/0/0 192.168.11.1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.20.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.21.2 255.255.255.0
#
ip route-static 172.16.1.0 24 GigabitEthernet1/0/0 192.168.20.1
ip route-static 172.17.1.0 24 GigabitEthernet2/0/0 192.168.10.1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.11.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 192.168.21.1 255.255.255.0
#
ip route-static 172.16.1.0 24 GigabitEthernet2/0/0 192.168.11.2
ip route-static 172.16.1.0 24 GigabitEthernet3/0/0 192.168.21.2
#
return
```

Related Tasks

2.9 Configuring FRR for IPv4 Static Routes

2.11.7 Example for Configuring IPv6 Static Routes

You can configure IPv6 static routes to interconnect any two devices on an IPv6 network.

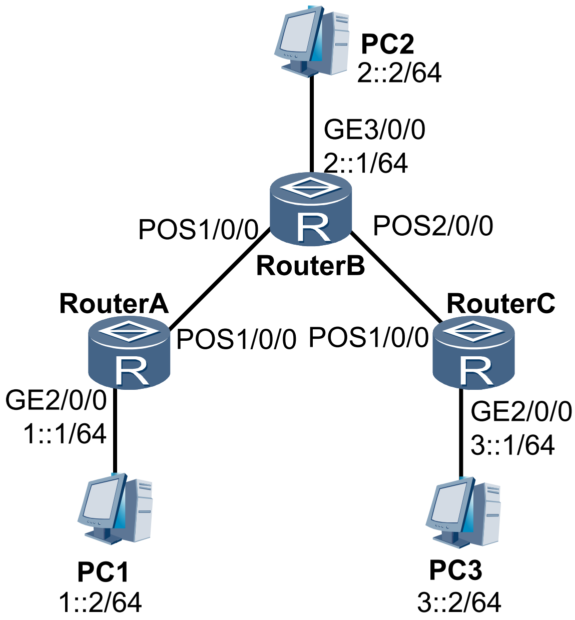
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 2-9**, the prefix length of all the IPv6 addresses is 64. It is required that IPv6 static routes be configured between routers to ensure that all hosts communicate with routers. POS interfaces on routers use IPv6 link-local addresses.

Figure 2-9 Networking diagram of configuring IPv6 static routes



Configuration Notes

When configuring an IPv6 static route, pay attention to the following:

- If the outbound interface is a broadcast interface, specify the next-hop address.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IPv6 addresses for GE interfaces on each router for interworking.
2. Configure the IPv6 static route and the default route to the destination address on each router.
3. Configure the IPv6 default gateway on each host to make any two hosts communicate.

Data Preparation

To complete the configuration, you need the following data:

- Default route with the outbound interface being POS 1/0/0 on Router A
- Static route with the destination address being 1:: 64 and outbound interface being POS 1/0/0 on Router B
- Static route with the destination address being 3:: 64 and outbound interface being POS 2/0/0 on Router B
- Default route with the outbound interface being POS 1/0/0 on Router C
- Default gateways of PC1, PC2, and PC3

Procedure

Step 1 Configure an IPv6 address for each interface. The configuration details are not mentioned here.

Step 2 Configure IPv6 static routes.

# Configure an IPv6 default route on Router A.

```
[~RouterA] ipv6 route-static :: 0 pos 1/0/0
[~RouterA] commit
```

# Configure two IPv6 static routes on Router B.

```
[~RouterB] ipv6 route-static 1:: 64 pos 1/0/0
[~RouterB] ipv6 route-static 3:: 64 pos 2/0/0
[~RouterB] commit
```

# Configure an IPv6 default route on Router C.

```
[~RouterC] ipv6 route-static :: 0 pos 1/0/0
[~RouterC] commit
```

Step 3 Configure host addresses and gateways.

Configure IPv6 addresses for hosts according to the networking diagram, and then configure the default gateway of PC1 as 1::1, default gateway of PC2 as 2::1, and default gateway of PC3 as 3::1.

Step 4 Verify the configuration.

# Check the IPv6 routing table of Router A.

```
[~RouterA] display ipv6 routing-table
Routing Table : _public_
         Destinations : 5        Routes : 5

 Destination  : ::                              PrefixLength : 0
 NextHop      : ::                              Preference   : 60
 Cost         : 0                               Protocol     : Static
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : Pos1/0/0                        Flags        : D

 Destination  : ::1                             PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : 1::                             PrefixLength : 64
 NextHop      : 1::1                            Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : GigabitEthernet2/0/0            Flags        : D

 Destination  : 1::1                            PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : FE80::                          PrefixLength : 10
 NextHop      : ::                              Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : NULL0                           Flags        : D
```

# Run the **ping** command to verify the connectivity.

```
[~RouterA] ping ipv6 3::1
  PING 3::1 : 56  data bytes, press CTRL_C to break
    Reply from 3::1
    bytes=56 Sequence=1 hop limit=254  time = 63 ms
    Reply from 3::1
    bytes=56 Sequence=2 hop limit=254  time = 62 ms
    Reply from 3::1
    bytes=56 Sequence=3 hop limit=254  time = 62 ms
    Reply from 3::1
    bytes=56 Sequence=4 hop limit=254  time = 63 ms
    Reply from 3::1
    bytes=56 Sequence=5 hop limit=254  time = 63 ms
  --- 3::1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 62/62/63 ms
```

Run the **tracert** command to verify the connectivity.

```
[~RouterA] tracert ipv6 3::1
 traceroute to 3::1  30 hops max,60 bytes packet
 1.  :: 11 ms  3 ms  4 ms
 2.  3::1 4 ms  3 ms  3 ms
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1::1/64
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
#
ipv6 route-static :: 0 Pos 1/0/0
#
return
```

- Configuration file of Router B

```
#
```

```
sysname RouterB
#
ipv6
#
interface GigabitEthernet3/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2::1/64
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
#
ipv6 route-static 1:: 64 Pos1/0/0
ipv6 route-static 3:: 64 Pos2/0/0
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
ipv6
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 3::1/64
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
#
ipv6 route-static :: 0 Pos1/0/0
#
return
```

Related Tasks

2.4 Configuring IPv6 Static Routes

3 OSPF Configuration

About This Chapter

By building OSPF networks, you can enable OSPF to discover and calculate routes in ASs. OSPF is applicable to a large-scale network that consists of hundreds of routers.

This section describes how to control OSPF routing information. Detailed operations include setting priorities for protocols, importing external routes, and filtering the received routes and LSAs.

3.9 Configuring OSPF IP FRR
In the case of a link fault, a device enabled with OSPF IP FRR can fast switch traffic to the backup link. This protects traffic and greatly improves the reliability of OSPF networks.

3.10 Configuring BFD for OSPF
If there are high requirements for data transmission, and OSPF convergence needs to be speeded up when the link status changes, you can configure BFD on OSPF links. After detecting a link failure, BFD notifies the routing protocol of the failure, which triggers fast convergence. When the neighbor relationship is Down, the BFD session is deleted dynamically.

3.11 Configuring OSPF Fast Convergence
By adjusting OSPF timers, you can implement OSPF fast network convergence.

3.12 Configuring OSPF GR Helper
To avoid traffic interruption and route flapping caused by the active/standby switchover, you can enable OSPF GR.

3.13 Improving the Stability of an OSPF Network
A stable OSPF network features less route flapping, normal router performance, and good network performance.

3.14 Improving the Security of an OSPF Network
On a network demanding high security, you can configure OSPF authentication and the GTSM to improve the security of the OSPF network.

3.15 Configuring the Network Management Function of OSPF
OSPF supports the network management function. You can bind the OSPF MIB to a certain OSPF process, and configure the trap function and log function.

3.16 Maintaining OSPF
Maintaining OSPF involves resetting OSPF, clearing OSPF statistics, and debugging OSPF.

3.17 Configuring Examples
This section provides several configuration examples of OSPF together with the configuration flowchart. The configuration examples explain networking requirements, configuration notes, and configuration roadmap.

3.1 OSPF Overview

OSPF, which is developed by the IETF, is a link-state IGP. OSPF is widely used in access networks and MANs.

Before the emergence of OSPF, the Routing Information Protocol (RIP) is widely used on networks as an Interior Gateway Protocol (IGP).

RIP is a routing protocol based on the distance vector algorithm. Due to its slow convergence, routing loops, and poor scalability, RIP is gradually replaced by OSPF.

As a link-state protocol, OSPF can solve many problems encountered by RIP. Additionally, OSPF has the following advantages:

- Supports area partition. An Autonomous System (AS) can be partitioned into areas to simplify management. The Link State Database (LSDB) of a device in an area needs to be consistent with only the LSDBs of other devices in this area. The decrease in the size of the LSDB greatly reduces the memory consumption and CPU usage of the device. In addition, less network bandwidth is consumed because of the decrease in routing information to be transmitted between areas.
- Receives or sends packets in multicast mode to reduce load on the router that does not run OSPF.
- Supports Classless Interdomain Routing (CIDR).
- Supports load balancing among equal-cost routes.
- Supports packet authentication.

With the preceding advantages, OSPF is widely accepted and used as an IGP.

NOTE

In this chapter, OSPF refers to OSPF Version 2 (OSPFv2), unless otherwise specified.

Typical Networking of OSPF

As shown in **Figure 3-1**, there are two most important concepts in an OSPF network, namely, areas and different types of device.

The number of devices increases with the increasing expansion of the network scale. This leads to a large LSDB on each router, which imposes a heavy burden on the router. OSPF solves this problem by partitioning an AS into different areas. An area is regarded as a logical group, which is identified by an area ID. At the border of an area resides a device rather than a link. A network segment (or a link) belongs to only one area. That is, the area to which each OSPF interface belongs needs to be specified.

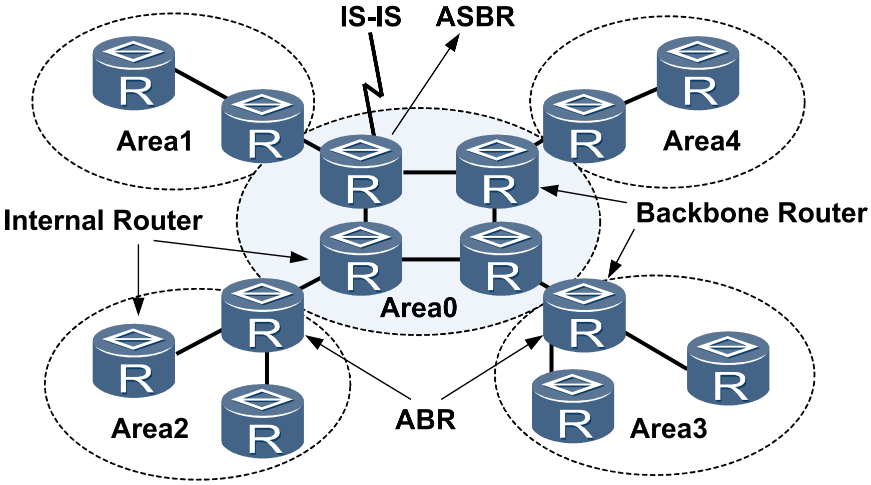
After area partition, route aggregation can be performed on the Area Border Router (ABR) to reduce the number of Link State Advertisements (LSAs) to be advertised to other areas. Route aggregation also minimizes the impacts caused by changes in the topology.

Based on their locations in an AS, the devices that run OSPF are classified into the following types:

- Internal routers
- ABRs

- Backbone routers
- AS Boundary Routers (ASBRs)

Figure 3-1 Typical networking diagram of OSPF



In an AS, inter-area routes and intra-area routes describe the network structure of the AS. AS external routes describe how to select a route to a destination outside an AS. OSPF classifies the imported AS external routes into Type 1 and Type 2 external routes.

Table 3-1 lists route types in descending order of priority.

Table 3-1 OSPF route type

Route	Description
Intra area	Indicates intra-area routes.
Inter area	Indicates inter-area routes.
Type1 external	Indicates Type 1 external routes. The cost of a Type 1 external route equals the cost for the OSPF device to reach an ASBR plus the cost of the route from the ASBR to the destination. When the cost of an external route approximately equals the cost of an AS internal route, this external route is considered highly reliable and can be configured as a Type 1 external route.

Route	Description
Type2 external	Indicates Type 2 external routes. The cost of a Type 2 external route equals the cost of the route from an ASBR to the destination. Therefore, during route calculation, OSPF considers only the cost of the route from an ASBR to the destination outside an AS, namely, the cost of a Type 2 external route. When the cost of the route from an ASBR to the destination outside an AS is much greater than the cost of the internal route to the ASBR, this external route has a low reliability and can be configured as a Type 2 external route.

OSPF Network Type

OSPF classifies networks into the following types according to the types of link layer protocol:

- Broadcast networks

  If the link layer protocol is Ethernet or Fiber Distributed Data Interface (FDDI), OSPF defaults the network type to broadcast.

  - Hello packets and packets from the Designated Router (DR) are sent in multicast mode by using address 224.0.0.5, which indicates the reserved IP multicast address for OSPF devices.
  - Link State Update (LSU) packets are sent to the DR in multicast mode by using address 224.0.0.6, which indicates the reserved IP multicast address for the OSPF DR. Then, the DR forwards the LSU packets to destination 224.0.0.5.
  - Database Description (DD) packets, Link State Request (LSR) packets, and all retransmission packets are sent in unicast mode.
  - Link State Acknowledgment (LSAck) packets are usually sent in multicast mode by using address (224.0.0.5). When a device receives repeated LSAs, or the LSAs are deleted due to the timeout of the maximum lifetime, LSAck packets are sent in unicast mode.

- Non-Broadcast Multiple Access (NBMA) networks

  If the link layer protocol is frame relay (FR), ATM, or X.25, OSPF defaults the network type to NBMA. In this type of network, protocol packets, such as Hello packets, DD packets, LSR packets, LSU packets, and LSAck packets, are sent in unicast mode.

- Point-to-Multipoint (P2MP) networks

  There is no concept of P2MP in link layer protocols. Therefore, a P2MP network must be forcibly changed from other network types. In this type of network, Hello packets are sent in multicast mode by using address 224.0.0.5; DD packets, LSR packets, LSU packets. and LSAck packets are sent in unicast mode.

- Point-to-point (P2P) networks

  If the link layer protocol is PPP, High-Level Data Link Control (HDLC), or Link Access Procedure, Balanced (LAPB), OSPF defaults the network type to P2P. In this type of network, protocol packets, such as Hello packets, DD packets, LSR packets, LSU packets, and LSAck packets, are sent in multicast mode by using address 224.0.0.5.

3.2 OSPF Features Supported by the NE5000E

The supported OSPF features include the OSPF multi-process, authentication, Non-stop Routing (NSR), Bidirectional Forwarding Detection (BFD), Traffic Engineering (TE), DiffServ-aware TE (DS-TE), IGP shortcut, forwarding adjacency, OSPF multi-instance, smart-discovery, Generalized TTL Security Mechanism (GTSM), fast convergence, and host routes advertisement.

Multi-process

OSPF supports multi-process. Multiple different OSPF processes can run on the same router, and are independent of each other. Route interaction between different OSPF processes is similar to route interaction between different routing protocols.

An interface of the router belongs to only a certain OSPF process.

Authentication

OSPF supports packet authentication. Only the OSPF packets that pass the authentication can be received. If packets fail to pass the authentication, the neighbor relationship cannot be established. The NE5000E supports the following authentication modes:

- Area authentication
- Interface authentication

When both area authentication and interface authentication are available, interface authentication is preferred.

OSPF NSR

Non-Stop Routing (NSR) is a routing technique that prevents a neighbor from sensing the fault on the control plane of a router that provides a slave control plane. With NSR, when the control plane of the router becomes faulty, the neighbor relationship set up through specific routing protocols, MPLS, and other protocols that carry services are not interrupted.

As networks develop fast, operators pose high requirements for reliability on IP networks. NSR, as a high availability (HA) solution, is thus introduced to ensure that services transmitted by a device are not affected when a hardware or software failure occurs on the device.

OSPF IP FRR

OSPF IP FRR pre-computes a backup link by using the Loop-Free Alternate (LFA) algorithm, and then adds the backup link and the primary link to the forwarding table. In the case of failures, OSPF IP FRR can fast switch traffic to the backup link before routes on the control plane converge. This prevents traffic interruption and thus protects traffic and improves reliability of an OSPF network.

OSPF IP FRR complies with RFC 5286, that is, Basic Specification for IP Fast Reroute Loop-Free Alternates, which protects traffic when links or nodes become faulty.

BFD for OSPF

By default, on broadcast networks, the interval for OSPF to send Hello packets is 10 seconds; on NBMA networks, the interval for sending Hello packets is 30 seconds. The interval for declaring a neighbor Down, that is, the dead time after which the neighbor relationship becomes invalid, is four times the interval for sending Hello packets. If the router does not receive a Hello packet from its neighbor within the dead time, the router deletes the neighbor. That is, the router detects the neighbor faults in seconds. This causes a large number of packets to be lost on a high-speed network.

Bidirectional Forwarding Detection (BFD) is introduced to solve the preceding problem in the existing detection mechanism. BFD ensures the detection interval in milliseconds. Instead of replacing the Hello mechanism of OSPF, BFD works with OSPF to fast detect the adjacency fault. In addition, BFD instructs OSPF to recalculate corresponding routes for correct packet forwarding.

OSPF supports the dynamic establishment or deletion of BFD sessions on broadcast, P2P, P2MP, or NBMA links.

Smart-discover

Generally, the router periodically sends Hello packets through OSPF interfaces. By exchanging Hello packets, the routers establish and maintain the neighbor relationship, and elect the DR and the Backup Designated Router (BDR) on the multi-access network (broadcast or NBMA network). During the establishment of the neighbor relationship or the election of the DR and the BDR on the multi-access network, interfaces send Hello packets only when the Hello timer expires. This affects the speed of establishing the neighbor relationship or electing the DR and the BDR.

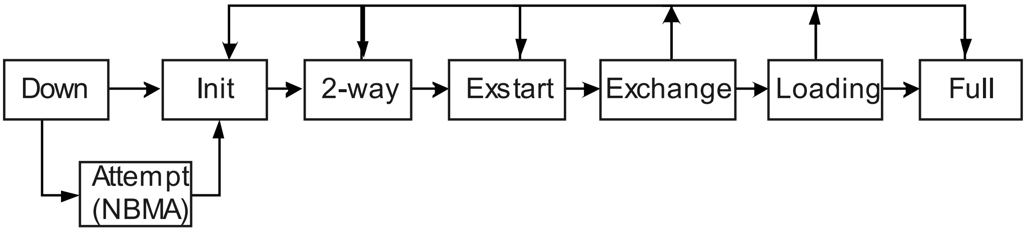
NOTE

- The interval for an interface to send Hello packets depends on the configured interval for sending Hello packets on the interface.
- The default value of the interval for sending Hello packets varies with the network type.

Configuring Smart-discover can solve the preceding problem.

- On a broadcast or NBMA network, the neighbor relationship can be established rapidly, and a DR and a BDR on the network can be elected rapidly.
  - When the neighbor status becomes 2-way for the first time or returns to Init from the 2-way or higher state shown in **Figure 3-2**, the interface enabled with Smart-discover sends Hello packets to a neighbor without waiting for the timeout of the Hello timer when detecting that the neighbor status changes.

Figure 3-2 Changes of the neighbor state machine



  - When the interface status of the DR or the BDR on the multi-access network changes, the interface enabled with Smart-discover sends Hello packets to the network segment and then participates in the DR or BDR election.

- On a P2P or P2MP network, the adjacency can be established rapidly. The principle of establishing adjacencies on a P2P and P2MP network is the same as that on a broadcast or NBMA network.

OSPF TE and DS-TE

OSPF Traffic Engineering (TE) supports the establishment and maintenance of Label Switched Path (LSP) of TE.

When constructing a constraint-based routed (CR) LSP, MPLS needs information about the traffic attributes of all links in the local area. MPLS obtains TE information of the links through OSPF.

OSPF supports a new type of LSA, which is called opaque LSA. The opaque LSA is used to carry TE information. You can run commands to determine whether OSPF can generate or process the opaque LSA that carries TE information.

The HUAWEI NetEngine5000E supports OSPF TE and DiffServ-aware TE (DS-TE). They can ensure high utilization of the network and provide differentiated services for the data flows with different priorities, thus providing the bandwidth guaranteed services of low delay and low packet loss for the voice and video data flows. It is difficult to deploy TE on the entire network. Therefore, in the actual networking, the DiffServ model is often used to implement Quality of Service (QoS).

To support DS-TE in MPLS, OSPF supports local overbooking multiplier type/length/value (TLV) and bandwidth constraint (BC) TLV.

NOTE

For detailed configuration of OSPF TE, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - MPLS*.

IGP Shortcut and Forwarding Adjacency

OSPF supports IGP shortcut and forwarding adjacency. The two features allow OSPF to use an LSP as an outgoing interface to reach a destination. Without the two features, OSPF cannot use the LSP as an outgoing interface even if the LSP to the destination exists.

Differences between IGP shortcut and forwarding adjacency are as follows:

- If only forwarding adjacency is enabled, OSPF can reach the destination by using the LSP.
- If only IGP shortcut is enabled, only the router enabled with IGP shortcut can use the LSP.

NOTE

For detailed configurations of IGP shortcut and forwarding adjacency, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - MPLS*.

OSPF VPN Multi-instance

OSPF supports multi-instance, which can run between Provider Edges (PEs) and Customer Edges (CEs) on VPNs.

On a BGP MPLS VPN, many sites of one VPN can use OSPF as the internal routing protocol. The sites, however, are handled as being from different ASs. In this manner, the OSPF routes learned on one site are transmitted as external routes to another site. This results in heavy OSPF traffic and some originally avoidable problems of network management.

In the implementation of the NE5000E, you can configure domain IDs on PEs to differentiate the VPNs where different sites reside. Different sites in one VPN consider that they are connected directly. In this case, PEs exchange OSPF routing information as if they were directly connected through a leased line. This improves network management and effectively uses OSPF.

📖 **NOTE**

For detailed configuration of OSPF VPN multi-instance, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN*.

Synchronization Between OSPF and LDP

On a network with primary and backup links, when the primary link becomes faulty, traffic is switched from the primary link to the backup link. In this process, traffic is interrupted in a short time. After the primary link recovers, traffic is switched back from the backup link to the primary link. In this process, traffic is interrupted in a comparatively long time.

Configuring synchronization between OSPF and LDP can ensure millisecond-level traffic interruption when traffic is switched back from the backup link to the primary link.

The principle of synchronization between OSPF and LDP is to delay route switchback by suppressing the establishment of the OSPF neighbor relationship until LDP convergence is complete. That is, before an LSP is established on the primary link, the backup link continues forwarding traffic. The backup link is deleted after the LSP is established on the primary link.

GTSM

The Generalized TTL Security Mechanism (GTSM) protects services above the IP layer against attacks by checking whether the Time-to-Live (TTL) value in the IP header is within a specified range. In applications, the GTSM is mainly used to protect the TCP/IP-based control plane, including routing protocols, against attacks of the CPU-utilization type, such as CPU overload.

📖 **NOTE**

For detailed configuration of OSPF GTSM, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - Security*.

OSPF Fast Convergence

OSPF fast convergence is an extended feature of OSPF implemented to speed up route convergence, which has the following functions:

- Supports OSPF Smart-discover.
- Supports partial route calculation (PRC).
- Controls the generation and receiving of LSAs through the intelligent timer.
- Controls route calculation through the intelligent timer.
- Fast convergence by priority of routes.

Advertising Host Routes

OSPF can advertise host routes, that is, advertise IP addresses of interfaces as host addresses. This ensures that host addresses are reachable on optical networks.

3.3 Configuring Basic OSPF Functions

Before building OSPF networks, you need to configure basic OSPF functions.

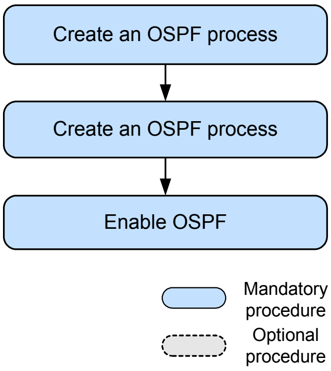
Pre-configuration Tasks

Before configuring basic OSPF functions, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer

Configuration Procedures

Figure 3-3 Flowchart of configuring basic OSPF functions



Related Tasks

3.17.1 Example for Configuring Basic OSPF Functions

3.3.1 Creating an OSPF Process

Creating an OSPF process is a prerequisite for configuring all OSPF features. When creating an OSPF process, you can manually specify the router ID for the router.

Context

To run OSPF, the router needs to have a router ID. A router ID of the router is a 32-bit unsigned integer, which uniquely identifies the router in an AS. To ensure the stability of OSPF, you need to manually configure a router ID for each device during network planning.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id | router-id router-id | vpn-instance vpn-instance-name ] *
```

An OSPF process is created, and the OSPF view is displayed.

- The parameter *process-id* specifies the ID of an OSPF process. The default value is 1.

  The NE5000E supports OSPF multi-process. You can create different processes for different types of service. The OSPF process ID is valid in the local area, without affecting packet exchange with other routers. Therefore, different routers can also exchange packets even though they have different process IDs.
- The parameter **router-id** *router-id* specifies the router ID of the router.

  By default, the system automatically selects an IP address of the interface as the router ID. When manually setting a router ID, ensure that the router ID of each device in an AS is unique. Generally, you can set the router ID to be the same as the IP address of a certain interface on the device.

  NOTE

  The router ID of each OSPF process must be unique on the entire network; otherwise, the OSPF neighbor relationship cannot be set up and routing information is incorrect. Configuring a unique router ID for each OSPF process on each OSPF device is recommended.
- The parameter **vpn-instance** *vpn-instance-name* specifies the name of a VPN instance.

  If a VPN instance is specified, the OSPF process belongs to the specified VPN instance. Otherwise, the OSPF process belongs to the public network instances.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

3.3.2 Creating an OSPF Area

OSPF packets are exchanged between devices in an OSPF area to synchronize routing information.

Context

More and more routers are deployed with the increasing expansion of the network scale. As a result, each router has to maintain a large LSDB, which becomes a heavy burden. OSPF solves this problem by dividing an AS into areas. An area is regarded as a logical router group. Each group is identified by an area ID. The borders of an area are routers, rather than links. A network segment (or a link) belongs to only one area. That is, each OSPF interface must belong to an area.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id | router-id router-id | vpn-instance vpn-instance-name ] *
```

The OSPF process is enabled, and the OSPF view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

Areas are not equally important. The area with the area ID being 0 is called the backbone area. The backbone area is responsible for forwarding inter-area routing information. In addition, routing information between non-backbone areas must be forwarded through the backbone area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.3.3 Enable OSPF

You can configure network segments belonging to an area on each OSPF-aware device so that OSPF routes can be discovered and then calculated in each AS.

Context

After creating an OSPF process, you need to configure the network segments included in an area. A network segment belongs to only one area. That is, you need to specify an area for each interface that runs OSPF. In this document, the network segment refers to the network segment to which the IP address of the OSPF interface belongs.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

OSPF can be enabled in an OSPF area or on a specific interface.

- Enable OSPF in an OSPF area.

1. Run:

   ```
   network ip-address wildcard-mask
   ```

   Network segments belonging to an area are configured.

   OSPF can properly run on an interface only when the following conditions are met:

   - The IP address mask length of the interface is equal to or greater than the mask length specified in the **network** command.
   - The primary IP address of the interface must be within the network segment specified by the **network** command.

By default, OSPF advertises the IP address of the loopback interface as a 32-bit host route, which is irrelevant to the mask length configured on the loopback interface. To advertise routes to the network segment of the loopback interface, configure the network type as NBMA or broadcast in the interface view. For details, see **Configuring Network Types of OSPF Interfaces**.

- Enable OSPF on an interface.

1. Run:

   ```
   quit
   ```

   Return to the OSPF process view.

2. Run:

   ```
   quit
   ```

   Return to the system view.

3. Run:

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

   The interface view is displayed.

4. Run:

   ```
   ospf enable process-id area area-id
   ```

   OSPF is enabled on the interface.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.3.4 Checking the Configuration

After basic OSPF functions are configured, you can check OSPF neighbors in each area, interfaces, and routing information.

Prerequisite

All configurations of basic OSPF functions are complete.

Procedure

- Run the **display ospf** [*process-id*] **peer** command in any view to check information about OSPF neighbors.
- Run the **display ospf** [*process-id*] **interface** command in any view to check information about OSPF interfaces.
- Run the **display ospf** [*process-id*] **routing** command in any view to check information about the OSPF routing table.

----End

Example

Run the **display ospf peer** command, and you can view the Router IDs, addresses, and status of OSPF neighbors. For example:

```
<HUAWEI> display ospf peer
          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 192.168.1.1 ( GE3/0/2 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: Full       Mode:Nbr is  Slave    Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3     MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 0
   Neighbor is up for 00:00:03
   Authentication Sequence: [ 0 ]
```

Run the **display ospf interface** command, and you can view the network types and status of OSPF interfaces. For example:

```
<HUAWEI> display ospf interface

          OSPF Process 1 with Router ID 192.168.1.1
                  Interfaces
 Area: 0.0.0.0          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/1/0           192.168.1.1    P2P     P-2-P   1      100

 Area: 0.0.0.1          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/0/0           172.16.1.1   Broadcast    DR   1      50
```

Run the **display ospf routing** command, and you can view the destination addresses, link costs, and next hops of OSPF routes. For example:

```
<HUAWEI> display ospf routing

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.16.1.0/24      2          Transit    192.168.1.2     3.3.3.3         0.0.0.1
 172.17.1.0/24      3          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0
 192.168.2.0/24     2          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0

 Total Nets: 3
 Intra Area: 1  Inter Area: 2  ASE: 0  NSSA: 0
```

3.4 Configuring Session Parameters for the OSPF Neighbor or Adjacency Relationship

On an OSPF network, all routing information is transmitted and exchanged between neighboring or adjacent routers. By maintaining neighbor relationships or adjacencies, you can stabilize the entire network.

Pre-configuration Tasks

Before configuring session parameters for the OSPF neighbor or adjacency relationship, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer

- Configuring Basic OSPF Functions

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

3.4.1 Setting the OSPF Packet Retransmission Limit

By limiting the number of DD packet retransmissions, Update packet retransmissions, or Request packet retransmissions, you can close the neighbor relationship when the number of packet retransmissions reaches the specified value.

Context

After an OSPF router sends one of the following packets, if it does not receive the LSAck packet within a specified time, it retransmits the packet. After the number of packet retransmissions reaches the set limit, the OSPF router tears down the adjacency relationship with its neighbor.

- DD packets
- Update packets
- Request packets

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF interface view is displayed.

Step 3 Run:

```
retransmission-limit [ max-number ]
```

The OSPF packet retransmission limit is set.

By default, the OSPF packet retransmission limit is not set. The default maximum number of packet retransmissions is 30.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.4.2 Configuring an Interface to Fill in the DD Packet with the Actual MTU

You can enable the adding of the actual MTU of an interface to a DD packet. In this manner, the actual MTU of the interface is added to the Interface MTU field in the DD packet, and the interface checks whether the MTU in the DD packet sent from the neighbor is the same as the local MTU. If the two MTU values are different, the neighbor relationship cannot be established.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf mtu-enable
```

The interface is configured to fill in the DD packet with the actual MTU and check whether the MTU in the DD packet from the neighbor exceeds the MTU of the local end.

By default, the MTU in the DD packet sent by an interface is 0.

CAUTION

Setting the MTU in a DD packet will lead to the reestablishment of the neighbor relationship.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.4.3 Checking the Configuration

After setting the session parameters for OSPF neighbors or OSPF adjacencies, you can view the OSPF neighbor list, brief information, and retransmission list.

Prerequisite

All configurations of session parameters of the OSPF neighbor or adjacency relationship are complete.

Procedure

- Run the **display ospf** [*process-id*] **peer** command to check information about OSPF neighbors.
- Run the **display ospf** [*process-id*] **brief** command to check brief information about the specified OSPF process.
- Run the **display ospf** [*process-id*] **retrans-queue** [*interface-type interface-number*] [*neighbor-id*] command to check the OSPF retransmission list.

----End

Example

Run the **display ospf peer** command, and you can view the router IDs, addresses, and status of OSPF neighbors. For example:

```
<HUAWEI> display ospf peer
          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 192.168.1.1 ( GE3/0/2 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: Full       Mode:Nbr is  Slave   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3     MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 0
   Neighbor is up for 00:00:05
   Authentication Sequence: [ 0 ]
```

Run the **display ospf brief** command, and you can view various OSPF timers and the number of packet retransmissions. For example:

```
<HUAWEI> display ospf brief

          OSPF Process 1 with Router ID 20.1.1.1
                  OSPF Protocol Information

 RouterID: 20.1.1.1          Border Router:
 Multi-VPN-Instance is not enabled
 Global DS-TE Mode: Non-Standard IETF Mode
 Graceful-restart capability: disabled
 Helper support capability  : not configured
 Applications Supported: MPLS Traffic-Engineering
 Spf-schedule-interval: max 10000 ms, start 500 ms, hold 1000 ms
 Default ASE parameters: Metric: 1 Tag: 1 Type: 2
 Route Preference: 10
 ASE Route Preference: 150
 SPF Computation Count: 4
 RFC 1583 Compatible
 Retransmission limitation is disabled
 Area Count: 1   Nssa Area Count: 0
 ExChange/Loading Neighbors: 0


 Area: 0.0.0.0          (MPLS TE not enabled)
 Authtype: None   Area flag: Normal
 SPF scheduled Count: 4
 ExChange/Loading Neighbors: 0

 Interface: 66.1.1.1 ( GE1/0/0 )
 Cost: 1       State: BDR         Type: Broadcast           MTU: 1500
 Priority: 1
 Designated Router: 66.1.1.2
 Backup Designated Router: 66.1.1.1
 Timers: Hello 10 , Dead 40 , Wait 40 , Poll 120 , Retransmit 5 , Transmit Delay 1
```

Run the **display ospf retrans-queue** command, and you can view the OSPF retransmission list. For example:

```
<HUAWEI> display ospf request-queue
          OSPF Process 1 with Router ID 1.1.1.1
                  OSPF Request List
  The Router's Neighbor is Router ID 4.4.4.4         Address 172.1.4.2
  Interface 172.1.4.1        Area 0.0.0.2
  Request list:
       Type       LinkState ID      AdvRouter         Sequence   Age
       Router     1.1.1.1           1.1.1.1           8000001b   677
```

3.5 Configuring OSPF Attributes in Different Types of Networks

By setting network types for OSPF interfaces and adjusting OSPF attributes, you can build OSPF networks flexibly.

Applicable Environment

According to the types of link layer protocols, OSPF classifies networks into the following types:

- P2MP: There is no concept of P2MP in link layer protocols. Therefore, a P2MP network must be forcibly changed from other network types.
- NBMA: If the link layer protocol is FR, ATM, or X.25, OSPF defaults the network type to NBMA.
- Broadcast: If the link layer protocol is GigabitEthernet or FDDI, OSPF defaults the network type to broadcast.
- P2P: If the link layer protocol is PPP, HDLC, or LAPB, OSPF defaults the network type to P2P.

When link layer protocols remain unchanged, you can change network types and configure OSPF features to flexibly build networks.

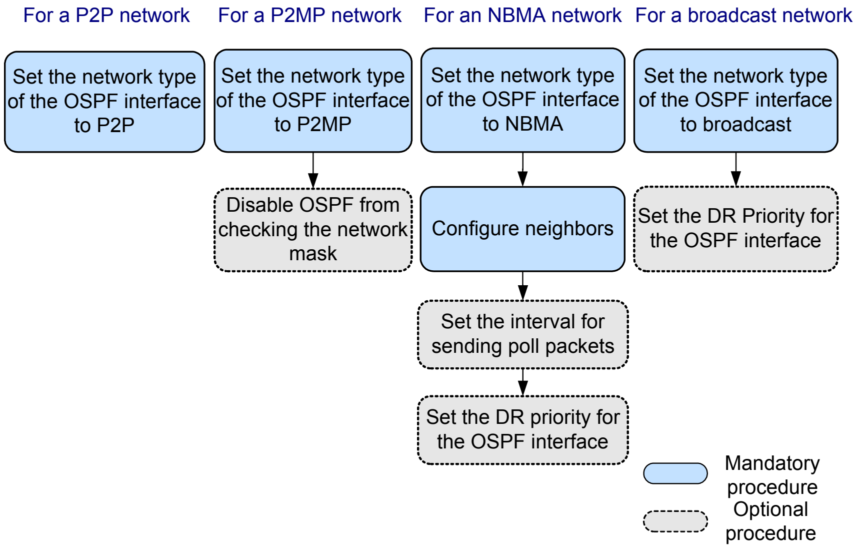
Pre-configuration Tasks

Before configuring OSPF attributes in different types of networks, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

Figure 3-4 Flowchart of configuring OSPF attributes in different types of networks


For a P2P network
For a P2MP network
For an NBMA network
For a broadcast network
Set the network type of the OSPF interface to P2P
Set the network type of the OSPF interface to P2MP
Set the network type of the OSPF interface to NBMA
Set the network type of the OSPF interface to broadcast
Disable OSPF from checking the network mask
Configure neighbors
Set the DR Priority for the OSPF interface
Set the interval for sending poll packets
Set the DR priority for the OSPF interface
Mandatory procedure
Optional procedure

3.5.1 Configuring the Network Type for an OSPF Interface

OSPF classifies networks into the broadcast network, P2P network, P2MP network, and NBMA network according to link layer protocols. By configuring network types for interfaces, you can change the network types of interfaces.

Context

You can configure one of the following network types for an interface as required:

- P2MP: There is no concept of P2MP in link layer protocols. Therefore, a P2MP network must be forcibly changed from other network types.
- NBMA: An NBMA network must be fully meshed. That is, any two routers on the NBMA network must be directly reachable. In most cases, however, this requirement cannot be met. In this case, you need to forcibly change the network type through commands.
- Broadcast: To speed up the establishment of the neighbor relationship, you can change the network type of broadcast to P2P network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf network-type { broadcast | nbma | p2mp | p2p }
```

The network type of the OSPF interface is configured.

By default, the network type of an interface depends on the physical interface. The network type of an Ethernet interface is broadcast; the network type of a serial or POS interface (encapsulated with PPP or HDLC) is P2P; the network type of an ATM or FR interface is NBMA.

Configuring the new network type for an interface will cause the OSPF session on the interface to be reestablished.

NOTE

Generally, the network types of OSPF interfaces on both ends of a link must be the same. Otherwise, routes cannot be correctly calculated.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.5.2 (Optional) Setting the DR Priority for the OSPF Interface of the Broadcast or NBMA Network Type

When configuring a broadcast network or an NBMA network, you can specify the DR priority for each interface to change the results of DR/BDR election on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf dr-priority priority
```

The DR priority of the OSPF interface is set. The greater the value, the higher the priority.

By default, the DR priority of an interface is 1.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

⚠ **CAUTION**

Restarting or shutting down the current interface will interrupt the OSPF adjacency relationship between devices. Therefore, perform the operation with caution.

Reconfiguring the DR priority for a device does not change the DR or BDR on a network. You can reelect a DR or BDR by using the following methods. This, however, will result in the interruption of the OSPF adjacency relationship between devices. Therefore, the following methods are used only when necessary.

- Restart the OSPF processes on all the routers.
- Run the **shutdown** and then **undo shutdown** commands on the interfaces where the OSPF adjacency relationship is established.

3.5.3 (Optional) Disabling OSPF from Checking the Network Mask on a P2MP Network

On a P2MP network, when the mask lengths of devices are different, you can disable devices from checking the network mask so that the OSPF neighbor relationship can be established.

Context

OSPF needs to check the network mask in the received Hello packet. When receiving a Hello packet that carries a different network mask from that of the local device, OSPF discards the Hello packet. To establish the OSPF neighbor relationship on a P2MP network, you need to disable OSPF from checking the network mask.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf network-type p2mp
```

The network type of the OSPF interface is configured as P2MP.

Step 4 Run:

```
ospf p2mp-mask-ignore
```

OSPF is disabled from checking the network mask on the P2MP network.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

3.5.4 Configuring Neighbors on the NBMA Network

An NBMA interface cannot find neighbors by broadcasting Hello packets. Thus, you need to manually specify the IP addresses of neighbors for this interface and set the election rights for these neighbors.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
peer ip-address [ dr-priority priority ]
```

Neighbors are configured on an NBMA network.

- The parameter *ip-address* specifies the IP address of a neighbor.
- The parameter **dr-priority** *priority* specifies the DR priority of the neighbor. The greater the value, the higher the priority.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.5.5 (Optional) Setting the Interval for Sending Hello Packets on the NBMA Network

On an NBMA network, a device sends a Hello packet to a Down neighbor at a poll interval.

Context

On an NBMA network, devices establish neighbor relationships with adjacencies by sending Hello packets.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf timer poll interval
```

The interval for sending Poll packets on the NBMA interface is set.

The parameter *interval* specifies the polling interval for sending Hello packets.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.5.6 Checking the Configuration

After OSPF attributes in different types of network are set, you can check OSPF neighbor information, and interface information.

Prerequisite

All configurations of OSPF attributes in different types of network are complete.

Procedure

- Run the **display ospf** [*process-id*] **interface** command to check information about OSPF interfaces.
- Run the **display ospf** [*process-id*] **peer** command to check information about OSPF neighbors.
- Run the **display ospf brief** command to check the interval for sending Hello packets on an NBMA network.

----End

Example

Run the **display ospf interface** command, and you can view the network types and status of OSPF interfaces. For example:

```
<HUAWEI> display ospf interface

          OSPF Process 1 with Router ID 192.168.1.1
                  Interfaces
 Area: 0.0.0.0          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/1/0           192.168.1.1    P2P     P-2-P   1      100

 Area: 0.0.0.1          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/0/0           172.16.1.1   Broadcast    DR   1      50
```

Run the **display ospf peer** command, and you can view information about OSPF neighbors, including the IP address, interface priority, and whether is the DR or BDR.

```
<HUAWEI> display ospf peer
          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 192.168.1.1 ( GE3/0/2 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: Full       Mode:Nbr is  Slave   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3     MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 0
   Neighbor is up for
   Authentication Sequence: [ 0 ]
```

Run the **display ospf brief** command, and you can view the interval for sending Hello packets on an NBMA network.

```
<HUAWEI> display ospf brief
          OSPF Process 1 with Router ID 20.1.1.1
                  OSPF Protocol Information

 RouterID: 20.1.1.1          Border Router:
 Multi-VPN-Instance is not enabled
 Global DS-TE Mode: Non-Standard IETF Mode
 Graceful-restart capability: disabled
 Helper support capability  : not configured
 Applications Supported: MPLS Traffic-Engineering
 Spf-schedule-interval: max 10000 ms, start 500 ms, hold 1000 ms
 Default ASE parameters: Metric: 1 Tag: 1 Type: 2
 Route Preference: 10
 ASE Route Preference: 150
 SPF Computation Count: 4
 RFC 1583 Compatible
 Retransmission limitation is disabled
 Area Count: 1   Nssa Area Count: 0
 ExChange/Loading Neighbors: 0


 Area: 0.0.0.0          (MPLS TE not enabled)
 Authtype: None   Area flag: Normal
 SPF scheduled Count: 4
 ExChange/Loading Neighbors: 0

 Interface: 66.1.1.1 ( GE1/0/0 )
 Cost: 1       State: BDR        Type: Broadcast            MTU: 1500
 Priority: 1
 Designated Router: 66.1.1.2
 Backup Designated Router: 66.1.1.1
 Timers: Hello 10 , Dead 40 , Wait 40 , Poll 120 , Retransmit 5 , Transmit Delay 1
```

3.6 Configuring OSPF Stub Areas

By configuring non-backbone areas at the edge of ASs as stub areas, you can reduce the size of the routing table and reduce the number of LSAs to be transmitted.

Applicable Environment

Dividing an AS into different areas can reduce the number of LSAs to be transmitted on the network and enhance OSPF extensibility. For some non-backbone areas at the edge of ASs, you can configure these areas as stub areas to further reduce the size of the routing table and the number of transmitted LSAs.

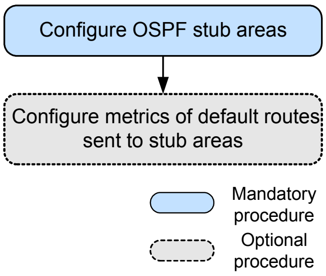
Pre-configuration Tasks

Before configuring OSPF stub areas, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

Figure 3-5 Flowchart of configuring OSPF stub areas


Configure OSPF stub areas
Configure metrics of default routes sent to stub areas
Mandatory procedure
Optional procedure

Related Tasks

3.17.2 Example for Configuring an OSPF Stub Area

3.6.1 Configuring the Current Area as a Stub Area

A stub area is a special area in which ABRs do not flood the received AS external routes. Thus, the number of LSAs is greatly reduced.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

Step 4 Run:

```
stub [ no-summary ]
```

The current area is configured as a stub area.

If the parameter **no-summary** is specified, it indicates that an ABR is disabled from sending summary LSAs to a stub area. To disable an ABR from sending summary LSAs to a stub area, you can specify the parameter **no-summary** in the **stub** only when the **stub** command is configured on the ABR.

To configure an area as a stub area, you need to run the **stub** command on all the routers in this area.

AS external routes in Type 5 LSAs cannot be advertised in a stub area. Therefore, the routers in the stub area learn AS external routes from an ABR. The ABR automatically generates a Type 3 summary LSA with the link state ID being 0.0.0.0 and the network mask being 0.0.0.0 and then advertises the LSA in the entire stub area.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

3.6.2 (Optional) Setting the Cost of the Default Route to a Stub Area

On a border router connected to a stub area, you can set the cost of the default route to the stub area.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

Step 4 Run:

```
stub [ no-summary ]
```

The current area is configured as a stub area.

Step 5 Run:

```
default-cost cost
```

The cost of the default route to the stub area is set.

The parameter *cost* specifies the cost of the Type 3 default route to a stub area. The default value is 1.

This command applies to only the ABR that is connected to a stub area.

Step 6 Run:

```
commit
```

The configuration is committed.

----End

3.6.3 Checking the Configuration

After OSPF stub areas are configured, you can check OSPF LSDB information, neighbors and routing table information.

Prerequisite

All configurations of OSPF stub areas are complete.

Procedure

- Run the **display ospf** [*process-id*] **lsdb** command to check information about the OSPF LSDB.
- Run the **display ospf** [*process-id*] **peer** command to check information about OSPF neighbors.
- Run the **display ospf** [*process-id*] **routing** command to check information about the OSPF routing table.

----End

Example

Run the **display ospf lsdb** command, and you can view the OSPF LSDB, including the types of LSA and link state IDs. For example:

```
<HUAWEI> display ospf lsdb

          OSPF Process 1 with Router ID 10.1.1.1
                  Link State Database

                          Area: 0.0.0.1
 Type      LinkState ID    AdvRouter       Age  Len   Sequence       Metric
 Router    10.1.1.1        10.1.1.1         26  48    80000005            1
 Router    100.1.1.2       100.1.1.2        28  48    80000004            1
```

Run the **display ospf peer** command, and you can view the router IDs, addresses, and status of OSPF neighbors.

```
<HUAWEI> display ospf peer
          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 192.168.1.1 ( GE3/0/2 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: Full       Mode:Nbr is  Slave   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3     MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 5
   Neighbor is up for 00:00:05
   Authentication Sequence: [ 0 ]
```

Run the **display ospf routing** command, and you can view the table of OSPF routes. For example:

```
<HUAWEI> display ospf routing

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.16.1.0/24      2          Transit    192.168.1.2     3.3.3.3         0.0.0.1
 172.17.1.0/24      3          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0
 192.168.1.0/24     1          Stub       192.168.1.2     3.3.3.3         0.0.0.1
 192.168.2.0/24     2          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0


 Total Nets: 4
 Intra Area: 2  Inter Area: 2  ASE: 0  NSSA: 0
```

When Router is in a common area, there are AS external routes in the routing table. After the area where Router resides is configured as a stub area, AS external routes are invisible, ASE is 0.

3.7 Adjusting OSPF Route Selection

By adjusting OSPF route selection, you can enable OSPF to meet the requirements of complex networks.

Applicable Environment

On complex networks, you can adjust OSPF parameters to flexibly adjust the networking and optimize load balancing.

Pre-configuration Tasks

Before adjusting OSPF route selection, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

Related Tasks

3.17.4 Example for Configuring OSPF Load Balancing

3.7.1 Setting the Link Cost for an OSPF Interface

OSPF can automatically calculate the link cost for an interface according to the interface bandwidth. You can also set the link cost for the interface by using the related command.

Context

OSPF can automatically calculate the link cost for an interface according to the interface bandwidth. You can also set the link cost for the interface through commands.

If you do not set the cost of an OSPF interface by using the **ospf cost** *cost* command, OSPF automatically calculates the cost of the interface according to the interface bandwidth. The calculation formula is as follows: Cost of the interface = Bandwidth reference value/Interface bandwidth. The integer of the calculated result is the cost of the interface. If the calculated result is smaller than 1, the cost value is 1. Changing the bandwidth reference value can change the cost of an interface.

By default, no cost is set for an OSPF interface, and the bandwidth reference value is 100Mbps/bandwidth. Therefore, interfaces with different bandwidths have different link costs.

Procedure

- Setting the link cost for an OSPF interface
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     The OSPF interface view is displayed.
  3. Run:
     ```
     ospf cost cost
     ```
     The cost of the OSPF interface is set.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.

- Setting the bandwidth reference value
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ospf [ process-id ]
     ```
     The OSPF process view is displayed.
  3. Run:
     ```
     bandwidth-reference value
     ```
     The bandwidth reference value is set.

     The parameter *value* specifies the bandwidth reference value used to calculate the link cost, in Mbit/s.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.

----End

3.7.2 Setting the Preference for Equal-cost OSPF Routes

When multiple routing protocols discover the routes to the same destination, you can set the priorities of the routes discovered by these routing protocols.

Context

After OSPF calculates equal-cost routes, you can run the **nexthop** command to select the route with the highest priority from the equal-cost routes as the next hop. The smaller the weight, the higher the priority of the route. The default weight is 255, indicating that OSPF discovers equal-cost routes and the number of equal-cost routes exceeds that specified in the **maximum load-balancing** *number* command. In this case, OSPF traffic will be balanced among these equal-cost routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
nexthop ip-address weight value
```

The preference is set for equal-cost routes.

- The parameter *ip-address* specifies the next-hop address of the equal-cost route.

- The parameter *value* specifies the weight of the next hop. The default value is 255. The smaller the weight, the higher the priority of the route.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.7.3 Setting the Maximum Number of Equal-Cost Routes

If multiple routes of the same routing protocol are destined for the same destination and have the same costs, these routes are called equal-cost routes and traffic will be balanced among these routes.

Context

The NE5000E supports load balancing among equal-cost routes. That is, you can configure multiple routes, which have the same destination and preference.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
maximum load-balancing number
```

The maximum number of equal-cost routes is set.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.7.4 Configuring External Route Selection Rules Compatible with RFC 1583

The routing rule defined in RFC 2328 is different from that in RFC 1583. When the same external route is calculated according to multiple LSAs, the routing rule configured through the rfc1583 compatible command is compatible with that defined in RFC 1583.

Context

All devices in an OSPF routing domain must be configured with the same route selection rule. At present, most OSPF routing domains adopt the route selection rules defined in RFC 2328.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 (Optional)Run:

```
rfc1583 compatible
```

The external route selection rules, which are compatible with RFC 1583, are configured.

By default, the routing rule of compatible 1583 is enabled.

NOTE

On a network, if OSPF routers have different configurations of the external route selection rules compatible with RFC 1583, external loops may occur.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.7.5 Checking the Configuration

After OSPF route selection is adjusted, you can check OSPF routing table information and interface information.

Prerequisite

All configurations of adjusting OSPF route selection are complete.

Procedure

- Run the **display ospf** [*process-id*] **interface** command to check information about OSPF interfaces.
- Run the **display ospf** [*process-id*] **routing** command to check information about the OSPF routing table.

----End

Example

Run the **display ospf interface** command, and you can view the network types and costs of OSPF interfaces. For example:

```
<HUAWEI> display ospf interface

          OSPF Process 1 with Router ID 192.168.1.1
                  Interfaces
 Area: 0.0.0.0          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/1/0           192.168.1.1    P2P     P-2-P   1      100

 Area: 0.0.0.1          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/0/0           172.16.1.1   Broadcast    DR   1      50
```

Run the **display ospf routing** command, and you can view the destination addresses and link costs of OSPF routes and whether load balancing is performed among these OSPF routes. For example:

```
<HUAWEI> display ospf routing
          OSPF Process 1 with Router ID 4.4.4.4

                   Routing Tables

 Routing for Network
 Destination        Cost    Type          NextHop         AdvRouter       Area
 172.16.1.0/24      4       Inter-area    192.168.2.1     2.2.2.2         0.0.0.2
                    4       Inter-area    192.168.2.3     2.2.2.2         0.0.0.2
 192.168.0.0/24     2       Inter-area    192.168.2.1     2.2.2.2         0.0.0.2

 Routing for ASEs
 Destination    Cost      Type       Tag         NextHop         AdvRouter
 100.0.0.0/8    1         Type2      1           192.168.2.1     1.1.1.1

 Total Nets: 4
 Intra Area: 1  Inter Area: 2  ASE: 1  NSSA: 0
```

3.8 Controlling OSPF Routing Information

This section describes how to control OSPF routing information. Detailed operations include setting priorities for protocols, importing external routes, and filtering the received routes and LSAs.

Pre-configuration Tasks

Before controlling OSPF routing information, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

3.8.1 Configuring OSPF to Import External Routes

Importing the routes discovered by other routing protocols can enrich OSPF routing information.

Context

OSPF can ensure loop-free intra-area routes and inter-area routes; however, OSPF cannot protect external routes against loops. Therefore, when configuring OSPF to import external routes, avoid the loops caused by manual configurations.

Do as follows on the router that functions as the ASBR running OSPF:

Procedure

- Configuring OSPF to import the routes discovered by other protocols
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     ospf [ process-id ]
     ```

     The OSPF process view is displayed.
  3. Run:

     ```
     import-route { protocol [ process-id ] [ cost cost | route-policy route-policy-name | tag tag | type type ] *}
     ```

     The routes discovered by other protocols are imported.

     - The parameter *protocol* specifies the routing protocol whose routes are imported. It can be **direct**, **static**, **rip**, **ospf**, **isis**, or **bgp**.
     - The parameter *process-id* specifies the process ID of the protocol whose routes are imported. The default value is 1.
     - The parameter **cost** *cost* specifies the cost of a route.
     - The parameter **type** *type* specifies the type of the metric. It can be 1 or 2.
     - The parameter **tag** *tag* specifies the tag in the external LSA.
     - The parameter **route-policy** *route-policy-name* indicates that the matching rules of the specified routing policy are applied.
  4. Run:

     ```
     commit
     ```

     The configuration is committed.
- Setting parameters for OSPF to import routes
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     ospf [ process-id ]
     ```

     The OSPF process view is displayed.
  3. Run:

     ```
     default { cost { cost | inherit-metric } | tag tag | type type } *
     ```

     The default values of parameters (the metric of routes, tag, and type) are set for importing routes.

     - The parameter **cost** *cost* specifies the default metric of the external route imported by OSPF.
     - The parameter **inherit-metric** indicates that the cost of the imported route is the cost carried in the route. If the cost is not specified, the default cost set through the **default** command is used as the cost of the imported route.

     When OSPF imports external routes, you can set default values for some additional parameters, such as the metric of routes to be imported, route tag, and route type. The route tag is used to identify the protocol-related information. For example, it can be used to differentiate AS numbers when OSPF receives BGP routes.

By default, the default metric of the external routes imported by OSPF is 1; the type of the imported external routes is Type 2; the default tag value is 1.

NOTE

You can run one of the following commands to set the cost of the imported route. The following commands are listed in descending order of priority:

- Run the **apply cost** command in a route-policy to set the cost of the imported route.
- Run the **import-route** command for OSPF to set the cost of the imported route.
- Run the **default** command to set the default cost of the imported route.

4. Run:

```
commit
```

The configuration is committed.

----End

3.8.2 Configuring OSPF to Advertise the Default Route to the OSPF Area

A default route is the one to be used when no matched routing entry is found in the routing table. You can either configure default static routes manually or enable dynamic routing protocols such as OSPF and ISIS to generate default routes.

Context

In a routing table, a default route is the route to the network 0.0.0.0 (with the mask being 0.0.0.0). You can check whether the default route is configured by using the **display ip routing-table** command. If the destination address of a packet does not match any entry in the routing table, the packet is sent through a default route. If no default route exists and the destination address of the packet does not match any entry in the routing table, the packet is discarded. An Internet Control Message Protocol (ICMP) packet is then sent, informing the originating host that the destination host or network is unreachable.

Procedure

- Configuring OSPF to Advertise the Default Route to the OSPF Area
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     ospf [ process-id ]
     ```

     The OSPF process view is displayed.
  3. Run the following commands as required:
     - Run:

       ```
       default-route-advertise [ [ always | permit-calculate-other ] | cost
       cost | type type | route-policy route-policy-name ] *
       ```

       OSPF is configured to advertise the default route to the OSPF area.

- **always** indicates that an LSA describing the default route is generated and then advertised regardless of whether there are the active default routes of other OSPF processes in the routing table of the local device.
- **permit-calculate-other** indicates that the local router is still allowed to calculate the default routes advertised by other routers after adverting its default route.
- **route-policy** *route-policy-name* indicates that the local device advertises default routes according to the parameters of the configured routing policy when there are matched default routing entries generated by other OSPF processes.

- Run:

```
default-route-advertise summary cost cost
```

The default cost of a Type 3 summary LSA is set.

Before selecting the preceding parameters, you need to configure VPN. Otherwise, this command cannot be run.

NOTE

- An ASE LSA that describes the default route is generated and then advertised only when there are active default routes of other OSPF processes in the routing table of the local device.
- Before advertising a default route, OSPF compares the preferences of default routes. Therefore, if a static default route is configured on an OSPF router, to add the default route advertised by OSPF to the current routing table, ensure that the preference of the configured static default route is lower than that of the default route advertised by OSPF.

4. Run:

```
commit
```

The configuration is committed.

----End

3.8.3 Configuring OSPF Route Aggregation

After an AS is divided into areas, configuring route aggregation can reduce routing information transmitted between areas, thus reducing the size of the routing table and improving route performance.

Context

Do as follows on the OSPF router.

Procedure

- Configuring ABR Route Aggregation

  Do as follows on the OSPF ABR:

  1. Run:

  ```
  system-view
  ```

  The system view is displayed.

  2. Run:

  ```
  ospf [ process-id ]
  ```

The OSPF process view is displayed.

3. Run:

   ```
   area area-id
   ```

   The OSPF area view is displayed.

4. Run:

   ```
   abr-summary ip-address mask [ [ advertise | not-advertise ] | cost cost ]
   *
   ```

   ABR route aggregation of OSPF is configured.

- Configuring ASBR Route Aggregation

  Do as follows on the OSPF ASBR:

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     ospf [ process-id ]
     ```

     The OSPF process view is displayed.

  3. Run:

     ```
     asbr-summary ip-address mask [ not-advertise | tag tag | cost cost |
     distribute-delay interval ] *
     ```

     ASBR route aggregation of OSPF is configured.

----End

3.8.4 Configuring OSPF to Filter the Received Routes

After a filtering policy is configured for the OSPF routes that need to be delivered to the routing management module, only the routes that match the policy will be added to the routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } import
```

OSPF is configured to filter the received routes.

- The parameter *acl-number* specifies the number of a basic ACL.
- The parameter **acl-name** *acl-name* specifies the name of an ACL.
- The parameter **ip-prefix** *ip-prefix-name* specifies the name of an IP prefix list.

OSPF is a link-state dynamic routing protocol, with routing information carried in the LSA. Therefore, the **filter-policy import** command cannot be used to filter the advertised or received LSAs.

The **filter-policy import** command is used to filter the routes calculated by OSPF. Only the routes that pass the filtering are added to the routing table. Routes that do not pass the filtering can not added to the OSPF routing table, but can be advertised. Therefore, the LSDB is not affected regardless of whether the received routes pass the filtering.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.8.5 Configuring OSPF to Filter the Routes to Be Advertised

After a filtering policy is configured for OSPF routes to be imported, only the routes that match the policy will be advertised.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 (Optional) Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } export
[ protocol [ process-id ] ]
```

OSPF is configured to filter the routes imported through the **import-route** command. Only the routes that pass the filtering are advertised.

- The parameter *acl-number* specifies the number of a basic ACL.
- The parameter **acl-name** *acl-name* specifies the name of an ACL.
- The parameter **ip-prefix** *ip-prefix-name* specifies the name of an IP prefix list.

You can specify the parameter *protocol* [*process-id*] to filter the routes of a certain routing protocol or a certain OSPF process. If *protocol* [*process-id*] is not specified, OSPF filters all the imported routes.

NOTE

- The **import-route** command cannot be used to import external default routes.
- OSPF filters the imported routes, and generates Type 5 LSAs to advertise only external routes that passing the filtering.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.8.6 Configuring OSPF to Filter Type 3 LSAs

By configuring filtering conditions for Type 3 LSAs, you can allow only the routes that pass the filtering to be received or advertised.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

Step 4 Run:

```
filter { acl-number | acl-name acl-name | ip-prefix ip-prefix-name | route-policy
route-policy-name } { export | import }
```

OSPF is configured to filter the Type 3 LSAs generated by ABRs.

- The parameter *acl-number* specifies the number of a basic ACL.
- The parameter **acl-name** *acl-name* specifies the name of an ACL.
- The parameter **ip-prefix** *ip-prefix-name* specifies the name of an IP prefix list.
- The parameter **route-policy** *route-policy-name* specifies the matching rules used to filter the Type 3 LSAs generated by ABRs.
- The parameter **export** indicates that the outgoing summary LSAs in the local area will be filtered.
- If the parameter **import** is specified, it indicates that the incoming summary LSAs in the local area are to be filtered.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

3.8.7 Checking the Configuration

After OSPF routing information is controlled, you can check OSPF LSDB information.

Prerequisite

All configurations of controlling OSPF routing information are complete.

Procedure

- Run the **display ospf** [*process-id*] **lsdb** command to check information about the OSPF LSDB.

----End

Example

Run the **display ospf lsdb** command, and you can view information about the OSPF LSDB, including the link state IDs in the LSAs and information about the routers that advertise or generate LSAs. For example:

```
<HUAWEI> display ospf lsdb
          OSPF Process 1 with Router ID 1.1.1.1
                  Link State Database

                          Area: 0.0.0.0
 Type      LinkState ID    AdvRouter          Age  Len    Sequence   Metric
 Router    2.2.2.2         2.2.2.2             98  36    8000000B       1
 Network   10.1.1.2        2.2.2.2             98  32    80000004       0
 Sum-Net   20.1.1.0        2.2.2.2            286  28    80000001       1
 Sum-Asbr  2.2.2.2         1.1.1.1             61  28    80000001       1

                  AS External Database
 Type      LinkState ID    AdvRouter          Age  Len    Sequence   Metric
 External  0.0.0.0         2.2.2.2           1128  36    80000001       1
 External  100.1.1.0       2.2.2.2            538  36    80000002       1

          Type 9 Opaque (Link-Local Scope) Database. Area: 0.0.0.0
 Type      LinkState ID    AdvRouter          Age  Len    Sequence   Interfaces
 Opq-Link  3.0.0.0         2.2.2.2             12  44    80000001    10.1.1.1

                  Type 10 Opaque (Area-Local Scope) Database
 Type      LinkState ID    AdvRouter          Age  Len    Sequence   Area
 Opq-Area  1.0.0.1         1.1.1.1              4  200   80000003    0.0.0.0
 Opq-Area  1.0.0.0         1.1.1.1           1641  28    80000001    0.0.0.0
```

3.9 Configuring OSPF IP FRR

In the case of a link fault, a device enabled with OSPF IP FRR can fast switch traffic to the backup link. This protects traffic and greatly improves the reliability of OSPF networks.

Applicable Environment

With the development of networks, services such as Voice over IP (VoIP) and on-line video services require high-quality real-time transmission. Nevertheless, if an OSPF fault occurs, traffic can be switched to a new link only after the processes, including fault detection at the millisecond level, notifying the fault to the routing control plane at the millisecond level, generating and flooding new topology information at the tens of milliseconds level, triggering SPF calculation at the tens of milliseconds level, and notifying and installing a new route at the hundreds-of-milliseconds level, are complete. As a result, it takes much more than 50 ms to restore the faulty link, which cannot meet the requirement for real-time services on the network.

With OSPF IP FRR that calculates a backup link in advance, devices can fast switch traffic to the backup link without interrupting traffic when the primary link becomes faulty. This protects traffic and thus greatly improves the reliability of OSPF networks.

OSPF IP FRR is applicable to the services that are sensitive to packet delay and packet loss.

After OSPF IP FRR is configured, the lower layer needs to fast respond to a link change so that traffic can be rapidly switched to the backup link. After FRR and BFD are bound, link failures can be detected rapidly. This ensures that traffic is rapidly switched to the backup link in the case of a link failure.

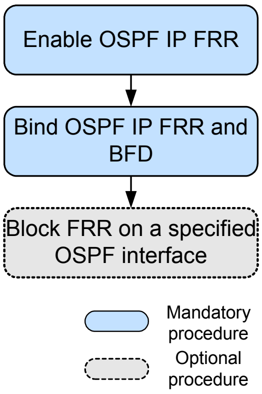
Pre-configuration Tasks

Before configuring OSPF IP FRR, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

Figure 3-6 Flowchart for configuring OSPF IP FRR



Related Tasks

3.17.6 Example for Configuring OSPF IP FRR

3.9.1 Enabling OSPF IP FRR

The Loop Free Alternate (LFA) is a method of implementing basic FRR functions.

Context

FRR calculation consumes a large number of CPU resources. When there are import features such as routing protocol, you need to delay FRR calculation.

After FRR calculation is delayed, devices process important services such as route calculation first.

Do as follows on the router that needs to protect traffic to be forwarded:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id | router-id router-id | vpn-instance vpn-instance-name ] *
```

An OSPF process is started and the OSPF view is displayed.

Step 3 Run:

```
frr
```

The OSPF IP FRR view is displayed.

Step 4 Run:

```
loop-free-alternate
```

OSPF IP FRR is enabled to generate a loop-free backup link.

NOTE

OSPF can generate a loop-free backup link only when the OSPF IP FRR traffic protection inequality is met.

----End

3.9.2 Binding IP FRR and BFD

Binding IP FRR and BFD enables the lower layer to fast respond to a link change so that traffic can be rapidly switched to the backup link when the primary link becomes faulty.

Context

After the parameter **frr-binding** is set to bind the BFD status to the link status of an interface, link failures can be detected rapidly. This ensures that traffic is rapidly switched to the backup link in the case of a link failure.

Do as follows on the router where IP FRR and BFD need to be bound:

Procedure

- Bind IP FRR and BFD in an OSPF process.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     ospf
     ```

     An OSPF process is started and the OSPF view is displayed.
  3. Run:

     ```
     bfd all-interfaces frr-binding
     ```

     IP FRR and BFD are bound in the OSPF process.

- Bind IP FRR and BFD on a specified OSPF interface.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     The interface view is displayed.
  3. Run:
     ```
     ospf bfd frr-binding
     ```
     IP FRR and BFD are bound on the interface.

----End

Follow-up Procedure

The BFD configuration on an interface takes precedence over that in an OSPF process. If BFD is enabled on an interface, a BFD session is established according to the BFD parameters set on the interface.

3.9.3 (Optional) Blocking FRR on an OSPF Interface

For the interfaces on a device bearing key services, you must ensure that this device detours around the backup path so that services will not affected during FRR calculation.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of an OSPF interface enabled with FRR is displayed.

Step 3 Run:

```
ospf frr block
```

FRR is blocked on the OSPF interface.

----End

3.9.4 Checking the Configuration

After configuring OSPF IP FRR, you can view the information about the backup next hop.

Prerequisite

All OSPF IP FRR configurations are complete.

Procedure

- Run the **display ospf** [*process-id*] **routing** command to check the information about the primary link and backup link of a route after configuring OSPF IP FRR.

----End

Example

Display the routes to a specified OSPF router, including information about the backup next hop. For example:

```
<HUAWEI> display ospf routing 10.1.1.1
         OSPF Process 1 with Router ID 1.1.1.1

 Destination    : 10.1.1.0/24
 AdverRouter    : 10.1.1.1           Area             : 0.0.0.0
 Cost           : 1                  Type             : Transit
 NextHop        : 10.1.1.2           Interface        : GE1/0/0
 Priority       : High               Age              : 17h03m33s
 Backup NextHop : 10.1.1.3           Backup Interface : GE1/0/1
 Backup Type    : LFA LINK
```

3.10 Configuring BFD for OSPF

If there are high requirements for data transmission, and OSPF convergence needs to be speeded up when the link status changes, you can configure BFD on OSPF links. After detecting a link failure, BFD notifies the routing protocol of the failure, which triggers fast convergence. When the neighbor relationship is Down, the BFD session is deleted dynamically.

Applicable Environment

The link fault or the topology change may cause devices to recalculate routes. Therefore, the convergence of routing protocols must be sped up to improve the network performance.

Link faults are inevitable. Therefore, a feasible solution is required to fast detect faults and notify routing protocols of the faults immediately. If BFD is associated with routing protocols, once a link fault occurs, BFD can speed up the convergence of routing protocols.

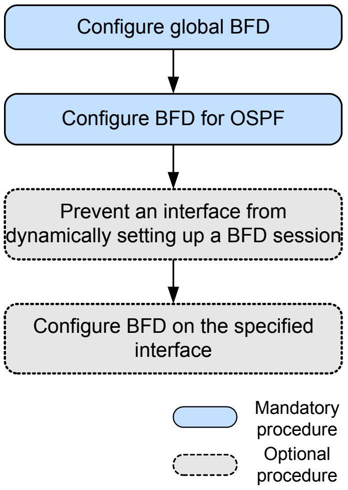
Pre-configuration Tasks

Before configuring BFD for OSPF, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

Figure 3-7 Flowchart of configuring BFD for OSPF


Configure global BFD
Configure BFD for OSPF
Prevent an interface from dynamically setting up a BFD session
Configure BFD on the specified interface
Mandatory procedure
Optional procedure

Related Tasks

3.17.7 Example for Configuring BFD for OSPF

3.10.1 Configuring BFD Globally

Before creating a BFD session, you need to enable BFD globally.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

BFD is configured globally, and the global BFD view is displayed.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

3.10.2 Configuring BFD for OSPF

On the two routers that need to establish a BFD session, you can configure BFD for all the interfaces in a certain OSPF process.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF view is displayed.

Step 3 Run:

```
bfd all-interfaces enable
```

BFD for OSPF is enabled to establish the BFD session.

If all the interfaces in a certain process are configured with BFD and their neighbor relationships are in the Full state, OSPF establishes BFD sessions on all the interfaces in the process.

Run the **bfd all-interfaces** { **min-rx-interval** *receive-interval* | **min-tx-interval** *transmit-interval* | **detect-multiplier** *multiplier-value* } * command to set parameters for BFD sessions.

- The parameter **min-rx-interval** *receive-interval* specifies the expected minimum interval for receiving BFD packets from the neighbor.
- The parameter **min-tx-interval** *transmit-interval* specifies the minimum interval for sending BFD packets to the neighbor.
- The parameter **detect-multiplier** *multiplier-value* specifies the local detection multiplier.

NOTE

If only the **bfd all-interfaces** { **min-rx-interval** *receive-interval* | **min-tx-interval** *transmit-interval* | **detect-multiplier** *multiplier-value* } * command is run to set BFD parameters, and the **bfd all-interfaces enable** command is not run, BFD cannot be enabled.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.10.3 (Optional) Preventing an Interface from Dynamically Establishing a BFD Session

To disable BFD on some interfaces, you need to prevent these interfaces from dynamically setting up BFD sessions.

Context

After the **bfd all-interfaces enable** command is run in an OSPF process, BFD sessions can be established on all the OSPF interfaces whose neighbor relationships are Full.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of the interface enabled with BFD for OSPF is displayed.

Step 3 Run:

```
ospf bfd block
```

The interface is prevented from dynamically establishing a BFD session.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.10.4 (Optional) Configuring BFD for a Specified Interface

To configure BFD only on some interfaces and not to enable OSPF BFD globally, or to require some interfaces to fast detect link failures after configuring OSPF BFD on them, you can configure BFD on the specified interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of the interface enabled with BFD for OSPF is displayed.

Step 3 Run:

```
ospf bfd enable
```

BFD is enabled on the interface to establish the BFD session.

If all the interfaces in a certain process are configured with BFD and their neighbor relationships are in the Full state, OSPF establishes BFD sessions on all the interfaces in the process by using default BFD parameters.

Run the **ospf bfd** { **min-rx-interval** *receive-interval* | **min-tx-interval** *transmit- interval* | **detect-multiplier** *multiplier-value* } * command to set parameters for BFD sessions.

NOTE

- The BFD priority configured on an interface is higher than the BFD priority configured in a process. That is, if BFD is enabled on an interface, BFD parameters on the interface are used to establish BFD sessions.
- If only the **ospf bfd** { **min-rx-interval** *receive-interval* | **min-tx-interval** *transmit- interval* | **detect-multiplier** *multiplier-value* } * command is run to set BFD parameters, and the **ospf bfd enable** command is not run, BFD cannot be enabled on the interface.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.10.5 Checking the Configuration

After BFD for OSPF is configured, you can check information about the BFD session.

Prerequisite

All configurations of BFD for OSPF are complete.

Procedure

- Run one of the following commands to check the BFD session:
  - **display ospf** [*process-id*] **bfd session** *interface-type interface-number* [*router-id*]
  - **display ospf** [*process-id*] **bfd session** { *router-id* | **all** }

----End

Example

Run the **display ospf bfd session all** command, and you can view the "BFDState:up" field, which indicates that the BFD session on the local router is Up. For example:

```
<HUAWEI> display ospf bfd session all

          OSPF Process 1 with Router ID 3.3.3.3
  Area 0.0.0.0 interface 2.2.2.1 ( Pos3/1/0 )'s BFD Sessions

 NeighborId:2.2.2.2          BFDState:up
 LocalIpAdd:2.2.2.1           RemoteIpAdd:2.2.2.2

   Total UP/DOWN BFD Session Number : 1 / 0
```

3.11 Configuring OSPF Fast Convergence

By adjusting OSPF timers, you can implement OSPF fast network convergence.

Pre-configuration Tasks

Before configuring OSPF fast convergence, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

Related Tasks

3.17.5 Example for Configuring OSPF Fast Convergence

3.11.1 Setting the Convergence Priority of OSPF Routes

By setting the convergence priority of OSPF routes, you can shorten the interruption of key services and thus improve the network reliability.

Context

With the integration of network services, different services such as data, voice, and video run on the same network infrastructure, and have different requirements for the network. For Video on Demand (VoD) services, the route convergence speed of the multicast source server is the most critical factor that affects multicast services. It is required that the routes to the multicast source should converge rapidly when network faults occur. On the BGP or MPLS VPN bearer network where OSPF is used to implement the IP connectivity of the backbone network, end-to-end routes between PEs need to be converged rapidly.

You can set priorities for specific routes by setting the convergence priority of OSPF routes so that these routes converge preferentially. This shortens the interruption of key services and improves the reliability of the entire network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF view is displayed.

Step 3 Run:

```
prefix-priority { critical | high | medium } ip-prefix ip-prefix-name
```

The convergence priority of OSPF routes is set.

After the convergence priority of OSPF routes is set, OSPF can calculate and flood LSAs, and synchronize LSDBs according to priorities. This speeds up route convergence. This speeds up route convergence. When an LSA meets multiple priorities, the highest priority takes effect. OSPF calculates LSAs in the sequence of intra-area routes, inter-area routes, and AS external routes. This command makes OSPF calculate route priorities. Convergence priorities are critical, high, medium, and low. To speed up the processing of LSAs with the higher priority, during LSA flooding, the LSAs need to be placed into the corresponding critical, high, medium, and low queues according to priorities.

NOTE

This command takes effect only on the public network.

----End

3.11.2 Setting the Interval for Sending Hello Packets

You can adjust the value of the Hello timer to change the speed of the establishment of the OSPF neighbor relationship and thus change the speed of network convergence.

Context

Hello packets are commonly used packets, which are periodically sent on OSPF interfaces to establish and maintain neighbor relationships. The intervals set on the interfaces connecting two OSPF neighbors need to be the same. Otherwise, the OSPF neighbor relationship cannot be established.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf timer hello interval
```

The interval for sending Hello packets is set on the OSPF interface.

By default, the interval for sending Hello packets on a P2P or broadcast interface is 10s; the interval for sending Hello packets on a P2MP or NBMA interface is 30s; the dead time on the same interface is four times the interval for sending Hello packets.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.3 Setting the Dead Time of the Neighbor Relationship

If no Hello packet is received from a neighbor within a dead interval, the neighbor is considered to be Down.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf timer dead interval
```

The dead time after which the neighbor relationship between two routers is set.

By default, the dead time of the neighbor relationship on a P2P or broadcast interface is 40s; the dead time of the neighbor relationship on a P2MP or NBMA interface is 120s; the dead time of the neighbor relationship on the same interface is four times the interval for sending Hello packets.

NOTE

Setting the dead interval of an OSPF neighbor to be longer than 20s is recommended. If the dead interval of an OSPF neighbor is shorter than 20s, the session may be closed.

Both the Hello timer and the Dead timer are restored to the default values after the network type is changed.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.4 Configuring Smart-discover

After Smart-discover is configured, when the neighbor status of a router changes or the DR or BDR on the multi-address network (broadcast network or NBMA network) changes, a router sends Hello packets to its neighbor immediately without waiting for the Hello timer to expire.

Context

Before Smart-discover is configured, when the neighbor status of the router changes or the DR/BDR on the multi-access network (broadcast or NBMA network) changes, the router does not send Hello packets to its neighbor until the Hello timer expires. This slows down the establishment of neighbor relationships between devices. After Smart-discover is configured, when the neighbor relationship status of the router changes or the DR/BDR on the multi-access network (broadcast or NBMA network) changes, the router can send Hello packets to its neighbor immediately without waiting for the expiration of the Hello timer. This speeds up the establishment of neighbor relationships and thus implements fast convergence of OSPF networks.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf smart-discover
```

Smart-discover is configured on the interface.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.5 Setting the Interval for Updating LSAs

You can set the interval for updating LSAs according to network connections and router resources.

Context

In OSPF, the interval for updating LSAs is defined as 1s. This aims to prevent network connections or frequent route flapping from consuming excessive network bandwidth or device resources.

On a stable network where routes need to be fast converged, you can cancel the interval for updating LSAs by setting the interval to 0 seconds. In this manner, the changes of the topology or the routes can be immediately advertised on the network through LSAs. Route convergence on the network is thus sped up.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
lsa-originate-interval { 0 | intelligent-timer max-interval start-interval hold-
interval [ other-type interval ] | other-type interval [ intelligent-timer max-
interval start-interval hold-interval ] }
```

The interval for updating LSAs is set.

- The parameter **intelligent-timer** indicates that the interval for updating router LSAs and network LSAs is set through an intelligent timer.
- The parameter *max-interval* specifies the maximum interval for updating LSAs, in milliseconds.
- The parameter *start-interval* specifies the initial interval for updating LSAs, in milliseconds.
- The parameter *hold-interval* specifies the hold interval for updating LSAs, in milliseconds.
- The parameter **other-type** *interval* indicates that the interval for updating LSAs excluding Router LSAs and Network LSAs is set.

By default, no intelligent timer is enabled. After an intelligent timer is enabled, the default maximum interval for updating LSAs is 5000 ms, the default initial interval is 500 ms, and the default hold interval is 1000 ms (the interval is expressed in milliseconds). Details about the interval for updating LSAs are as follows:

1. The initial interval for updating LSAs is specified by *start-interval*. The default value is 5000.
2. The interval for updating LSAs for the nth (n ≥ 2) time is equal to *hold-interval* x 2 x (n-1).
3. When the interval specified by *hold-interval* x 2 x (n-1) reaches the maximum interval specified by *max-interval*, OSPF updates LSAs at the maximum interval for three

consecutive times. Then, OSPF goes back to Step **3.1** and updates LSAs at the initial interval specified by *start-interval*.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.6 Setting the Interval for Receiving LSAs

You can set the interval for receiving LSAs according to network connections and router resources.

Context

In OSPF, the interval for receiving LSAs is 1s. This aims to prevent network connections or frequent route flapping from consuming excessive network bandwidth or device resources.

On a stable network where routes need to be fast converged, you can cancel the interval for receiving LSAs by setting the interval to 0 seconds. In this manner, the changes of the topology or the routes can be immediately advertised to the network through LSAs. Route convergence on the network is thus sped up.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
lsa-arrival-interval { interval | intelligent-timer max-interval start-interval
hold-interval }
```

The interval for receiving LSAs is set.

- The parameter *interval* specifies the interval for receiving LSAs, in milliseconds.
- The parameter **intelligent-timer** indicates that the interval for receiving router LSAs or network LSAs is set through an intelligent timer.
- The parameter *max-interval* specifies the maximum interval for receiving LSAs, in milliseconds.
- The parameter *start-interval* specifies the initial interval for receiving LSAs, in milliseconds.
- The parameter *hold-interval* specifies the hold interval for receiving LSAs, in milliseconds.

On a stable network where routes need to be fast converged, you can set the interval for receiving LSAs to 0 seconds so that the changes of the topology or the routes can be detected immediately.

By default, no intelligent timer is enabled. After an intelligent timer is enabled, the default maximum interval for receiving LSAs is 1000 ms, the default initial interval is 500 ms, and the default hold interval is 500 ms. Details about the interval for receiving LSAs are as follows:

1. The initial interval for receiving LSAs is specified by the parameter *start-interval*.
2. The interval for receiving LSAs for the nth (n ⩾ 2) time is equal to *hold-interval* x 2 x (n-1).
3. When the interval specified by *hold-interval* x 2 x (n-1) reaches the maximum interval specified by *max-interval*, OSPF receives LSAs at the maximum interval for three consecutive times. Then, OSPF goes back to Step **3.1** and receives LSAs at the initial interval specified by *start-interval*.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.7 Setting the Interval for the SPF Calculation

By setting the interval for SPF calculation, you can reduce resource consumption caused by frequent network changes.

Context

When the OSPF LSDB changes, the shortest path needs to be recalculated. If a network changes frequently and the shortest path is calculated continually, many system resources are consumed and thus system performance is degraded. By configuring an intelligent timer and properly setting the interval for the SPF calculation, you can prevent excessive system memory and bandwidth resources from being occupied.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
spf-schedule-interval { interval1 | intelligent-timer max-interval start-interval
hold-interval | millisecond interval2 }
```

The interval for the SPF calculation is set.

- The parameter *interval1* specifies the interval for the SPF calculation, in milliseconds.
- The parameter **intelligent-timer** indicates that the interval for the SPF calculation is set through an intelligent timer.
- The parameter *max-interval* specifies the maximum interval for the SPF calculation, in milliseconds.
- The parameter *start-interval* specifies the initial interval for the SPF calculation, in milliseconds.

- The parameter *hold-interval* specifies the hold interval for the SPF calculation, in milliseconds.
- The parameter **millisecond** *interval2* specifies the interval for the SPF calculation, in milliseconds.

By default, an intelligent timer is enabled; the maximum interval for the SPF calculation is 10000 ms, the initial interval is 500 ms, and the hold interval is 1000 ms (the interval is expressed in milliseconds).

After an intelligent timer is enabled, the interval for the SPF calculation is as follows:

1. The initial interval for the SPF calculation is specified by the parameter *start-interval*.
2. The interval for the SPF calculation for the nth (n ≥ 2) time is equal to *hold-interval* x 2 x (n-1).
3. When the interval specified by *hold-interval* x 2 x (n-1) reaches the maximum interval specified by *max-interval*, OSPF performs the SPF calculation at the maximum interval for three consecutive times. Then, OSPF goes back to 3.1 and performs the SPF calculation at the initial interval specified by *start-interval*.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.11.8 Checking the Configuration

After OSPF fast network convergence is implemented, you can check OSPF brief information.

Prerequisite

All configurations of OSPF fast convergence are complete.

Procedure

- Run the **display ospf** [*process-id*] **brief** command to check brief information about the specified OSPF process.

----End

Example

Run the **display ospf brief** command, and you can view detailed information about OSPF timers.

```
<HUAWEI> display ospf brief

          OSPF Process 1 with Router ID 20.1.1.1
                  OSPF Protocol Information

 RouterID: 20.1.1.1          Border Router:
 Multi-VPN-Instance is not enabled
 Global DS-TE Mode: Non-Standard IETF Mode
 Graceful-restart capability: disabled
 Helper support capability  : not configured
 Applications Supported: MPLS Traffic-Engineering
 Spf-schedule-interval: max 10000 ms, start 500 ms, hold 1000 ms
 Default ASE parameters: Metric: 1 Tag: 1 Type: 2
 Route Preference: 10
```

```
ASE Route Preference: 150
SPF Computation Count: 4
RFC 1583 Compatible
Retransmission limitation is disabled
Area Count: 1   Nssa Area Count: 0
ExChange/Loading Neighbors: 0


Area: 0.0.0.0          (MPLS TE not enabled)
Authtype: None   Area flag: Normal
SPF scheduled Count: 4
ExChange/Loading Neighbors: 0

Interface: 66.1.1.1 ( GE1/0/0 )
Cost: 1       State: BDR        Type: Broadcast           MTU: 1500
Priority: 1
Designated Router: 66.1.1.2
Backup Designated Router: 66.1.1.1
Timers: Hello 10 , Dead 40 , Wait 40 , Poll 120 , Retransmit 5 , Transmit Delay 1
```

3.12 Configuring OSPF GR Helper

To avoid traffic interruption and route flapping caused by the active/standby switchover, you can enable OSPF GR.

Applicable Environment

Graceful Restart (GR) is a technology used to ensure normal traffic forwarding and non-stop forwarding of key services during the restart of routing protocols. GR is one of high availability (HA) technologies. HA technologies comprise a set of comprehensive techniques, such as fault-tolerant redundancy, link protection, faulty node recovery, and traffic engineering. As a fault-tolerant redundancy technology, GR is widely used to ensure non-stop forwarding of key services during master/slave switchover and system upgrade.

NOTE

The NE5000E supports only the GR Helper.

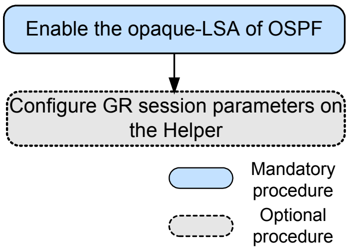
Pre-configuration Tasks

Before configuring OSPF GR, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

Figure 3-8 Flowchart of configuring OSPF GR Helper


Enable the opaque-LSA of OSPF
Configure GR session parameters on the Helper
Mandatory procedure
Optional procedure

3.12.1 Enabling the Opaque-LSA Capability of OSPF

The opaque-LSA capability of OSPF needs to be enabled first because OSPF supports GR through Type 9 LSAs.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF view is displayed.

Step 3 Run:

```
opaque-capability enable
```

The opaque-LSA capability is enabled.

The opaque-LSA capability of OSPF needs to be enabled first because OSPF supports GR through Type 9 LSAs.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.12.2 (Optional) Configuring GR Session Parameters on the Helper

After an OSPF process is restarted through GR, the Restarter and the Helper reestablish the neighbor relationship, exchange routing information, synchronize the LSDB, and update the routing table and forwarding table. This implements OSPF fast convergence and stabilizes the network topology.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [process-id]
```

The OSPF view is displayed.

Step 3 Configuring GR Session Parameters on the Helper.

1. Run:

   ```
   graceful-restart helper-role ignore-external-lsa
   ```

   The Helper is configured to ignore AS-external LSAs.

   By default, the Helper checks AS-external LSAs.

2. Run:

   graceful-restart helper-role planned-only

   The Helper is configured to support only Planned GR.

   By default, the Helper supports both Planned GR and Unplanned GR.

Step 4 Run:

commit

The configuration is committed.

----End

3.12.3 Checking the Configuration

After OSPF GR is configured, you can check the OSPF GR status.

Prerequisite

All configurations of OSPF GR are complete.

Procedure

- Run the **display ospf graceful-restart** [*process-id*] **graceful-restart** [**verbose**] command to check information about OSPF GR.

----End

Example

Run the **display ospf graceful-restart** command, and you can view the configurations about OSPF GR.

```
<HUAWEI> display ospf graceful-restart

         OSPF Process 1 with Router ID 2.2.2.21

 Helper-policy support               : planned, un-planned, external lsa check
 Current GR state                    : Normal

 Number of restarting neighbors : 0

 Last exit reason:
  On Helper     : none
```

3.13 Improving the Stability of an OSPF Network

A stable OSPF network features less route flapping, normal router performance, and good network performance.

Applicable Environment

By setting timers, you can reduce the number of unnecessary packets on networks and reduce the load on the device. Network performance is thus improved.

Pre-configuration Tasks

Before improving the security of an OSPF network, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

3.13.1 Setting the Priority of OSPF

If multiple dynamic routing protocols are running on the same device, routing information needs to be shared and which route to select needs to be determined. In this case, you can set preferences of routing protocols. In this manner, when different protocols discover the routes to the same destination, the route discovered by the protocol with the highest preference will be selected.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
preference [ ase ] { preference | route-policy route-policy-name } *
```

The priority of OSPF is set.

- If the parameter **ase** is specified, it indicates that the preference of AS external routes is set.
- The parameter **preference** specifies the preference of OSPF routes. The smaller the value, the higher the preference.
- If the parameter **route-policy** *route-policy-name* is specified, it indicates that the preference is set for specified routes according to the routing policy.

By default, the preference of OSPF routes is 10. When the parameter **ase** is specified, the default preference of AS external routes is 150.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.2 Setting the Delay in Transmitting LSAs on the Interface

It takes time to transmit OSPF packets on a link. Therefore, a certain delay is added to the aging time of an LSA before the LSA is sent.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf trans-delay interval
```

The delay in transmitting LSAs is set on the interface.

By default, the delay in transmitting LSAs is 1s.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.3 Setting the Interval for Retransmitting LSAs Between Adjacent Routers

After a router sends an LSA to its neighbor, the router expects to receive an LSAck packet from its neighbor. If the router does not receive an LSAck packet within the LSA retransmission interval, it retransmits the LSA to the neighbor.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run:

```
ospf timer retransmit interval
```

The interval for retransmitting LSAs between adjacent routers is set.

By default, the interval for retransmitting LSAs is 5 seconds.

NOTE

The interval for retransmitting LSAs between adjacent routers cannot be set too small. Otherwise, certain LSAs are retransmitted unnecessarily. Generally, the interval needs to be greater than the round trip time of a packet transmitted between two routers.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.4 Configuring Secure Synchronization

The configuration of secure synchronization prevents traffic loss after a device is restarted.

Context

When the routers in an area just finish synchronizing the LSDBs, the LSDBs of these routers are different from each other. As a result, route flapping occurs. You can configure secure synchronization to solve this problem. This, however, may delay the establishment of the OSPF adjacency relationship.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [process-id ]
```

The OSPF view is displayed.

Step 3 Run:

```
safe-sync enable
```

Secure synchronization is configured.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.5 Configuring a Stub Router

When the router has a heavy load and cannot forward any other packets, you can configure it as a stub router. After the router is configured as a stub router, other OSPF router do not use this router to forward data but they can have a route to this stub router.

Context

A stub router is used to control traffic and instruct other OSPF routers not to use it to forward data. Other OSPF routers can have a route to the stub router.

The metric of links in the Router LSAs generated by the stub router is set to the maximum value (65535).

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
stub-router [ on-startup [ interval ] ]
```

A stub router is configured.

The parameter **on-startup** [*interval*] specifies the interval during which the router remains to be a stub router. By default, the interval is 500 seconds.

NOTE

There is no relation between the stub router configured through this command and the router in a stub area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.6 Suppressing an Interface from Receiving or Sending OSPF Packets

By suppressing the OSPF interface from receiving and sending OSPF packets, you can prevent routers on a certain network from obtaining OSPF routing information and prevent the local router from receiving the routing updates advertised by other routers.

Context

After an OSPF interface is set to be in the silent state, the interface can still advertise its direct routes. Hello packets on the interface, however, cannot be forwarded. Therefore, no neighbor relationship can be established on the interface. This can enhance the networking adaptability of OSPF and reduce the consumption of system resources.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
silent-interface { all | interface-type interface-number }
```

The interface is suppressed from receiving or sending OSPF packets.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.13.7 Checking the Configuration

After OSPF features are configured to improve the stability of an OSPF network, you can check OSPF brief information, and OSPF routing table information .

Prerequisite

All configurations of improving the stability of an OSPF network are complete.

Procedure

- Run the **display ospf** [*process-id*] **brief** command to check brief information about the specified OSPF process.
- Run the **display ip routing-table** command to check information about the IP routing table.

----End

Example

Run the **display ospf** [*process-id*] **brief** command, and you can view details about timers of OSPF packets.

```
<HUAWEI> display ospf brief

          OSPF Process 1 with Router ID 1.1.1.1
                  OSPF Protocol Information

 RouterID: 1.1.1.1          Border Router:
 Multi-VPN-Instance is not enabled
 Global DS-TE Mode: Non-Standard IETF Mode
 Graceful-restart capability: disabled
 Helper support capability  : not configured
 Applications Supported: MPLS Traffic-Engineering
 Spf-schedule-interval: max 3000 ms, start 200 ms, hold 500 ms
 Default ASE parameters: Metric: 1 Tag: 1 Type: 2
 Route Preference: 10
 ASE Route Preference: 150
 SPF Computation Count: 14
 RFC 1583 Compatible
 Retransmission limitation is disabled
 Area Count: 1   Nssa Area Count: 0
 ExChange/Loading Neighbors: 0
 BUILD_VERSION: Image: Svn: 3120, Date: 26-01-2010, Time: 00:10 am

 Area: 0.0.0.0          (MPLS TE not enabled)
 Authtype: None   Area flag: Normal
 SPF scheduled Count: 14
 ExChange/Loading Neighbors: 0
 Interface: 192.168.1.1 ( Ethernet3/0/2 )
 Cost: 1       State: DR         Type: Broadcast           MTU: 1500
 Priority: 1
 Designated Router: 192.168.1.1
 Backup Designated Router: 192.168.1.3
 Timers: Hello 10 , Dead 30 , Wait 30 , Poll 120 , Retransmit 5 , Transmit Delay 1
```

Run the **display ip routing-table** command, and you can view that the cost is set to 65535, which indicating that the device is configured as a stub router, as well as the priority of the routing protocol.

```
<HUAWEI> display ip routing-table

Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables :_public_
         Destinations : 4       Routes : 4

Destination/Mask    Proto  Pre  Cost      Flags NextHop         Interface
        1.1.1.1/32  OSPF   10   65536       D  10.1.1.1        Pos1/0/0
        2.2.2.2/32  Direct 0    0           D  127.0.0.1       InLoopBack0
        4.4.4.4/32  OSPF   10   65536       D  10.1.3.2        Pos2/0/0
       10.1.1.0/30  Direct 0    0           D  10.1.1.2        Pos1/0/0
       10.1.2.0/30  OSPF   10   65536       D  10.1.1.1        Pos1/0/0
```

3.14 Improving the Security of an OSPF Network

On a network demanding high security, you can configure OSPF authentication and the GTSM to improve the security of the OSPF network.

Applicable Environment

With the increase in attacks on TCP/IP networks and the defects in the TCP/IP protocol suite, network attacks have a greater impact on the network security. Especially attacks on network devices will cause the crash of the network. By configuring the GTSM and authentication, you can improve the security of an OSPF network.

The NE5000E supports the following authentication modes:

- Simple authentication
- MD5 authentication
- HMAC-MD5 authentication

NOTE

The NE5000E supports OSPF GTSM. For detailed configuration of OSPF GTSM, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - Security*

NOTE

Pre-configuration Tasks

Before improving the security of an OSPF network, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

3.14.1 Configuring the Area Authentication Mode

OSPF supports packet authentication. Only the packets that pass the authentication can be received. If packets fail to pass the authentication, the neighbor relationship cannot be established.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
area area-id
```

The OSPF area view is displayed.

Step 4 Run any of the following command to configure the authentication mode of the OSPF area as required:

- Run:

  ```
  authentication-mode simple [ [ plain ] plain-text | cipher cipher-text ]
  ```

  Simple authentication is configured for the OSPF area.

  - **plain** indicates the plain text password.
  - **cipher** indicates the cipher text password. For Message Digest 5 (MD5) or Hashed Message Authentication Code-MD5 (HMAC-MD5) authentication, the authentication mode is in cipher text by default.

- Run:

  ```
  authentication-mode { md5 | hmac-md5 } [ key-id { plain plain-text | [ cipher ]
  cipher-text } ]
  ```

  MD5 authentication is configured for the OSPF area.

  - **md5** indicates the MD5 cipher text authentication mode.
  - **hmac-md5** indicates the HMAC-MD5 cipher text authentication mode.
  - *key-id* specifies the ID of the authentication key.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

3.14.2 Configuring the Interface Authentication Mode

The interface authentication mode is used among neighbor routers to set the authentication mode and password. Its priority is higher than that of the area authentication mode.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPF interface view is displayed.

Step 3 Run any of the following commands to configure the interface authentication mode as required:

- Run:

  ```
  ospf authentication-mode simple [ [ plain ] plain-text | cipher cipher-text ]
  ```

  Simple authentication is configured for the OSPF interface.

  - **simple** indicates simple authentication.
  - **plain** indicates the plain text password. You can input only the plain text password. When you view the configuration file, the password is displayed in plain text. For simple authentication, the authentication mode is in plain text by default.
  - **cipher** indicates the cipher text password. For MD5 or HMAC-MD5 authentication, the authentication mode is in cipher text by default.

- Run:

  ```
  ospf authentication-mode { md5 | hmac-md5 } [ key-id { plain plain-text |
  [ cipher ] cipher-text } ]
  ```

  MD5 authentication is configured for the OSPF interface.

- Run:

  ```
  ospf authentication-mode null
  ```

  The OSPF interface is not authenticated.

  - **md5** indicates the MD5 cipher text authentication mode.
  - **hmac-md5** indicates the HMAC-MD5 cipher text authentication mode.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

3.14.3 Checking the Configuration

After the OSPF functions are configured to improve the security of an OSPF network, you can check the configuration information.

Prerequisite

All configurations of improving the security of an OSPF network are complete.

Procedure

- Run the **display this** command to view the configurations of the system in the current view.

----End

Example

Run the **display this** command, and you can view the authentication mode. For example:

```
<HUAWEI> system-view
[~HUAWEI] display this
#
interface Pos1/0/0
 link-protocol ppp
 ospf authentication-mode simple
#
```

3.15 Configuring the Network Management Function of OSPF

OSPF supports the network management function. You can bind the OSPF MIB to a certain OSPF process, and configure the trap function and log function.

Applicable Environment

Through the Simple Network Management Protocol (SNMP), the OSPF Management Information Base (MIB) manages multicast information exchanged between the NMS and agents.

Pre-configuration Tasks

Before configuring the network management function of OSPF, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPF Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf mib-binding process-id
```

The OSPF process is bound to the MIB.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display ospf** [*process-id*] **brief** command to check brief information about the binding between the OSPF process and the MIB.

```
<HUAWEI> display ospf brief

          OSPF Process 1 with Router ID 100.1.1.200
                  OSPF Protocol Information

 RouterID: 100.1.1.200      Border Router:
 Multi-VPN-Instance is not enabled
 Global DS-TE Mode: Non-Standard IETF Mode
 Graceful-restart capability: disabled
 Helper support capability  : not configured
 Applications Supported: MPLS Traffic-Engineering
 Spf-schedule-interval: max 10000ms, start 500ms, hold 1000ms
 Default ASE parameters: Metric: 1 Tag: 1 Type: 2
 Route Preference: 10
 ASE Route Preference: 150
 SPF Computation Count: 2
 RFC 1583 Compatible
 Retransmission limitation is disabled
 SendPacket Peak-Control: (Disabled)
 This process is currently bound to MIB
 Area Count: 1   Nssa Area Count: 0
 ExChange/Loading Neighbors: 0

 Area: 0.0.0.0          (MPLS TE not enabled)
 Authtype: None   Area flag: Normal
 SPF scheduled Count: 1
 ExChange/Loading Neighbors: 0

 Interface: 1.1.1.1 (LoopBack0)
 Cost: 0       State: P-2-P     Type: P2P       MTU: 1500
 Timers: Hello 10 , Dead 40 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

3.16 Maintaining OSPF

Maintaining OSPF involves resetting OSPF, clearing OSPF statistics, and debugging OSPF.

3.16.1 Clearing OSPF

You can reset OSPF counters.

Context

⚠ **CAUTION**

OSPF information cannot be restored after you clear it. So, confirm the action before you use the command.

To clear OSPF information, run the following **reset** commands in the user view.

Procedure

- Run the **reset ospf** [*process-id*] **counters** [**neighbor** [*interface-type interface-number*] [*router-id*]] command to reset OSPF counters.
  - **counters** indicates OSPF counters.
  - **neighbor** indicates neighbor information on the specified interface.
- Run the **reset ospf** [*process-id*] **frr** command in the user view to perform OSPF IP FRR calculation again.
- Run the **reset ospf** [*process-id*] **peer** [*interface-type interface-number*] *router-id* command to restart OSPF peers.

----End

3.16.2 Resetting OSPF

Restarting OSPF can reset OSPF.

Context

CAUTION

The OSPF adjacency relationship between the routers will be torn down after you run the **reset ospf** command to reset OSPF connections. So, confirm the action before you use the command.

To reset OSPF connections, run the following **reset** commands in the user view.

Procedure

- Run the **reset ospf** [*process-id*] **process** command in the user view to restart the OSPF process.

----End

3.17 Configuring Examples

This section provides several configuration examples of OSPF together with the configuration flowchart. The configuration examples explain networking requirements, configuration notes, and configuration roadmap.

3.17.1 Example for Configuring Basic OSPF Functions

This section describes how to configure basic OSPF functions, including enabling OSPF on each router and specifying network segments in different areas.

Networking Requirements

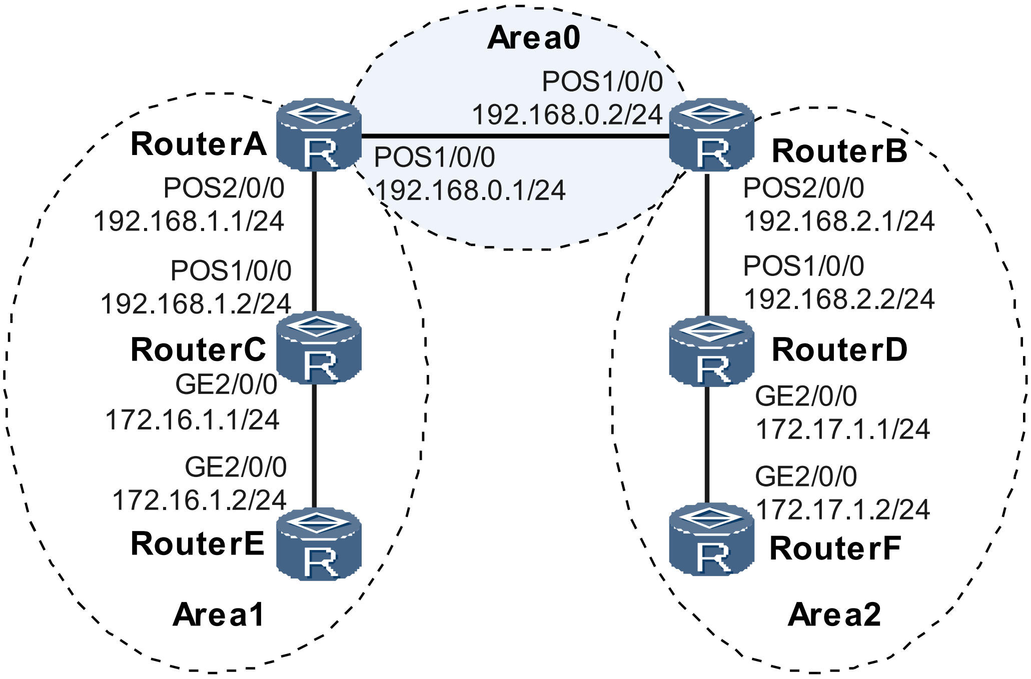
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-9**, all routers run OSPF, and the entire AS is partitioned into three areas. Router A and Router B function as ABRs to forward the routes between areas.

After the configuration is complete, each router should learn the routes to all network segments in the AS.

Figure 3-9 Networking diagram of configuring basic OSPF functions



Configuration Notes

When configuring basic OSPF functions, pay attention to the following:

- The backbone area is responsible for forwarding inter-area routes. In addition, the routing information between non-backbone areas must be forwarded through the backbone area. OSPF defines the following rules for the backbone area:
  - Connectivity must be available between non-backbone areas and the backbone area.
  - Connectivity must be available over the backbone area.

- The intervals for sending Hello, Dead, and Poll packets on the local interface must be the same as that on the peer interface. Otherwise, the OSPF neighboring relationship cannot be established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable OSPF on each router.
2. Specify network segments in different areas.

Data Preparation

To complete the configuration, you need the following data:

Device Name	Router ID	Process ID	IP Address
Router A	1.1.1.1	1	Area 0: 192.168.0.0/24 Area 1: 192.168.1.0/24
Router B	2.2.2.2	1	Area 0: 192.168.0.0/24 Area 2: 192.168.2.0/24
Router C	3.3.3.3	1	Area 1: 192.168.1.0/24 and 172.16.1.0/24
Router D	4.4.4.4	1	Area 2: 192.168.2.0/24 and 172.17.1.0/24
Router E	5.5.5.5	1	Area 1: 172.16.1.0/24
Router F	6.6.6.6	1	Area 2: 172.17.1.0/24

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions.

# Configure Router A.

```
[~RouterA] router id 1.1.1.1
[~RouterA] ospf 1
[~RouterA-ospf-1] area 0
[~RouterA-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] quit
[~RouterA-ospf-1] area 1
[~RouterA-ospf-1-area-0.0.0.1] network 192.168.1.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.1] quit
[~RouterA-ospf-1]  commit
```

# Configure Router B.

```
[~RouterB] router id 2.2.2.2
[~RouterB] ospf 1
```

```
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.0.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] area 2
[~RouterB-ospf-1-area-0.0.0.2] network 192.168.2.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.2] quit
[~RouterB-ospf-1]  commit
```

# Configure Router C.

```
[~RouterC] router id 3.3.3.3
[~RouterC] ospf 1
[~RouterC-ospf-1] area 1
[~RouterC-ospf-1-area-0.0.0.1] network 192.168.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.1] network 172.16.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.1] commit
[~RouterC-ospf-1-area-0.0.0.1] quit
```

# Configure Router D.

```
[~RouterD] router id 4.4.4.4
[~RouterD] ospf 1
[~RouterD-ospf-1] area 2
[~RouterD-ospf-1-area-0.0.0.2] network 192.168.2.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.2] network 172.17.1.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.2] commit
[~RouterD-ospf-1-area-0.0.0.2] quit
```

# Configure Router E.

```
[~RouterE] router id 5.5.5.5
[~RouterE] ospf 1
[~RouterE-ospf-1] area 1
[~RouterE-ospf-1-area-0.0.0.1] network 172.16.1.0 0.0.0.255
[~RouterE-ospf-1-area-0.0.0.1] commit
[~RouterE-ospf-1-area-0.0.0.1] quit
```

# Configure Router F.

```
[~RouterF] router id 6.6.6.6
[~RouterF] ospf 1
[~RouterF-ospf-1] area 2
[~RouterF-ospf-1-area-0.0.0.2] network 172.17.1.0 0.0.0.255
[~RouterF-ospf-1-area-0.0.0.2] commit
[~RouterF-ospf-1-area-0.0.0.2] quit
```

Step 3 Verify the configuration.

# Display the OSPF neighbors of Router A.

```
[~RouterA] display ospf peer
          OSPF Process 1 with Router ID 1.1.1.1
                  Neighbors
 Area 0.0.0.0 interface 192.168.0.1(Pos1/0/0)'s neighbors
Router ID: 2.2.2.2      Address: 192.168.0.2
State: Full  Mode:Nbr is  Master  Priority: 1
   DR: 0.0.0.0   BDR: 0.0.0.0   MTU: 0
   Dead timer due in 36  sec
   Retrans timer interval: 5
   Neighbor is up for 00:15:04
   Authentication Sequence: [ 0 ]
                  Neighbors
 Area 0.0.0.1 interface 192.168.1.1(Pos2/0/0)'s neighbors
Router ID: 3.3.3.3       Address: 192.168.1.2
State: Full  Mode:Nbr is  Master  Priority: 1
   DR: 0.0.0.0   BDR: 0.0.0.0   MTU: 0
   Dead timer due in 39  sec
   Retrans timer interval: 5
   Neighbor is up for 00:07:32
   Authentication Sequence: [ 0 ]
```

# Display the OSPF routes of Router A.

```
[~RouterA] display ospf routing

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.16.1.0/24      2          Transit    192.168.1.2     3.3.3.3         0.0.0.1
 172.17.1.0/24      3          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0
 192.168.2.0/24     2          Inter-area 192.168.0.2     2.2.2.2         0.0.0.0

 Total Nets: 3
 Intra Area: 1  Inter Area: 2  ASE: 0  NSSA: 0
[~RouterA] display ospf routing direct

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 192.168.0.0/24     1          Direct     192.168.0.1     1.1.1.1         0.0.0.0
 192.168.1.0/24     1          Direct     192.168.1.1     1.1.1.1         0.0.0.1

 Total Routes: 2
```

# Display the LSDB of Router A.

```
[~RouterA] display ospf lsdb

          OSPF Process 1 with Router ID 1.1.1.1
                  Link State Database

                          Area: 0.0.0.0
 Type      LinkState ID    AdvRouter        Age  Len    Sequence       Metric
 Router    1.1.1.1         1.1.1.1           93  48     80000004            1
 Router    2.2.2.2         2.2.2.2           92  48     80000004            1
 Sum-Net   172.16.1.0      1.1.1.1         1287  28     80000002            2
 Sum-Net   192.168.1.0     1.1.1.1         1716  28     80000001            1
 Sum-Net   172.17.1.0      2.2.2.2         1336  28     80000001            2
 Sum-Net   192.168.2.0     2.2.2.2           87  28     80000002            1

                          Area: 0.0.0.1
 Type      LinkState ID    AdvRouter        Age  Len    Sequence       Metric
 Router    1.1.1.1         1.1.1.1         1420  48     80000002            1
 Router    3.3.3.3         3.3.3.3         1294  60     80000003            1
 Router    5.5.5.5         5.5.5.5         1296  36     80000002            1
 Network   172.16.1.1      3.3.3.3         1294  32     80000001            0
 Sum-Net   172.17.1.0      1.1.1.1         1325  28     80000001            3
 Sum-Net   192.168.0.0     1.1.1.1         1717  28     80000001            1
 Sum-Net   192.168.2.0     1.1.1.1         1717  28     80000001            2
```

# Display the routing table on Router D and perform the ping operation to test the connectivity.

```
[~RouterD] display ospf routing

          OSPF Process 1 with Router ID 4.4.4.4
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.16.1.0/24      4          Inter-area 192.168.2.1     2.2.2.2         0.0.0.2
 192.168.0.0/24     2          Inter-area 192.168.2.1     2.2.2.2         0.0.0.2
 192.168.1.0/24     3          Inter-area 192.168.2.1     2.2.2.2         0.0.0.2
 Total Nets: 3
 Intra Area: 0  Inter Area: 3  ASE: 0  NSSA: 0
[~RouterD] display ospf routing direct

         OSPF Process 1 with Router ID 4.4.4.4
                  Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.17.1.0/24      1          Direct     172.17.1.1      4.4.4.4         0.0.0.2
 192.168.2.0/24     1          Direct     192.168.2.2     4.4.4.4         0.0.0.2

 Total Routes: 2
[~RouterD] ping 172.16.1.1
  PING 172.16.1.1: 56  data bytes, press CTRL_C to break
    Reply from 172.16.1.1: bytes=56 Sequence=1 ttl=253 time=62 ms
    Reply from 172.16.1.1: bytes=56 Sequence=2 ttl=253 time=16 ms
    Reply from 172.16.1.1: bytes=56 Sequence=3 ttl=253 time=62 ms
    Reply from 172.16.1.1: bytes=56 Sequence=4 ttl=253 time=94 ms
    Reply from 172.16.1.1: bytes=56 Sequence=5 ttl=253 time=63 ms
  --- 172.16.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 16/59/94 ms
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
router id 1.1.1.1
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.0.1 255.255.255.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
ip address 192.168.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
 area 0.0.0.1
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
router id 2.2.2.2
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
ip address 192.168.0.2 255.255.255.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
 area 0.0.0.2
  network 192.168.2.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
router id 3.3.3.3
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.1
  network 192.168.1.0 0.0.0.255
  network 172.16.1.0 0.0.0.255
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
router id 4.4.4.4
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.2 255.255.255.0
#
ospf 1
 area 0.0.0.2
  network 192.168.2.0 0.0.0.255
  network 172.17.1.0 0.0.0.255
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
router id 5.5.5.5
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.16.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.1
  network 172.16.1.0 0.0.0.255
#
```

```
return
```

- Configuration file of the Router F

```
#
 sysname RouterF
#
router id 6.6.6.6
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.17.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.2
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

3.3 Configuring Basic OSPF Functions

3.17.2 Example for Configuring an OSPF Stub Area

This section describes how to configure a stub area that imports static routes to reduce the number of LSAs advertised in this area without affecting the route reachability.

Networking Requirements

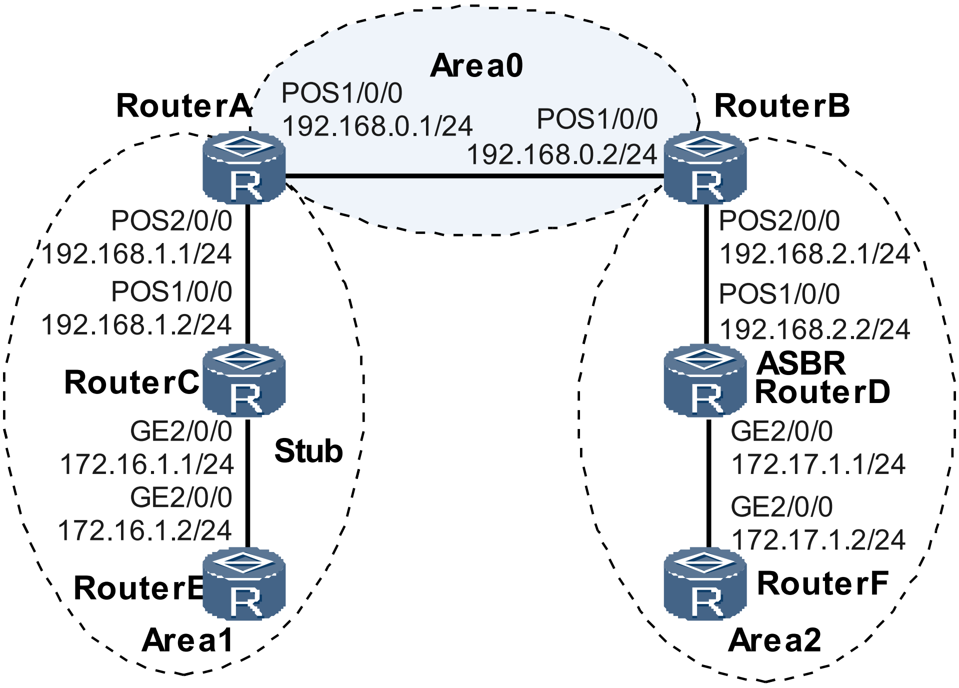
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/ interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-10**, all routers run OSPF, and the entire AS is partitioned into three areas. Router A and Router B function as ABRs to advertise routes between areas; Router D functions as the ASBR to import external routes, that is, static routes.

It is required to configure Area 1 as a stub area to reduce the LSAs advertised to this area without affecting the route reachability.

Figure 3-10 Networking diagram of configuring an OSPF stub area



Configuration Notes

When configuring an OSPF stub area, pay attention to the following:

- The backbone area cannot be configured as a stub area.
- An ASBR cannot exist in a stub area. That is, external routes are not flooded in the stub area.
- A virtual link cannot pass through a stub area.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router to realize interconnection.
2. Configure static routes on Router D and import it into OSPF.
3. Configure Area 1 as a stub area by running the **stub** command on all routers in Area 1 and check the OSPF routing information on Router C.
4. Prevent Router A from advertising Type 3 LSAs to the stub area, and check the OSPF routing information on Router C.

Data Preparation

To complete the configuration, you need the following data:

Device Name	Router ID	Process ID	IP Address

Router A	1.1.1.1	1	Area 0: 192.168.0.0/24 Area 1: 192.168.1.0/24
Router B	2.2.2.2	1	Area 0: 192.168.0.0/24 Area 2: 192.168.2.0/24
Router C	3.3.3.3	1	Area 1: 192.168.1.0/24 and 172.16.1.0/24
Router D	4.4.4.4	1	Area 2: 192.168.2.0/24 and 172.17.1.0/24
Router E	5.5.5.5	1	Area 1: 172.16.1.0/24
Router F	6.6.6.6	1	Area 2: 172.17.1.0/24

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions. For details, see **3.17.1 Example for Configuring Basic OSPF Functions**.

Step 3 Configure Router D to import static routes.

```
[~RouterD] ip route-static 200.0.0.0 8 null 0
[~RouterD] ospf 1
[~RouterD-ospf-1] import-route static type 1
[~RouterD-ospf-1] commit
[~RouterD-ospf-1] quit
```

# Display ABR and ASBR information on Router C.

```
[~RouterC] display ospf abr-asbr
          OSPF Process 1 with Router ID 3.3.3.3
                  Routing Table to ABR and ASBR
 Type        Destination       Area       Cost  NextHop         RtType
 Intra-area  1.1.1.1           0.0.0.1    1     192.168.1.1     ABR
 Inter-area  4.4.4.4           0.0.0.1    3     192.168.1.1     ASBR
```

# Display the OSPF routing table on Router C.

NOTE

If the area where Router C resides is a common area, external routes exist in the routing table.

```
[~RouterC] display ospf routing

          OSPF Process 1 with Router ID 3.3.3.3
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 172.17.1.0/24      4          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1
 192.168.0.0/24     2          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1
 192.168.2.0/24     3          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1

 Routing for ASEs
 Destination        Cost       Type       Tag        NextHop         AdvRouter
 200.0.0.0/8        4          Type1      1          192.168.1.1     4.4.4.4

 Total Nets: 4
 Intra Area: 0  Inter Area: 3  ASE: 1  NSSA: 0
```

Step 4 Configure Area 1 as a stub area.

# Configure Router A.

```
[~RouterA] ospf 1
[~RouterA-ospf-1] area 1
[~RouterA-ospf-1-area-0.0.0.1] stub
[~RouterA-ospf-1-area-0.0.0.1] commit
[~RouterA-ospf-1-area-0.0.0.1] quit
```

# Configure Router C.

```
[~RouterC] ospf 1
[~RouterC-ospf-1] area 1
[~RouterC-ospf-1-area-0.0.0.1] stub
[~RouterC-ospf-1-area-0.0.0.1] commit
[~RouterC-ospf-1-area-0.0.0.1] quit
```

# Configure Router E.

```
[~RouterE] ospf 1
[~RouterE-ospf-1] area 1
[~RouterE-ospf-1-area-0.0.0.1] stub
[~RouterE-ospf-1-area-0.0.0.1] commit
[~RouterE-ospf-1-area-0.0.0.1] quit
```

# Display the routing table on Router C.

NOTE

After the area where Router C resides is configured as a stub area, a default route rather than AS external routes exists in the routing table.

```
[~RouterC] display ospf routing

          OSPF Process 1 with Router ID 3.3.3.3
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area
 0.0.0.0/0          2          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1
 172.17.1.0/24      4          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1
 192.168.0.0/24     2          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1
 192.168.2.0/24     3          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1

 Total Nets: 4
 Intra Area: 0  Inter Area: 4  ASE: 0  NSSA: 0
```

Step 5 # Prevent Router A from advertising Type 3 LSAs to the stub area.

```
[~RouterA] ospf
[~RouterA-ospf-1] area 1
[~RouterA-ospf-1-area-0.0.0.1] stub no-summary
[~RouterA-ospf-1-area-0.0.0.1] commit
[~RouterA-ospf-1-area-0.0.0.1] quit
```

Step 6 Verify the configuration.

# Display the OSPF routing table on Router C.

```
[~RouterC] display ospf routing

          OSPF Process 1 with Router ID 3.3.3.3
                   Routing Tables
 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area
 0.0.0.0/0          2          Inter-area 192.168.1.1     1.1.1.1         0.0.0.1

 Total Nets: 1
 Intra Area: 0  Inter Area: 1  ASE: 0  NSSA: 0
```

NOTE

After the advertisement of summary LSAs to the stub area is disabled, the routing entries on the router in the stub area are further reduced, and only the default route to a destination outside the stub area is reserved.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
router id 1.1.1.1
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.0.1 255.255.255.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
 area 0.0.0.1
  network 192.168.1.0 0.0.0.255
  stub no-summary
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
router id 2.2.2.2
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
ip address 192.168.0.2 255.255.255.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
 area 0.0.0.2
  network 192.168.2.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
```

```
#
router id 3.3.3.3
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.1
  network 192.168.1.0 0.0.0.255
  network 172.16.1.0 0.0.0.255
  stub
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
router id 4.4.4.4
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.2 255.255.255.0
#
ospf 1
 import-route static type 1
 area 0.0.0.2
  network 192.168.2.0 0.0.0.255
  network 172.17.1.0 0.0.0.255
#
ip route-static 200.0.0.0 255.0.0.0 NULL0
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
router id 5.5.5.5
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.16.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.1
  network 172.16.1.0 0.0.0.255
  stub
#
return
```

- Configuration file of the Router F

```
#
 sysname RouterF
#
router id 6.6.6.6
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.17.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.2
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

3.6 Configuring OSPF Stub Areas

3.17.3 Example for Configuring OSPF DR Election

This section describes how to set the DR priority on an interface for DR election on a broadcast network.

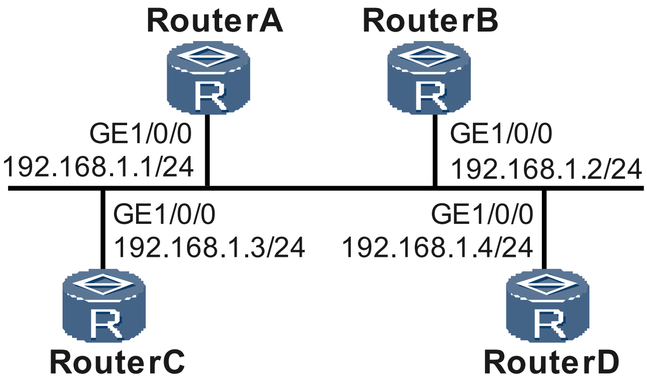
Networking Requirements

CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-11**, the interface of Router A has the highest priority of 100 on the network and is elected as the DR; Router C has the second highest priority and is elected as the BDR. Router B has the priority of 0 and cannot be elected as a DR or a BDR; Router D is not configured with a priority and adopts the default value 1.

Figure 3-11 Networking diagram of configuring OSPF DR election



Configuration Notes

Reconfiguring the DR priority for the router does not change the DR or BDR on a network. You can reelect a DR or BDR by using the following methods. This, however, will result in the interruption of the OSPF neighboring relationship between devices. Therefore, the following methods are used only when necessary.

- Restart the OSPF processes on all routers.
- Configure the **shutdown** and **undo shutdown** commands on the interfaces where the OSPF neighboring relationships are established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router to realize interconnection.
2. Configure the Router ID of each router.
3. Use the default DR priorities and check whether the router is the DR or BDR.
4. Configure the DR priority on an interface and check whether the router is the DR or BDR.

Data Preparation

To complete the configuration, you need the following data:

- Data of Router A, including the router ID (1.1.1.1) and priority (100)
- Data of Router B, including the router ID (2.2.2.2) and priority (0)
- Data of Router C, including the router ID (3.3.3.3) and priority (2)
- Data of Router D, including the router ID (4.4.4.4) and priority (1)

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions.

# Configure Router A.

```
[~RouterA] router id 1.1.1.1
[~RouterA] ospf 1
[~RouterA-ospf-1] area 0
[~RouterA-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] commit
[~RouterA-ospf-1-area-0.0.0.0] quit
```

# Configure Router B.

```
[~RouterB] router id 2.2.2.2
[~RouterB] ospf 1
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
```

# Configure Router C.

```
[~RouterC] router id 3.3.3.3
[~RouterC] ospf 1
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
```

# Configure Router D.

```
[~RouterD] router id 4.4.4.4
[~RouterD] ospf 1
[~RouterD-ospf-1] area 0
```

```
[~RouterD-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.0] commit
[~RouterD-ospf-1-area-0.0.0.0] quit
```

# Display the status of the DR or BDR.

```
[~RouterA] display ospf peer

          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 192.168.1.1 ( GE1/0/0 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: 2-Way  Mode:Nbr is  Slave   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 0
   Neighbor is up for
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.1 ( GE1/0/0 )'s neighbors
 Router ID: 3.3.3.3         Address: 192.168.1.3
   State: Full  Mode:Nbr is  Master   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  34  sec
   Retrans timer interval: 5
   Neighbor is up for 00:02:17
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.1 ( GE1/0/0 )'s neighbors
 Router ID: 4.4.4.4         Address: 192.168.1.4
   State: Full  Mode:Nbr is  Master   Priority: 1
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  32  sec
   Retrans timer interval: 5
   Neighbor is up for 00:02:17
   Authentication Sequence: [ 0 ]
```

When checking the neighbors of Router A, you can view the DR priority and the neighbor status. By default, the DR priority is 1. Now Router D functions as the DR and Router C functions as the BDR.

Step 3 Set the DR priority on each interface.

# Configure Router A.

```
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] ospf dr-priority 100
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

# Configure Router B.

```
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] ospf dr-priority 0
[~RouterB-GigabitEthernet1/0/0] commit
[~RouterB-GigabitEthernet1/0/0] quit
```

# Configure Router C.

```
[~RouterC] interface gigabitethernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] ospf dr-priority 2
[~RouterC-GigabitEthernet1/0/0] commit
[~RouterC-GigabitEthernet1/0/0] quit
```

# Display the status of the DR or BDR.

```
[~RouterD] display ospf peer

          OSPF Process 1 with Router ID 4.4.4.4
```

```
               Neighbors

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 1.1.1.1         Address: 192.168.1.1
   State: Full  Mode:Nbr is  Slave   Priority: 100
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  38  sec
   Retrans timer interval: 5
   Neighbor is up for 02:07:19
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: Full  Mode:Nbr is  Slave   Priority: 0
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  38  sec
   Retrans timer interval: 5
   Neighbor is up for 02:07:19
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 3.3.3.3         Address: 192.168.1.3
   State: Full  Mode:Nbr is  Slave   Priority: 2
   DR: 192.168.1.4   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  30  sec
   Retrans timer interval: 5
   Neighbor is up for 02:07:19
   Authentication Sequence: [ 0 ]
```

Step 4 Restart the OSPF processes.

In the user view of each router, run the **reset ospf 1 process** command to restart the OSPF process.

Step 5 Verify the configuration.

# Display the status of OSPF neighbors.

```
[~RouterD] display ospf peer

          OSPF Process 1 with Router ID 4.4.4.4
               Neighbors

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 1.1.1.1         Address: 192.168.1.1
   State: Full  Mode:Nbr is  Slave   Priority: 100
   DR: 192.168.1.1   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  35  sec
   Retrans timer interval: 5
   Neighbor is up for 00:03:04
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 2.2.2.2         Address: 192.168.1.2
   State: 2-Way  Mode:Nbr is  Slave   Priority: 0
   DR: 192.168.1.1   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  35  sec
   Retrans timer interval: 0
   Neighbor is up for
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 192.168.1.4 ( GE1/0/0 )'s neighbors
 Router ID: 3.3.3.3         Address: 192.168.1.3
   State: Full  Mode:Nbr is  Slave   Priority: 2
   DR: 192.168.1.1   BDR: 192.168.1.3    MTU: 0
   Dead timer due in  37  sec
   Retrans timer interval: 5
   Neighbor is up for 00:03:03
   Authentication Sequence: [ 0 ]
```

Display the status of OSPF interfaces.

```
[~RouterA] display ospf interface

          OSPF Process 1 with Router ID 1.1.1.1
                  Interfaces
 Area: 0.0.0.0          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/1/0           192.168.1.1    Broadcast     DR   1      100
[~RouterB] display ospf interface

          OSPF Process 1 with Router ID 2.2.2.2
                  Interfaces
 Area: 0.0.0.0          (MPLS TE not enabled)
 Interface          IP Address      Type         State    Cost    Pri
 Pos1/1/0           192.168.1.1    Broadcast    DROther   1      100
```

If the neighbor is in the Full state, it indicates that the device establishes neighboring relationship with its neighbor. If the neighbor remains in the 2-Way state, it indicates that neither of them is the DR or BDR. In this case, they need not exchange LSAs.

If the status of the OSPF interface is DROther, it indicates that the interface is neither DR nor BDR.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
router id 1.1.1.1
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.1 255.255.255.0
 ospf dr-priority 100
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
router id 2.2.2.2
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
 ospf dr-priority 0
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
router id 3.3.3.3
```

```
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.3 255.255.255.0
 ospf dr-priority 2
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
router id 4.4.4.4
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.4 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

3.17.4 Example for Configuring OSPF Load Balancing

This section describes how to configure OSPF load balancing, including enabling load balancing and setting priorities for equal-cost routes.

Networking Requirements

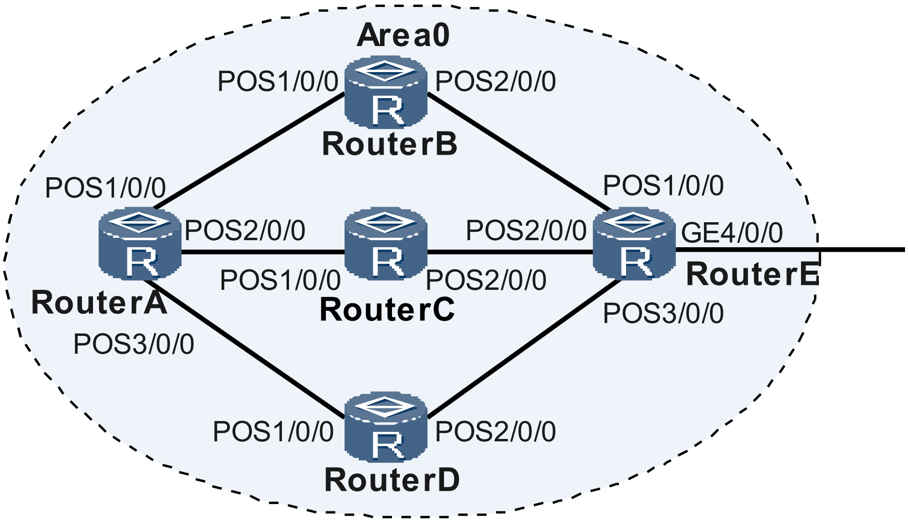
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/ interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-12**, the following conditions are required:

- Router A, Router B, Router C, Router D, and Router E are connected to each other through OSPF.
- Router A, Router B, Router C, Router D, and Router E belong to Area 0.
- Load balancing is configured so that the traffic on Router A is sent to Router E through Router C and Router D.

Figure 3-12 Networking diagram of configuring OSPF load balancing



Device Name	Interface	IP Address	Device Name	Interface	IP Address
RouterA	POS1/0/0	10.1.1.1/24	RouterD	POS1/0/0	10.1.3.2/24
	POS2/0/0	10.1.2.1/24		POS2/0/0	192.168.2.1/24
	POS3/0/0	10.1.3.1/24	RouterE	POS1/0/0	192.168.0.2/24
RouterB	POS1/0/0	10.1.1.2/24		POS2/0/0	192.168.1.2/24
	POS2/0/0	192.168.0.1/24		POS3/0/0	192.168.2.2/24
RouterC	POS1/0/0	10.1.2.2/24		GE4/0/0	172.17.1.1/24
	POS2/0/0	192.168.1.1/24			

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router to realize interconnection.
2. Configure load balancing on Router A.
3. Set the preference of equal-cost routes on Router A.
4. Configure per-packet load balancing mode on Router A.

Data Preparation

To complete the configuration, you need the following data.

- Data of Router A, including router ID (1.1.1.1), OSPF process ID (1), and network segment addresses of Area 0 (10.1.1.0/24, 10.1.2.0/24 and 10.1.3.0/24)
- Data of Router B, including router ID (2.2.2.2), OSPF process ID (1), and network segment addresses of Area 0 (10.1.1.0/8 and 192.168.0.0/8)
- Data of Router C, including router ID (3.3.3.3), OSPF process ID (1), and network segment addresses of Area 0 (10.1.2.0/8 and 192.168.1.0/8)

- Data of Router D, including router ID (4.4.4.4), OSPF process ID (1), and network segment addresses of Area 0 (10.1.3.0/8 and 192.168.2.0/8)
- Data of Router E, including router ID (5.5.5.5), OSPF process ID (1), and network segment addresses of Area 0 (192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24, and 172.17.1.0/24)
- Number of equal-cost routes for load balancing on Router A as 2
- Next hop weights of the routes from Router A to Router B, Router C, and Router D as 2, 1, and 1.

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions. For details, see **3.17.1 Example for Configuring Basic OSPF Functions**.

Step 3 Display the routing table of Router A.

As shown in the routing table, Router A has three valid next hops: Router B (10.1.1.2), Router C (10.1.2.2), and Router D (10.1.3.2). This is because the default maximum number of equal-cost routes is 6.

```
<RouterA> display ip routing-table
Route Flags: R - relay, D - download for forwarding
----------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 15       Routes : 15
Destination/Mask    Proto  Pre  Cost  Flags     NextHop         Interface
       10.1.1.0/24  Direct 0    0       D        10.1.1.1         Pos1/0/0
       10.1.1.1/32  Direct 0    0       D        127.0.0.1        Pos1/0/0
       10.1.1.2/32  Direct 0    0       D        10.1.1.2         Pos1/0/0
       10.1.2.0/24  Direct 0    0       D        10.1.2.1         Pos2/0/0
       10.1.2.1/32  Direct 0    0       D        127.0.0.1        Pos2/0/0
       10.1.2.2/32  Direct 0    0       D        10.1.2.2         Pos2/0/0
       10.1.3.0/24  Direct 0    0       D        10.1.2.1         Pos3/0/0
       10.1.3.1/32  Direct 0    0       D        127.0.0.1        Pos3/0/0
       10.1.3.2/32  Direct 0    0       D        10.1.2.2         Pos3/0/0
    127.0.0.0/8     Direct 0    0       D        127.0.0.1        InLoopBack0
    127.0.0.1/32    Direct 0    0       D        127.0.0.1        InLoopBack0
    192.168.0.0/24  OSPF   10   2       D        10.1.1.2         Pos1/0/0
    192.168.1.0/24  OSPF   10   2       D        10.1.2.2         Pos2/0/0
    192.168.2.0/24  OSPF   10   2       D        10.1.2.2         Pos3/0/0
     172.17.1.0/24  OSPF   10   3       D        10.1.1.2         Pos1/0/0
                    OSPF   10   3       D        10.1.2.2         Pos2/0/0
                    OSPF   10   3       D        10.1.3.2         Pos3/0/0
```

Step 4 Configure two routes on Router A to perform load balancing.

```
[~RouterA] ospf 1
[~RouterA-ospf-1] maximum load-balancing 2
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

# Display the routing table of Router A. You can view two routes performing load balancing on Router A. The maximum number of equal-cost routes is set to 2. Therefore, the next hops Router B (10.1.1.2) and Router C (10.1.2.2) are valid.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
----------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 15       Routes : 15
Destination/Mask    Proto  Pre  Cost  Flags     NextHop         Interface
       10.1.1.0/24  Direct 0    0       D        10.1.1.1         Pos1/0/0
       10.1.1.1/32  Direct 0    0       D        127.0.0.1        Pos1/0/0
       10.1.1.2/32  Direct 0    0       D        10.1.1.2         Pos1/0/0
       10.1.2.0/24  Direct 0    0       D        10.1.2.1         Pos2/0/0
       10.1.2.1/32  Direct 0    0       D        127.0.0.1        Pos2/0/0
       10.1.2.2/32  Direct 0    0       D        10.1.2.2         Pos2/0/0
       10.1.3.0/24  Direct 0    0       D        10.1.2.1         Pos3/0/0
       10.1.3.1/32  Direct 0    0       D        127.0.0.1        Pos3/0/0
       10.1.3.2/32  Direct 0    0       D        10.1.2.2         Pos3/0/0
    127.0.0.0/8     Direct 0    0       D        127.0.0.1        InLoopBack0
    127.0.0.1/32    Direct 0    0       D        127.0.0.1        InLoopBack0
    192.168.0.0/24  OSPF   10   2       D        10.1.1.2         Pos1/0/0
    192.168.1.0/24  OSPF   10   2       D        10.1.2.2         Pos2/0/0
    192.168.2.0/24  OSPF   10   2       D        10.1.2.2         Pos3/0/0
     172.17.1.0/24  OSPF   10   3       D        10.1.1.2         Pos1/0/0
                    OSPF   10   3       D        10.1.2.2         Pos2/0/0
```

Step 5 Set the preference of equal-cost routes on Router A.

```
[~RouterA] ospf 1
[~RouterA-ospf-1] nexthop 10.1.1.2 weight 2
[~RouterA-ospf-1] nexthop 10.1.2.2 weight 1
[~RouterA-ospf-1] nexthop 10.1.3.2 weight 1
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

Step 6 Verify the configuration.

# Display the routing table of Router A.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download for forwarding
----------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 15       Routes : 15
Destination/Mask    Proto  Pre  Cost  Flags     NextHop         Interface
       10.1.1.0/24  Direct 0    0       D        10.1.1.1         Pos1/0/0
       10.1.1.1/32  Direct 0    0       D        127.0.0.1        Pos1/0/0
       10.1.1.2/32  Direct 0    0       D        10.1.1.2         Pos1/0/0
       10.1.2.0/24  Direct 0    0       D        10.1.2.1         Pos2/0/0
       10.1.2.1/32  Direct 0    0       D        127.0.0.1        Pos2/0/0
       10.1.2.2/32  Direct 0    0       D        10.1.2.2         Pos2/0/0
       10.1.3.0/24  Direct 0    0       D        10.1.2.1         Pos3/0/0
       10.1.3.1/32  Direct 0    0       D        127.0.0.1        Pos3/0/0
       10.1.3.2/32  Direct 0    0       D        10.1.2.2         Pos3/0/0
    127.0.0.0/8     Direct 0    0       D        127.0.0.1        InLoopBack0
    127.0.0.1/32    Direct 0    0       D        127.0.0.1        InLoopBack0
    192.168.0.0/24  OSPF   10   2       D        10.1.1.2         Pos1/0/0
    192.168.1.0/24  OSPF   10   2       D        10.1.2.2         Pos2/0/0
    192.168.2.0/24  OSPF   10   2       D        10.1.2.2         Pos3/0/0
     172.17.1.0/24  OSPF   10   3       D        10.1.2.2         Pos2/0/0
                    OSPF   10   3       D        10.1.3.2         Pos3/0/0
```

As shown in the routing table, the priority of the route with the next hop addresses being 10.1.2.2 and 10.1.3.2 is higher than that of the route with the next hop address being 10.1.1.2. Thus, Router A has only two valid next hops, Router C (10.1.2.2) and Router D (10.1.3.2).

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
 #
interface pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.1.1 255.255.255.0
#
```

```
interface pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.2.1 255.255.255.0
#
interface pos3/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.3.1 255.255.255.0
#
ospf 1 router-id 1.1.1.1
 maximum load-balancing 2
 nexthop 10.1.1.2 weight 2
 nexthop 10.1.2.2 weight 1
 nexthop 10.1.3.2 weight 1
 area 0.0.0.0
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 10.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
sysname RouterB
#
interface pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.1.2 255.255.255.0
#
interface pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.0.1 255.255.255.0
#
ospf 1 router-id 2.2.2.2
 area 0.0.0.0
  network 10.1.1.0 0.255.255.255
  network 192.168.0.0 0.255.255.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.2 255.255.255.0
#
interface pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.1.1 255.255.255.0
#
ospf 1 router-id 3.3.3.3
 area 0.0.0.0
  network 10.1.2.0 0.255.255.255
  network 192.168.1.0 0.0.255.255
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
interface pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.3.2 255.255.255.0
#
interface pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.1 255.255.255.0
#
ospf 1 router-id 4.4.4.4
 area 0.0.0.0
  network 10.1.3.0 0.255.255.255
  network 192.168.2.0 0.0.255.255
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
interface GigabitEthernet4/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
#
interface pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.0.2 255.255.255.0
#
interface pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
#
interface pos3/0/0
 link-protocol ppp
 undo shutdown
 ip address 192.168.2.2 255.255.255.0
#
ospf 1 router-id 4.4.4.4
 area 0.0.0.0
  network 192.168.0.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

3.7 Adjusting OSPF Route Selection

3.17.5 Example for Configuring OSPF Fast Convergence

This section describes how to configure OSPF fast convergence, such as adjusting the timer parameter and configuring BFD.

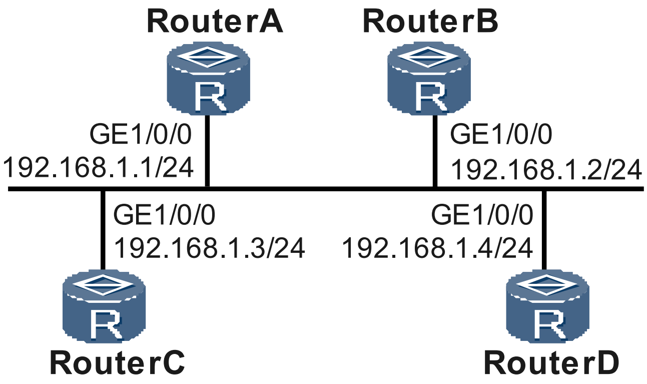
Networking Requirements

CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-13**, on the broadcast network, OSPF runs on the four devices, which belong to the same OSPF area.

Figure 3-13 Networking diagram of configuring OSPF fast convergence



Configuration Notes

When configuring OSPF fast convergence, pay attention to the following:

- You can decrease the interval for sending Hello packets and values of the Dead, Poll, and Wait timers to fast converge OSPF networks. Frequent packet transmission, however, loads the device down. On the other hand, the fast convergence of OSPF networks slows down if the values of these timers are set too large. Therefore, instead of setting values blindly to speed up fast convergence or improve the device performance, you should set appropriate values according to the stability of the network.
- The intervals for sending Hello packets and values of the Dead, Poll, and Wait timers on the local interface must be the same as that on the peer interface. Otherwise, the OSPF neighboring relationship cannot be established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router to realize interconnection.
2. Configure the BFD function on each router.
3. Adjust the holddown time of the OSPF neighbors on each router.
4. Configure the Smart-discover function on each router.
5. Adjust the intervals for configuration updating, LSA receiving, and SPF calculation that are set through an intelligent timer on each router.

Data Preparation

To complete the configuration, you need the following data.

- Holddown time of the OSPF neighbors
- Intervals for LSA updating, LSA receiving, and SPF calculation

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions. For details, see **3.17.1 Example for Configuring Basic OSPF Functions**.

Step 3 Adjust the holddown time of the OSPF neighbors on each router.

# Configure Router A.

```
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] ospf timer dead 30
[~RouterA-GigabitEthernet1/0/0] commit
```

NOTE

- By default, the holddown time of OSPF neighbors is 40s, which is four times the interval for sending Hello packets.
- The holddown time of neighbors of the OSPF-capable interfaces must be longer than the interval for sending Hello packets, and the values of **dead** *interval* on the routers in the same network segment must be the same.
- In this configuration example, the following configurations on each router are the same as that on Router A and are thus not mentioned here.
  - Adjust the holddown time of the OSPF neighbors on each router.
  - Configure the Smart-discover function on each router.
  - Adjust the intervals for configuration updating, LSA receiving, and SPF calculation that are set through an intelligent timer on the router.

Step 4 Configure the Smart-discover function on each router.

# Configure Router A.

```
[~RouterA-GigabitEthernet1/0/0] ospf smart-discover
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

Step 5 Configure the BFD function on each router.

# Configure Router A.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] ospf
[~RouterA-ospf-1] bfd all-interfaces enable
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

Step 6 Adjust the intervals for configuration updating, LSA receiving, and SPF calculation that are set through an intelligent timer on each router.

# Configure Router A.

```
[~RouterA] ospf 1
[~RouterA-ospf-1] lsa-originate-interval intelligent-timer 3000 200 500
[~RouterA-ospf-1] lsa-arrival-interval intelligent-timer 3000 200 500
[~RouterA-ospf-1] spf-schedule-interval intelligent-timer 3000 200 500
[~RouterA-ospf-1] commit
```

Step 7 Verify the configuration.

# Run the **display ospf brief** command on each router to view the brief information about OSPF. Take Router A as an example. You can view the value of timers.

```
[~RouterA] display ospf brief
```

```
         OSPF Process 1 with Router ID 1.1.1.1
                 OSPF Protocol Information

RouterID: 1.1.1.1          Border Router:
Multi-VPN-Instance is not enabled
Global DS-TE Mode: Non-Standard IETF Mode
Graceful-restart capability: disabled
Helper support capability  : not configured
Applications Supported: MPLS Traffic-Engineering
Spf-schedule-interval: max 3000 ms, start 200 ms, hold 500 ms
Default ASE parameters: Metric: 1 Tag: 1 Type: 2
Route Preference: 10
ASE Route Preference: 150
SPF Computation Count: 14
RFC 1583 Compatible
Retransmission limitation is disabled
bfd enabled
BFD Timers: Tx-Interval 1000 , Rrx-Interval 1000 , Multiplier 3
Area Count: 1   Nssa Area Count: 0
ExChange/Loading Neighbors: 0
BUILD_VERSION: Image: Svn: 3120, Date: 26-01-2010, Time: 00:10 am


Area: 0.0.0.0          (MPLS TE not enabled)
Authtype: None   Area flag: Normal
SPF scheduled Count: 14
ExChange/Loading Neighbors: 0

Interface: 192.168.1.1 ( GigabitEthernet3/0/2 )
Cost: 1       State: DR         Type: Broadcast            MTU: 1500
Priority: 1
Designated Router: 192.168.1.1
Backup Designated Router: 192.168.1.3
Timers: Hello 10 , Dead 30 , Wait 30 , Poll 120 , Retransmit 5 , Transmit Delay 1
BFD Timers: Tx-Interval 1000 , Rrx-Interval 1000 , Multiplier 3
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
router id 1.1.1.1
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.1 255.255.255.0
 ospf timer dead 30
 ospf smart-discover
#
ospf 1
 bfd all-interfaces enable
 spf-schedule-interval intelligent-timer 3000 200 500
 lsa-arrival-interval intelligent-timer 3000 200 500
 lsa-originate-interval intelligent-timer 3000 200 500
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
```

```
#
router id 2.2.2.2
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
 ospf timer dead 30
 ospf smart-discover
#
ospf 1
 bfd all-interfaces enable
 spf-schedule-interval intelligent-timer 3000 200 500
 lsa-arrival-interval intelligent-timer 3000 200 500
 lsa-originate-interval intelligent-timer 3000 200 500
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
router id 3.3.3.3
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.3 255.255.255.0
 ospf timer dead 30
 ospf smart-discover
#
ospf 1
 bfd all-interfaces enable
 spf-schedule-interval intelligent-timer 3000 200 500
 lsa-arrival-interval intelligent-timer 3000 200 500
 lsa-originate-interval intelligent-timer 3000 200 500
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
router id 4.4.4.4
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 192.168.1.4 255.255.255.0
 ospf timer dead 30
 ospf smart-discover
#
ospf 1
 bfd all-interfaces enable
 spf-schedule-interval intelligent-timer 3000 200 500
 lsa-arrival-interval intelligent-timer 3000 200 500
 lsa-originate-interval intelligent-timer 3000 200 500
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

Related Tasks

3.11 Configuring OSPF Fast Convergence

3.17.6 Example for Configuring OSPF IP FRR

This section describes how to configure OSPF IP FRR with an example, including how to block FRR on the interface that is not expected to be an interface of a backup link and how to bind OSPF IP FRR to a BFD session.

Networking Requirements

CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

When a fault occurs on the primary link T, traffic is switched to a backup link. In such a scenario, two problems arise:

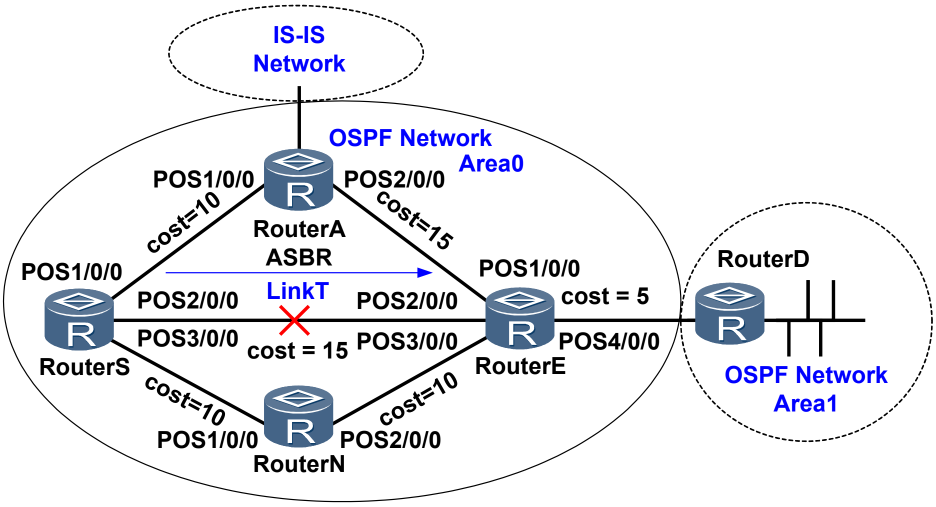
- It takes hundreds of milliseconds for the traffic to be switched to a backup link during OSPF fault restoration. During this period, services are interrupted.
- Traffic will pass Router A after link switching. Router A is an ASBR and is not expected to function as a backup device.

When a fault occurs on the network, OSPF IP FRR can fast switch traffic to the backup link without waiting for route convergence. This ensures uninterrupted traffic transmission. In addition, you can also configure Router A to detour around the backup link.

As shown in **Figure 3-14**:

- All routers run OSPF.
- The link cost meets the OSPF IP FRR traffic protection inequality.
- When the primary link T fails, Router S immediately switches traffic to the backup link. Thus, the traffic is forwarded through Router N.
- Based on the network planning, the link where Router A resides does not function as an FRR backup link.

Figure 3-14 Networking diagram for configuring OSPF IP FRR



Configuration Notes

When configuring OSPF IP FRR, note the following points:

Before configuring OSPF IP FRR, you need to block FRR on the interface that is not expected to be an interface of a backup link. After that, the link where the interface resides is not calculated as a backup link during FRR calculation.

During the configuration of OSPF IP FRR, the lower layer needs to fast respond to a link change so that traffic can be rapidly switched to the backup link. After the **bfd all-interfaces frr-binding** command is run, the BFD session status is associated with the link status of an interface (when the BFD session goes Down, the link status of the interface becomes Down) so that link faults can be rapidly detected.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router.
2. Configure BFD for OSPF on all the devices in Area 0.
3. Set the costs of links to ensure that link T is preferred to transmit traffic.
4. Block FRR on a specified interface of Router S.
5. Enable OSPF IP FRR on Router S to protect the traffic forwarded by Router S.

Data Preparation

To complete the configuration, you need the following data.

Device	Router ID	Interface	IP Address
Router S	1.1.1.1	POS 1/0/0	10.1.1.1/24

		POS 2/0/0	10.1.2.1/24
		POS 3/0/0	10.1.3.1/24
Router A	2.2.2.2	POS 1/0/0	10.1.1.2/24
		POS 2/0/0	20.1.1.2/24
Router N	3.3.3.3	POS 1/0/0	10.1.3.2/24
		POS 2/0/0	20.1.3.2/24
Router E	4.4.4.4	POS 1/0/0	20.1.1.1/24
		POS 2/0/0	20.1.2.1/24
		POS 3/0/0	20.1.3.1/24
		GE 4/0/0	172.17.1.1/24

Procedure

Step 1 Assign an IP address to each interface. The configuration details are not mentioned here.

Step 2 Configure basic OSPF functions. For details, see **Example for Configuring Basic OSPF Functions**.

Step 3 Configure BFD for OSPF on all the devices in Area 0. For details, see **Example for Configuring BFD for OSPF**.

Step 4 Set the costs of links to ensure that link T is preferred to transmit traffic.

# Configure Router S.

```
[~RouterS] interface pos1/0/0
[~RouterS-Pos1/0/0] ospf cost 10
[~RouterS-Pos1/0/0] quit
[~RouterS] interface pos2/0/0
[~RouterS-Pos2/0/0] ospf cost 15
[~RouterS-Pos2/0/0] quit
[~RouterS] interface pos3/0/0
[~RouterS-Pos3/0/0] ospf cost 10
[~RouterS-Pos3/0/0] quit
[~RouterS] commit
```

# Configure Router A.

```
[~RouterA] interface pos2/0/0
[~RouterA-Pos2/0/0] ospf cost 15
[~RouterA-Pos2/0/0] quit
[~RouterA] commit
```

# Configure Router N.

```
[~RouterN] interface pos2/0/0
```

```
[~RouterN-Pos2/0/0] ospf cost 10
[~RouterN-Pos2/0/0] quit
[~RouterN] commit
```

Step 5 Block FRR on a specified interface of Router S.

```
[~RouterS] interface pos1/0/0
[~RouterS-Pos1/0/0] ospf frr block
[~RouterS-Pos1/0/0] quit
[~RouterS] commit
```

Step 6 Enable OSPF IP FRR on Router S.

# Enable OSPF IP FRR on Router S.

```
[~RouterS] ospf
[~RouterS-ospf-1] frr
[~RouterS-ospf-1-frr] loop-free-alternate
[~RouterS-ospf] commit
```

Step 7 Verify the configuration.

# Run the **display ospf routing router-id** command on Router S to view routing information.

```
[~RouterS-ospf-1-frr] display ospf routing router-id 4.4.4.4

          OSPF Process 1 with Router ID 1.1.1.1

 Destination :    4.4.4.4             Route Type :       Intra-area
 Area        :    0.0.0.1             AdvRouter  :       4.4.4.4
 Type        :    ASBR                Age        :       00h31m27s
 URT Cost    :    59
 NextHop     :    20.1.2.1.           Interface  :       POS2/0/0
 Backup Nexthop : 10.1.3.2         Backup Interface : POS3/0/0
 Backup Type : LFA LINK
```

The preceding display shows that a backup route is generated on Router S.

----End

Configuration Files

- Configuration file of Router S

```
#
 sysname RouterS
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
 ospf cost 5
#
interface POS2/0/0
 link-protocol ppp
 ip address 10.1.2.1 255.255.255.0
 ospf cost 15
#
interface POS3/0/0
 link-protocol ppp
 ip address 10.1.3.1 255.255.255.0
 ospf frr block
 ospf cost 10
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
ospf 1 router-id 1.1.1.1
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
  loop-free-alternate
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 10.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router A

```
#
 sysname RouterA
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
 ospf cost 5
#
interface POS2/0/0
 link-protocol ppp
 ip address 20.1.1.2 255.255.255.0
 ospf cost 5
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
ospf 1 router-id 2.2.2.2
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
  loop-free-alternate
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 20.1.2.0 0.0.0.255
#
return
```

- Configuration file of Router N

```
#
 sysname RouterN
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 10.1.3.2 255.255.255.0
 ospf cost 10
#
interface POS2/0/0
 link-protocol ppp
 ip address 20.1.3.2 255.255.255.0
 ospf cost 10
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
ospf 1 router-id 3.3.3.3
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
 area 0.0.0.1
  network 10.1.3.0 0.0.0.255
  network 20.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 20.1.1.1 255.255.255.0
#
interface POS2/0/0
 link-protocol ppp
 ip address 20.1.2.1 255.255.255.0
#
interface POS3/0/0
 link-protocol ppp
 ip address 20.1.3.1 255.255.255.0
#
interface GigabitEthernet4/0/0
 link-protocol ppp
 ip address 172.17.1.1 255.255.255.0
 ospf cost 5
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
ospf 1 router-id 4.4.4.4
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 10.1.3.0 0.0.0.255
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

3.9 Configuring OSPF IP FRR

3.17.7 Example for Configuring BFD for OSPF

This section describes how to configure BFD for OSPF. After BFD for OSPF is configured, BFD can fast detect link faults and report them to OSPF so that service traffic can be transmitted through the backup link.

Networking Requirements

CAUTION

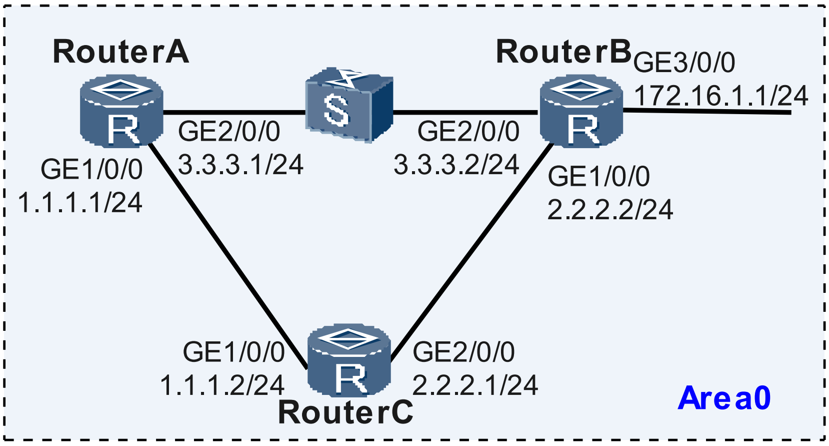
On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-15**, the following conditions are required:

- OSPF runs on Router A, Router B, and Router C.

- BFD is enabled on the OSPF processes of Router A, Router B, and Router C.
- Service traffic is transmitted along the active link Router A → Router B. The link Router A → Router C → Router B functions as the standby link.
- BFD of the interface is configured on the link between Router A and Router B. When the link fails, BFD can quickly detect the fault and notify OSPF of the fault so that service traffic can be transmitted through the standby link.

Figure 3-15 Networking diagram for configuring BFD for OSPF



Configuration Notes

None.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on each router to realize interconnection.
2. Enable global BFD.
3. Enable OSPF BFD on Router A and Router B.

Data Preparation

To complete the configuration, you need the following data:

- Data of Router A, including the router ID (1.1.1.1), OSPF process number (1), and network segment addresses of Area 0 (3.3.3.0/24 and 1.1.1.0/24)
- Data of Router B, including the router ID (2.2.2.2), OSPF process number (1), and network segment addresses of Area 0 (3.3.3.0/24, 2.2.2.0/24, and 172.16.1.0/24)
- Data of Router C, including the router ID (3.3.3.3), OSPF process number (1), and network segment addresses of Area 0 (1.1.1.0/24 and 2.2.2.0/24)
- Minimum interval for sending and receiving BFD packets and local detection multiplier on Router A and Router B

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions.

# Configure Router A.

```
[~RouterA] router id 1.1.1.1
[~RouterA] ospf 1
[~RouterA-ospf-1] area 0
[~RouterA-ospf-1-area-0.0.0.0] network 1.1.1.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] network 3.3.3.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] commit
[~RouterA-ospf-1-area-0.0.0.0] quit
[~RouterA-ospf-1] quit
```

# Configure Router B.

```
[~RouterB] router id 2.2.2.2
[~RouterB] ospf 1
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 2.2.2.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 3.3.3.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 172.16.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] quit
```

# Configure Router C.

```
[~RouterC] router id 3.3.3.3
[~RouterC] ospf 1
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 1.1.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 2.2.2.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
[~RouterC-ospf-1] quit
```

# After the preceding configurations, run the **display ospf peer** command. You can view that the neighboring relationships are established between Router A and Router B, and between Router B and Router C. Take the display on Router A as an example.

```
<RouterA> display ospf peer

          OSPF Process 1 with Router ID 1.1.1.1
                Neighbors

 Area 0.0.0.0 interface 3.3.3.1 ( GE2/0/0 )'s neighbors
 Router ID: 2.2.2.2         Address: 3.3.3.2
   State: Full  Mode:Nbr is  Master   Priority: 1
   DR: 3.3.3.1   BDR: 3.3.3.2    MTU: 0
   Dead timer due in  35  sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:33
   Authentication Sequence: [ 0 ]

 Area 0.0.0.0 interface 1.1.1.1 ( GE1/0/0 )'s neighbors
 Router ID: 3.3.3.3         Address: 1.1.1.2
   State: Full  Mode:Nbr is  Master   Priority: 1
   DR: 1.1.1.1  BDR: 1.1.1.2  MTU: 0
   Dead timer due in  39  sec
   Retrans timer interval: 5
   Neighbor is up for 00:00:08
   Authentication Sequence: [ 0 ]
```

# Display information about the OSPF routing table on Router A, and you can view the routing entries to Router B and Router C. The next hop address of the route to 172.16.1.0/24 is 3.3.3.2, and service traffic is transmitted on the active link Router A → Router B.

```
<RouterA> display ospf routing

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area
 2.2.2.0/24         2          Stub       1.1.1.2         3.3.3.3         0.0.0.0
 172.16.1.0/24      2          Stub       3.3.3.2         2.2.2.2         0.0.0.0

 Total Nets: 2
 Intra Area: 2  Inter Area: 0  ASE: 0  NSSA: 0
```

Step 3 Configure OSPF BFD.

# Enable global BFD on Router A.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] ospf
[~RouterA-ospf-1] bfd all-interfaces enable
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

# Enable global BFD on Router B.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] ospf
[~RouterB-ospf-1] bfd all-interfaces enable
[~RouterB-ospf-1] commit
[~RouterB-ospf-1] quit
```

# Enable global BFD on Router C.

```
[~RouterC] bfd
[~RouterC-bfd] quit
[~RouterC] ospf
[~RouterC-ospf-1] bfd all-interfaces enable
[~RouterC-ospf-1] commit
[~RouterC-ospf-1] quit
```

# After the preceding configurations, run the **display ospf bfd session all** command on Router A, Router B, or Router C. You can view that the BFD session is Up.

Take the display on Router A as an example.

```
[~RouterA] display ospf bfd session all
          OSPF Process 1 with Router ID 1.1.1.1
  Area 0.0.0.0 interface 1.1.1.1(GE1/0/0)'s BFD Sessions

 NeighborId:2.2.2.2          AreaId:0.0.0.0          Interface:GE1/0/0
 BFDState:up                 rx    :1000             tx       :1000
 Multiplier:3                BFD Local Dis:0         LocalIpAdd:1.1.1.1
 RemoteIpAdd:1.1.1.2         Diagnostic Info:0

  Area 0.0.0.0 interface 3.3.3.1(GE2/0/0)'s BFD Sessions

 NeighborId:3.3.3.3          AreaId:0.0.0.0          Interface:GE2/0/0
 BFDState:up                 rx    :1000             tx       :1000
 Multiplier:3                BFD Local Dis:0         LocalIpAdd:3.3.3.1
 RemoteIpAdd:3.3.3.2         Diagnostic Info:0
```

Step 4 Configure BFD on an interface.

# Configure BFD on GigabitEthernet 2/0/0 of Router A. Set the minimum intervals for sending and receiving packets to 500 ms and the local detection multiplier to 4.

```
[~RouterA] interface gigabitethernet 2/0/0
[~RouterA-GigabitEthernet2/0/0] ospf bfd enable
[~RouterA-GigabitEthernet2/0/0] ospf bfd min-tx-interval 500 min-rx-interval 500
detect-multiplier 4
[~RouterA-GigabitEthernet2/0/0] commit
[~RouterA-GigabitEthernet2/0/0] quit
```

# Configure BFD on GigabitEthernet 2/0/0 of Router B. Set the minimum intervals for sending and receiving packets to 500 ms and the local detection multiplier to 4.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] ospf bfd enable
[~RouterB-GigabitEthernet2/0/0] ospf bfd min-tx-interval 500 min-rx-interval 500
detect-multiplier 4
[~RouterB-GigabitEthernet2/0/0] commit
[~RouterB-GigabitEthernet2/0/0] quit
```

# After the preceding configurations, run the **display ospf bfd session all** command on Router A or Router B. You can view that the BFD session is Up.

Take the display on Router B as an example.

```
[~RouterB] display ospf bfd session all

          OSPF Process 1 with Router ID 2.2.2.2
  Area 0.0.0.0 interface 3.3.3.2(GE2/0/0)'s BFD Sessions

 NeighborId:1.1.1.1          AreaId:0.0.0.0          Interface: GE2/0/0
 BFDState:up                rx     :500             tx       :500
 Multiplier:4                BFD Local Dis:0         LocalIpAdd:3.3.3.2
 RemoteIpAdd:3.3.3.1         Diagnostic Info:0

  Area 0.0.0.0 interface 2.2.2.2(GE1/0/0)'s BFD Sessions

 NeighborId:3.3.3.3          AreaId:0.0.0.0          Interface: GE1/0/0
 BFDState:up                 rx    :1000              tx       :1000
 Multiplier:3                BFD Local Dis:0         LocalIpAdd:2.2.2.2
 RemoteIpAdd:2.2.2.1         Diagnostic Info:0
```

Step 5 Verify the configuration.

# Run the **shutdown** command on GigabitEthernet 2/0/0 of Router B to simulate an active link failure.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] shutdown
```

# Display the routing table on Router A. You can view that the standby link Router A → Router C → Router B takes effect after the primary link fails, and the next hop address of the route to 172.16.1.0/24 is 1.1.1.2.

```
<RouterA> display ospf routing

          OSPF Process 1 with Router ID 1.1.1.1
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area
 2.2.2.0/24         2          Stub       1.1.1.2         3.3.3.3         0.0.0.0
 172.16.1.0/24      3          Stub       1.1.1.2         2.2.2.2         0.0.0.0

 Total Nets: 2
 Intra Area: 2  Inter Area: 0  ASE: 0  NSSA: 0
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 router id 1.1.1.1
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 3.3.3.1 255.255.255.0
 ospf bfd enable
 ospf bfd min-tx-interval 500 min-rx-interval 500 detect-multiplier 4
#
ospf 1
 bfd all-interfaces enable
 area 0.0.0.0
  network 3.3.3.0 0.0.0.255
  network 1.1.1.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
 router id 2.2.2.2
#
 bfd
#
 interface GigabitEthernet1/0/0
 undo shutdown
 ip address 2.2.2.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 3.3.3.2 255.255.255.0
 ospf bfd enable
 ospf bfd min-tx-interval 500 min-rx-interval 500 detect-multiplier 4
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
ospf 1
 bfd all-interfaces enable
 area 0.0.0.0
  network 3.3.3.0 0.0.0.255
  network 2.2.2.0 0.0.0.255
  network 172.16.1.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
 router id 3.3.3.3
#
 bfd
#
 interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 2.2.2.1 255.255.255.0
 ospf bfd enable
#
ospf 1
 bfd all-interfaces enable
 area 0.0.0.0
  network 1.1.1.0 0.0.0.255
  network 2.2.2.0 0.0.0.255
#
return
```

Related Tasks

3.10 Configuring BFD for OSPF

3.17.8 Example for Configuring OSPF-BGP Synchronization

This section describes how to configure OSPF-BGP synchronization to minimize the impact of device restart upon the BGP traffic on the network.

Networking Requirements

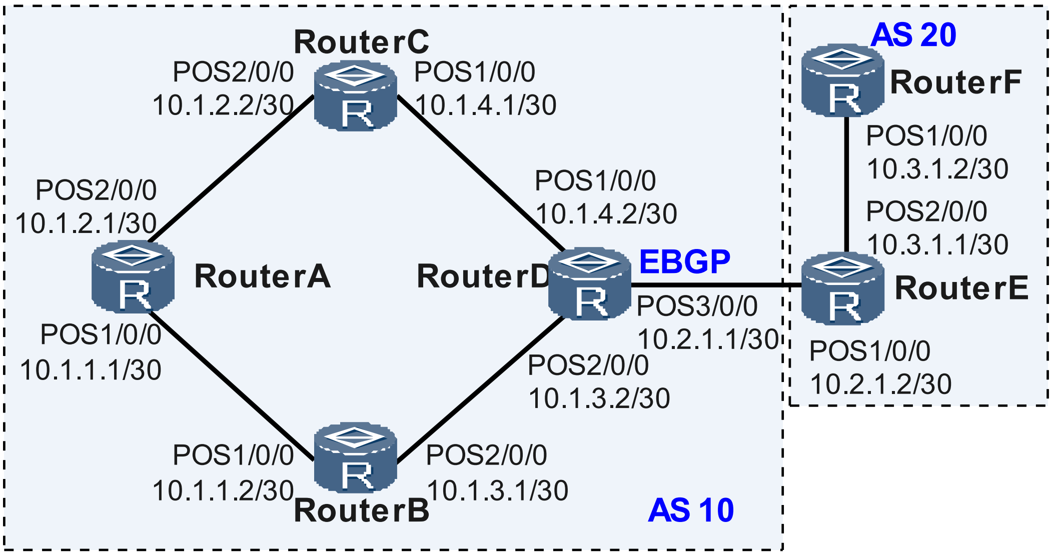
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 3-16**, all routers run BGP. An EBGP connection is set up between RouterD and RouterE. IBGP connections are set up between certain routers in AS 10, and OSPF is used as an IGP protocol.

It is required to enable OSPF-BGP synchronization on Router B so that the restart of Router B does not interrupt the traffic from Router A to AS 20.

Figure 3-16 Networking diagram of configuring OSPF-BGP synchronization



Configuration Notes

None.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable OSPF on Router A, Router B, Router C, and Router D (except 10.2.1.1/30) and specify the same area for all OSPF interfaces.
2. Set up IBGP connections on Router A, Router B, Router C, and Router D (except 10.2.1.1/30).
3. Configure the OSPF cost on Router C.
4. Set up EBGP connections between Router D and Router E.
5. Configure BGP to import direct routes and OSPF processes on Router D.
6. Configure BGP on Router E.

Data Preparation

To complete the configuration, you need the following data:

- Data of Router A, including the Router ID (1.1.1.1), number of the AS to which Router A belongs (10), OSPF process number (1), and network segment addresses of Area 0 (10.1.1.0/30 and 10.1.2.0/30)
- Data of Router B, including the Router ID (2.2.2.2), number of the AS to which Router B belongs (10), OSPF process number (1), and network segment addresses of Area 0 (10.1.1.0/30 and 10.1.3.0/30)
- Data of Router C, including the Router ID (3.3.3.3), number of the AS to which Router C belongs (10), OSPF process number (1), and network segment addresses of Area 0 (10.1.2.0/30 and 10.1.4.0/30)

- Data of Router D, including the Router ID (4.4.4.4), number of the AS to which Router D belongs (10), OSPF process number (1), and network segment addresses of Area 0 (10.1.3.0/30 and 10.1.4.0/30)
- Data of Router E, including the Router ID (5.5.5.5) and number of the AS to which Router E belongs (20)

Procedure

Step 1 Assign an IP address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPF functions. For details, see **3.17.1 Example for Configuring Basic OSPF Functions**.

Step 3 Configure IBGP fully meshed connections.

# Configure Router A.

```
<RouterA> system-view
[~RouterA] interface loopback 0
[~RouterA-LoopBack0] ip address 1.1.1.1 32
[~RouterA-LoopBack0] quit
[~RouterA] bgp 10
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 2.2.2.2 as-number 10
[~RouterA-bgp] peer 2.2.2.2 connect-interface LoopBack 0
[~RouterA-bgp] peer 3.3.3.3 as-number 10
[~RouterA-bgp] peer 3.3.3.3 connect-interface LoopBack 0
[~RouterA-bgp] peer 4.4.4.4 as-number 10
[~RouterA-bgp] peer 4.4.4.4 connect-interface LoopBack 0
[~RouterA-bgp] quit
[~RouterA] commit
```

# Configure Router B.

```
<RouterB> system-view
[~RouterB] interface loopback 0
[~RouterB-LoopBack0] ip address 2.2.2.2 32
[~RouterB-LoopBack0] quit
[~RouterB] bgp 10
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 1.1.1.1 as-number 10
[~RouterB-bgp] peer 1.1.1.1 connect-interface LoopBack 0
[~RouterB-bgp] peer 3.3.3.3 as-number 10
[~RouterB-bgp] peer 3.3.3.3 connect-interface LoopBack 0
[~RouterB-bgp] peer 4.4.4.4 as-number 10
[~RouterB-bgp] peer 4.4.4.4 connect-interface LoopBack 0
[~RouterB-bgp] quit
[~RouterB] commit
```

# Configure Router C.

```
<RouterC> system-view
[~RouterC] interface loopback 0
[~RouterC-LoopBack0] ip address 3.3.3.3 32
[~RouterC-LoopBack0] quit
[~RouterC] bgp 10
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 1.1.1.1 as-number 10
[~RouterC-bgp] peer 1.1.1.1 connect-interface LoopBack 0
[~RouterC-bgp] peer 2.2.2.2 as-number 10
[~RouterC-bgp] peer 2.2.2.2 connect-interface LoopBack 0
[~RouterC-bgp] peer 4.4.4.4 as-number 10
[~RouterC-bgp] peer 4.4.4.4 connect-interface LoopBack 0
[~RouterC-bgp] quit
[~RouterC] commit
```

# Configure Router D.

```
<RouterD> system-view
[~RouterD] interface loopback 0
[~RouterD-LoopBack0] ip address 4.4.4.4 32
[~RouterD-LoopBack0] quit
[~RouterD] bgp 10
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 1.1.1.1 as-number 10
[~RouterD-bgp] peer 1.1.1.1 connect-interface LoopBack 0
[~RouterD-bgp] peer 2.2.2.2 as-number 10
[~RouterD-bgp] peer 2.2.2.2 connect-interface LoopBack 0
[~RouterD-bgp] peer 3.3.3.3 as-number 10
[~RouterD-bgp] peer 3.3.3.3 connect-interface LoopBack 0
[~RouterD-bgp] quit
[~RouterD] commit
```

Step 4 Configure EBGP connections.

# Configure Router D.

```
[~RouterD] bgp 10
[~RouterD-bgp] peer 10.2.1.2 as-number 20
[~RouterD-bgp] import-route direct
[~RouterD-bgp] import-route ospf 1
[~RouterD-bgp] quit
[~RouterD] commit
```

# Configure Router E.

```
[~RouterE] bgp 20
[~RouterE-bgp] peer 10.2.1.1 as-number 10
[~RouterE-bgp] ipv4-family unicast
[~RouterE-bgp-af-ipv4] network 10.3.1.0 30
[~RouterE-bgp-af-ipv4] quit
[~RouterE-bgp] commit
```

Step 5 Configure the OSPF cost on Router C.

```
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] ospf cost 2
[~RouterC-Pos1/0/0] quit
[~RouterC] interface pos 2/0/0
[~RouterC-Pos2/0/0] ospf cost 2
[~RouterC-Pos2/0/0] commit
[~RouterC-Pos2/0/0] quit
[~RouterC] commit
```

NOTE

After the OSPF cost is set to 2 on Router C, Router A selects only Router B as the intermediate router to the network segment 10.2.1.0, and Router C becomes a standby of Router B.

# Check the routing table on RouterA. As shown in the routing table, the route to the network segment 10.1.3.0 is learned through BGP, and the outbound interface is POS 1/0/0.

```
[~RouterA] display ip routing-table
Route Flags: R - relied, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 20       Routes : 20
Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface
        1.1.1.1/32  Direct 0    0        D     127.0.0.1       InLoopBack0
        2.2.2.2/32  OSPF   10   3        D     10.1.1.2        Pos1/0/0
        4.4.4.0/24  BGP    255  0        RD    4.4.4.4         Pos1/0/0
        4.4.4.4/32  OSPF   10   3        D     10.1.1.2        Pos1/0/0
        5.5.5.0/24  BGP    255  0        RD    10.2.1.2        Pos1/0/0
       10.1.1.0/30  Direct 0    0        D     10.1.1.1        Pos1/0/0
       10.1.1.1/32  Direct 0    0        D     127.0.0.1       InLoopBack0
       10.1.1.2/32  Direct 0    0        D     10.1.1.2        Pos1/0/0
       10.1.2.0/30  Direct 0    0       D     10.1.2.1        Pos2/0/0
       10.1.2.1/32  Direct 0    0       D     127.0.0.1       InLoopBack0
       10.1.2.2/32  Direct 0    0       D     10.1.2.2        Pos2/0/0
      127.0.0.0/8   Direct 0    0       D     127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0       D     127.0.0.1       InLoopBack0
       10.1.3.0/30  OSPF   10   2       D     10.1.1.2        Pos1/0/0
       10.1.3.1/32  BGP    255  0       RD    4.4.4.4         Pos1/0/0
       10.1.4.0/30  OSPF   10   3       D     10.1.1.2        Pos1/0/0
                    OSPF   10   3       D     10.1.2.2        Pos2/0/0
       10.1.4.1/32  BGP    255  0       RD    4.4.4.4         Pos1/0/0
       10.2.1.0/30  BGP    255  0       RD    4.4.4.4         Pos1/0/0
       10.2.1.2/32  BGP    255  0       RD    4.4.4.4         Pos1/0/0
       10.3.1.0/30  BGP    255  0       RD    4.4.4.4         Pos1/0/0
```

# Display the routing table on Router B.

```
[~RouterB] display ip routing-table
Route Flags: R - relied, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 19       Routes : 19
Destination/Mask    Proto  Pre  Cost      Flags NextHop         Interface
        2.2.2.2/32  Direct 0    0           D   127.0.0.1        InLoopBack0
        1.1.1.1/32  OSPF   10   2           D   10.1.1.1         Pos1/0/0
        4.4.4.0/24  BGP    255  0           RD  10.1.3.2         Pos2/0/0
        4.4.4.4/32  OSPF   10   2           D   10.1.3.2         Pos2/0/0
        5.5.5.0/24  BGP    255  0           RD  10.2.1.2         Pos2/0/0
       10.1.1.0/30  Direct 0    0           D   10.1.1.2         Pos1/0/0
       10.1.1.1/32  Direct 0    0           D   10.1.1.1         Pos1/0/0
       10.1.1.2/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.2.0/30  OSPF   10   2           D   10.1.1.1         Pos1/0/0
       10.1.3.0/30  Direct 0    0           D   10.1.3.1         Pos2/0/0
       10.1.3.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.3.2/32  Direct 0    0           D   10.1.3.2         Pos2/0/0
      127.0.0.0/8   Direct 0    0           D   127.0.0.1        InLoopBack0
      127.0.0.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.4.0/30  OSPF   10   2           D   10.1.3.2         Pos2/0/0
       10.1.4.1/32  BGP    255  0           RD  10.1.3.2         Pos2/0/0
       10.2.1.0/30  BGP    255  0           RD  10.1.3.2         Pos2/0/0
       10.2.1.2/32  BGP    255  0           RD  10.1.3.2         Pos2/0/0
       10.3.1.0/30  BGP    255  0           RD  10.1.3.2         Pos2/0/0
```

As shown in the routing table, Router B learns the route to the network segment 10.3.1.0 through BGP, and the outbound interface is POS 2/0/0. The routes to the network segments 10.1.2.0 and 10.1.4.0 can be learned through OSPF. The costs of these routes are the same, namely, 2.

Step 6 Enable OSPF-BGP synchronization on Router B.

```
[~RouterB] ospf 1
[~RouterB-ospf-1] stub-router on-startup
[~RouterB-ospf-1] quit
[~RouterB] commit
```

Step 7 Verify the configuration.

# Restart router RouterB.

NOTE

Confirm the action before you use the **reboot** command because the **reboot** command breaks down the network in a short time. In addition, before restarting the router, ensure that the configuration file of the router is saved.

```
<RouterB> reboot
System will reboot! Continue?[Y/N] y
```

# Display the routing table on Router A. As shown in the routing table, the route to the network 10.3.1.0 is learned through BGP, and the outbound interface is POS 2/0/0.

```
[~RouterA] display ip routing-table
```

```
Route Flags: R - relied, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 20       Routes : 20
Destination/Mask    Proto  Pre  Cost      Flags NextHop         Interface
        1.1.1.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
        2.2.2.2/32  OSPF   10   4           D   10.1.2.2         Pos2/0/0
        4.4.4.0/24  BGP    255  0           RD  4.4.4.4          Pos2/0/0
        4.4.4.4/32  OSPF   10   4           D   10.1.2.2         Pos2/0/0
        5.5.5.0/24  BGP    255  0           RD  10.2.1.2         Pos2/0/0
       10.1.1.0/30  Direct 0    0           D   10.1.1.1         Pos1/0/0
       10.1.1.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.1.2/32  Direct 0    0           D   10.1.1.2         Pos1/0/0
       10.1.2.0/30  Direct 0    0           D   10.1.2.1         Pos2/0/0
       10.1.2.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.2.2/32  Direct 0    0           D   10.1.2.2         Pos2/0/0
      127.0.0.0/8   Direct 0    0           D   127.0.0.1        InLoopBack0
      127.0.0.1/32  Direct 0    0           D   127.0.0.1        InLoopBack0
       10.1.3.0/30  OSPF   10   2           D   10.1.1.2         Pos1/0/0
       10.1.3.1/32  BGP    255  0           RD  4.4.4.4          Pos2/0/0
       10.1.4.0/30  OSPF   10   3           D   10.1.2.2         Pos2/0/0
       10.1.4.1/32  BGP    255  0           RD  4.4.4.4          Pos2/0/0
       10.2.1.0/30  BGP    255  0           RD  4.4.4.4          Pos2/0/0
       10.2.1.2/32  BGP    255  0           RD  4.4.4.4          Pos2/0/0
      10.3.1.0/30   BGP    255  0           RD  4.4.4.4          Pos2/0/0
```

# Display the routing table on Router B. As shown in the routing table, only OSPF routes exist in the routing table temporarily and their costs are at least 65535. This is because IGP routes can be converged faster than BGP routes.

```
[~RouterB] display ip routing-table
Route Flags: R - relied, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 15       Routes : 15
Destination/Mask    Proto  Pre  Cost      Flags NextHop        Interface
        1.1.1.1/32  OSPF   10   65536       D  10.1.1.1        Pos1/0/0
        2.2.2.2/32  Direct 0    0           D  127.0.0.1       InLoopBack0
        4.4.4.4/32  OSPF   10   65536       D  10.1.3.2        Pos2/0/0
       10.1.1.0/30  Direct 0    0           D  10.1.1.2        Pos1/0/0
       10.1.1.1/32  Direct 0    0           D  10.1.1.1        Pos1/0/0
       10.1.1.2/32  Direct 0    0           D  127.0.0.1       InLoopBack0
       10.1.2.0/30  OSPF   10   65536       D  10.1.1.1        Pos1/0/0
       10.1.3.0/30  Direct 0    0           D  10.1.3.1        Pos2/0/0
       10.1.3.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
       10.1.3.2/32  Direct 0    0           D  10.1.3.2        Pos2/0/0
      127.0.0.0/8   Direct 0    0           D   127.0.0.1      InLoopBack0
      127.0.0.1/32  Direct 0    0           D   127.0.0.1      InLoopBack0
       10.1.4.0/30  OSPF   10   65536       D  10.1.3.2        Pos2/0/0
      127.0.0.0/8   Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
```

# Display the routing table on Router B again.

```
[~RouterB] display ip routing-table
Route Flags: R - relied, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 19       Routes : 19
Destination/Mask    Proto  Pre  Cost       Flags NextHop       Interface
        2.2.2.2/32  Direct 0    0           D    127.0.0.1        InLoopBack0
        1.1.1.1/32  OSPF   10   2           D    10.1.1.1         Pos1/0/0
        4.4.4.0/24  BGP    255  0           RD   10.1.3.2         Pos2/0/0
        4.4.4.4/32  OSPF   10   2           D    10.1.3.2         Pos2/0/0
        5.5.5.0/24  BGP    255  0           RD   10.2.1.2         Pos2/0/0
       10.1.1.0/30  Direct 0    0           D    10.1.1.2         Pos1/0/0
       10.1.1.1/32  Direct 0    0           D    10.1.1.1         Pos1/0/0
       10.1.1.2/32  Direct 0    0           D    127.0.0.1        InLoopBack0
       10.1.2.0/30  OSPF   10   2           D    10.1.1.1         Pos1/0/0
```

```
      10.1.3.0/30  Direct 0    0           D    10.1.3.1         Pos2/0/0
      10.1.3.1/32  Direct 0    0           D    127.0.0.1        InLoopBack0
      10.1.3.2/32  Direct 0    0           D    10.1.3.2         Pos2/0/0
     127.0.0.0/8   Direct 0    0           D    127.0.0.1        InLoopBack0
     127.0.0.1/32  Direct 0    0           D    127.0.0.1        InLoopBack0
      10.1.4.0/30  OSPF   10   2           D    10.1.3.2         Pos2/0/0
      10.1.4.1/32  BGP    255  0           RD   10.1.3.2         Pos2/0/0
      10.2.1.0/30  BGP    255  0           RD   10.1.3.2         Pos2/0/0
      10.2.1.2/32  BGP    255  0           RD   10.1.3.2         Pos2/0/0
      10.3.1.0/30  BGP    255  0           RD   10.1.3.2         Pos2/0/0
```

As shown in the routing table, after BGP routes on Router B are converged, the routing information is restored to the one that is displayed before the restart.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
router id 1.1.1.1
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.1.1 255.255.255.252
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.2.1 255.255.255.252
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 10
 router-id 1.1.1.1
 peer 2.2.2.2 as-number 10
 peer 2.2.2.2 connect-interface LoopBack 0
 peer 3.3.3.3 as-number 10
 peer 3.3.3.3 connect-interface LoopBack 0
 peer 4.4.4.4 as-number 10
 peer 4.4.4.4 connect-interface LoopBack 0
 #
ipv4-family unicast
  undo synchronization
  peer 4.4.4.4 enable
  peer 10.1.1.2 enable
  peer 10.1.2.2 enable
#
ospf 1
 area 0.0.0.0
  network 1.1.1.1 0.0.0.0
  network 10.1.1.0 0.0.0.3
  network 10.1.2.0 0.0.0.3
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
router id 2.2.2.2
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.1.2 255.255.255.252
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.3.1 255.255.255.252
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 10
 router-id 2.2.2.2
 peer 1.1.1.1 as-number 10
 peer 1.1.1.1 connect-interface LoopBack 0
 peer 3.3.3.3 as-number 10
 peer 3.3.3.3 connect-interface LoopBack 0
 peer 4.4.4.4 as-number 10
 peer 4.4.4.4 connect-interface LoopBack 0
 #
ipv4-family unicast
  undo synchronization
  peer 10.1.1.1 enable
  peer 10.1.3.2 enable
#
ospf 1
 area 0.0.0.0
  network 10.1.1.0 0.0.0.3
  network 10.1.3.0 0.0.0.3
  network 2.2.2.2 0.0.0.0
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
router id 3.3.3.3
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.4.1 255.255.255.252
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.2.2 255.255.255.252
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 10
 router-id 3.3.3.3
 peer 1.1.1.1 as-number 10
 peer 1.1.1.1 connect-interface LoopBack 0
 peer 2.2.2.2 as-number 10
 peer 2.2.2.2 connect-interface LoopBack 0
 peer 4.4.4.4 as-number 10
 peer 4.4.4.4 connect-interface LoopBack 0
 #
ipv4-family unicast
  undo synchronization
  peer 10.1.4.2 enable
  peer 10.1.2.1 enable
#
ospf 1
 area 0.0.0.0
  network 10.1.2.0 0.0.0.3
  network 10.1.4.0 0.0.0.3
  network 3.3.3.3 0.0.0.0
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
router id 4.4.4.4
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.4.2 255.255.255.252
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.1.3.2 255.255.255.252
#
interface Pos3/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.2.1.1 255.255.255.252
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 10
 router-id 4.4.4.4
 peer 10.2.1.2 as-number 20
 peer 1.1.1.1 as-number 10
 peer 1.1.1.1 connect-interface LoopBack 0
 peer 2.2.2.2 as-number 10
 peer 2.2.2.2 connect-interface LoopBack 0
 peer 3.3.3.3 as-number 10
 peer 3.3.3.3 connect-interface LoopBack 0
 #
ipv4-family unicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 2.2.2.2 enable
  peer 1.1.1.1 enable
  peer 5.5.5.5 enable
  peer 3.3.3.3 enable
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 10.1.3.0 0.0.0.3
  network 10.1.4.0 0.0.0.3
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
router id 5.5.5.5
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.2.1.2 255.255.255.252
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ip address 10.3.1.1 255.255.255.252
```

```
#
interface LoopBack0
 ip address 5.5.5.5 255.255.255.255
#
bgp 20
 router-id 5.5.5.5
 peer 10.2.1.1 as-number 10
#
ipv4-family unicast
  undo synchronization
  network 10.3.1.0 255.255.255.252
  peer 10.2.1.1 enable
#
return
```

4 OSPFv3 Configuration

About This Chapter

This chapter describes the OSPFv3 fundamentals and configuration steps for basic OSPFv3 functions, OSPFv3 area features, OSPFv3 routing information and adjusting and optimizing OSPFv3 networks, along with typical examples.

detecting a link failure, BFD notifies the routing protocol of the failure, which triggers fast convergence. When the neighbor relationship is Down, the BFD session is deleted dynamically.

4.9 Configuring OSPFv3 Fast Convergence
To achieve fast convergence, you can adjust the timer.

4.10 Configuring the OSPFv3 GR Helper
As a neighbor of the GR restarter, the GR helper can identify GR signaling. The GR helper maintains its adjacency with the GR restarter when the restarter performs the master/slave switchover, helping the restarter recover the network topology.

4.11 Improving the Stability of an OSPFv3 Network
A stable OSPFv3 network features less route flapping, normal router performance, and good network performance.

4.12 Configuring the Network Management Function of OSPFv3
OSPFv3 supports the network management function. You can configure the OSPFv3 Management Information Base (MIB) and bind it to an OSPFv3 process through the Simple Network Management Protocol (SNMP). In this manner, the OSPFv3 MIB manages multicast information exchanged between the Network Management Station (NMS) and agents.

4.13 Resetting OSPFv3
Restarting OSPFv3 can reset OSPFv3. In addition, you can choose to reset OSPFv3 through GR.

4.14 Configuration Examples
This section provides several configuration examples of OSPFv3 together with the configuration flowchart. The configuration examples explain networking requirements, configuration notes, and the configuration roadmap.

4.1 OSPFv3 Overview

The Open Shortest Path First Version 3.0 (OSPFv3) supports the version 6 of the Internet Protocol (IPv6). OSPFv3 conforms to RFC 2740 (OSPF for IPv6).

OSPFv3 and OSPFv2 have the following in common:

- 32-bit router ID, area ID, and Link State Advertisement (LSA) link-state ID
- Five types of packets such as Hello, Database Description (DD), Link State Request (LSR), Link State Update (LSU), and Link State Acknowledgement (LSAck) packets
- Neighbor discovery and adjacency establishment mechanisms
- Flooding and aging mechanisms of LSAs
- LSA types

OSPFv3 and OSPFv2 differ as follows:

- OSPFv3 runs based on a link; OSPFv2 runs based on a network segment.
- OSPFv3 can run multiple instances on the same link.
- The topology of OSPFv3 is independent of IPv6 address prefixes.
- OSPFv3 identifies its neighbors with the IPv6 link-local addresses.
- OSPFv3 has three new types of LSA flooding scopes.

4.2 OSPFv3 Features Supported by the NE5000E

The NE5000E supports OSPFv3 multi-process and OSPFv3 GR.

At present, the NE5000E supports the following OSPFv3 features:

- Basic features stipulated in RFC 2740
- OSPFv3 stub areas
- OSPFv3 multi-process, namely, running of multiple OSPFv3 processes on one router
- OSPFv3 GR
  - If a router restarts or performs the master/slave switchover, it directly ages all the entries in the Forwarding Information Base (FIB). This interrupts forwarding. The neighboring routers remove the router from the neighbor list and inform other routers of the router failure. Then, SPF needs to be calculated again. If the router recovers after a short period of time, the neighbor relationship becomes unstable. This results in route flapping.
  - If a router restarts because of anomalies, you can enable OSPFv3 Graceful Restart (GR) to avoid service interruption during the restart of the router.
  - IP FRR

    With OSPFv3 IP FRR, devices can rapidly switch traffic from faulty links to backup links without interrupting traffic. This protects traffic and greatly improves the reliability of OSPFv3 networks.
  - BFD for OSPFv3

    BFD keeps track of liveliness of network links and detects any fault in the links much faster than the normal keep-alive protocols. When OSPFv3 is associated with BFD sessions, link failures are notified immediately to OSPFv3 by BFD and OSPFv3 can take actions to perform route calculation and converge in the new network topology.

- OSPFv3 Fast Convergence

  To achieve fast convergence, you can adjust the timer.

4.3 Configuring Basic OSPFv3 Functions

You need to enable OSPFv3 and specify interfaces and area IDs before configuring other functions.

Applicable Environment

Enable the OSPFv3 process and specify its router ID before configuring OSPFv3; otherwise, other functions cannot take effect.

You must enable OSPFv3 and specify the interface and area ID before configuring other functions. OSPFv3 configurations, however, are independent of interface-related features.

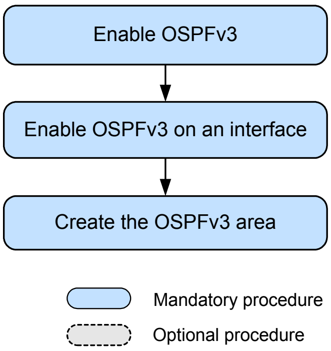
Pre-configuration Tasks

Before configuring basic OSPFv3 functions, complete the following tasks:

- Making the adjacent nodes accessible at the network layer
- Enabling IPv6 capabilities

Configuration Procedures

Figure 4-1 Flowchart of configuring basic OSPFv3 functions



Related Tasks

4.14.1 Example for Configuring Basic OSPFv3 Functions

4.3.1 Enabling OSPFv3

Creating an OSPFv3 process is a prerequisite for configuring OSPFv3 features. By creating an OSPFv3 process, you can manually specify the router ID for an OSPFv3 process.

Context

OSPFv3 supports multiple processes. Multiple OSPFv3 processes running on one router are differentiated by process IDs. An OSPFv3 process ID is set when OSPFv3 is enabled and is only locally valid. It does not affect the packet exchange with other routers.

In the format of an IPv4 address, a router ID is a 32-bit unsigned integer that uniquely identifies one router within an AS. The router ID of OSPFv3 must be manually set. If no router ID is set, OSPFv3 fails to run normally.

When manually setting the router ID, ensure that the router IDs of any two routers in an AS are different. When multiple processes are enabled on the same router, it is necessary to specify a unique router ID for each process.

To ensure the stable running of OSPFv3, you need to allocate router IDs and set them in network planning.

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id] [vpn-instance vpn-instance-name]
```

OSPFv3 is enabled and the OSPFv3 view is displayed.

Step 3 Run:

```
router-id router-id
```

A Router ID is set.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.3.2 Enabling OSPFv3 on an Interface

For an interface with multiple instances, you need to specify which instance on the interface is enabled in the OSPFv3 process when enabling OSPFv3 on the interface.

Context

After enabling OSPFv3 in the system view, you need to enable OSPFv3 on the interface.

Because an interface may have multiple instances, you need to specify which instance of the interface is enabled in the OSPFv3 process when OSPFv3 is enabled on the interface. If no instance ID is specified, the value defaults to 0. The same instance must be enabled on the interfaces between which the neighbor relationship is set up.

Do as follows on the router that runs OSPFv3.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 (Optional) Run the **ospfv3 network-type** { **broadcast** | **nbma** | **p2mp** [**non-broadcast**] | **p2p** } [**instance** *instance-id*] command to configure the network type of an interface.

When an interface supports multi-instances, you must specify the value of *instance-id* when enabling OSPFv3 on the interface. If the value of *instance-id* is not specified, the default value 0 is adopted. In this case, the configured network type of an interface mismatches the actual network type of the interface. This step is mandatory in such a case.

Step 4 Run:

```
ospfv3 process-id area area-id [ instance instance-id ]
```

OSPFv3 is enabled on the interface.

The area ID can be a decimal integer or in the IPv4 address format, but it is displayed in the IPv4 address format.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

4.3.3 Creating an OSPFv3 Area

After an AS is divided into different areas, and OSPFv3 interfaces and areas to which these interfaces belong are specified, OSPFv3 can discover and calculate routes in the AS.

Context

When configuring the OSPFv3 devices in the same OSPFv3 area, note that most data configurations should be performed based on the area. Otherwise, the neighboring routers cannot exchange information with each other, which may even result in the congestion of routing information or loops.

Do as follows on each router that needs to run OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
area area-id
```

The OSPFv3 area view is displayed.

The area ID can be a decimal integer or in the IPv4 address format, but it is displayed in the IPv4 address format.

An OSPFv3 area cannot be deleted directly. Only after all the configurations in the area view are removed and the status of the interfaces in this area become Down, this area is automatically deleted.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.3.4 Checking the Configuration

After the basic OSPFv3 functions are configured, you can run commands to view information about neighbors, interfaces, and the OSPFv3 routing table.

Prerequisite

The configurations of basic OSPFv3 functions are complete.

Procedure

- Run the **display ospfv3** [*process-id*] [**area** *area-id*] **peer** [*interface-type interface-number*] [**verbose**] command in any view to view information about OSPFv3 neighbors.
- Run the **display ospfv3** [*process-id*] [**area** *area-id*] [*interface-type interface-number*] command in any view to view information about an OSPFv3 interface.
- Run the **display ospfv3** [*process-id*] **routing** command in any view to view information about an OSPFv3 routing table.

----End

Example

Run the **display ospfv3 peer** command, and you can view the router ID, interface priority, and interface status on the OSPF neighbor. For example:

```
<HUAWEI> display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface     Instance ID
1.1.1.1           1   Full/ -          00:00:30    GE1/0/0            0
```

Run the **display ospfv3 interface** command, and you can view the network types and status of OSPF interfaces. For example:

```
<HUAWEI> display ospfv3 interface
GE1/0/0 is up, line protocol is up
  Interface ID 0x102
  Interface MTU 1500
  IPv6 Prefixes
    FE80::2E0:FFF:FE4E:F101 (Link-Local Address)
    2000::1/64
  Interface Event: 1
  Interface Lsa Count: 1
  Interface Lsa Checksum: 0x9b74
  OSPFv3 Process (1), Area 0.0.0.1, Instance ID 0
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State Full, Priority 1
    No designated router on this link
    No backup designated router on this link
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
```

Run the **display ospfv3 routing** command, and you can view the destination addresses, link costs, and next hops of OSPFv3 routes. For example:

```
<HUAWEI> display ospfv3 routing
    Destination                       Metric
     Next-hop
       Backup Next-hop
    45::/64                             10
      via FE80::2000:10FF:FE03:4, Ethernet3/0/0
       backup via FE80::2000:10FF:FE03:4, Pos1/0/1, LFA LINK-NODE
```

4.4 Configuring OSPFv3 Attributes in Different Types of Networks

By setting network types for OSPFv3 interfaces and adjusting OSPFv3 attributes, you can build OSPFv3 networks flexibly.

Applicable Environment

Based on the types of link layer protocols, OSPFv3 classifies networks into the following types:

- P2MP: There is no concept of P2MP in link layer protocols. Therefore, a P2MP network must be forcibly changed from other network types.
- NBMA: If the link layer protocol is FR, ATM, or X.25, OSPFv3 defaults the network type to NBMA.
- Broadcast: If the link layer protocol is GigabitEthernet or FDDI, OSPFv3 defaults the network type to broadcast.
- P2P: If the link layer protocol is PPP, HDLC, or LAPB, OSPFv3 defaults the network type to P2P.

If link layer protocols remain unchanged, you can change network types and configure OSPFv3 features to flexibly build networks.

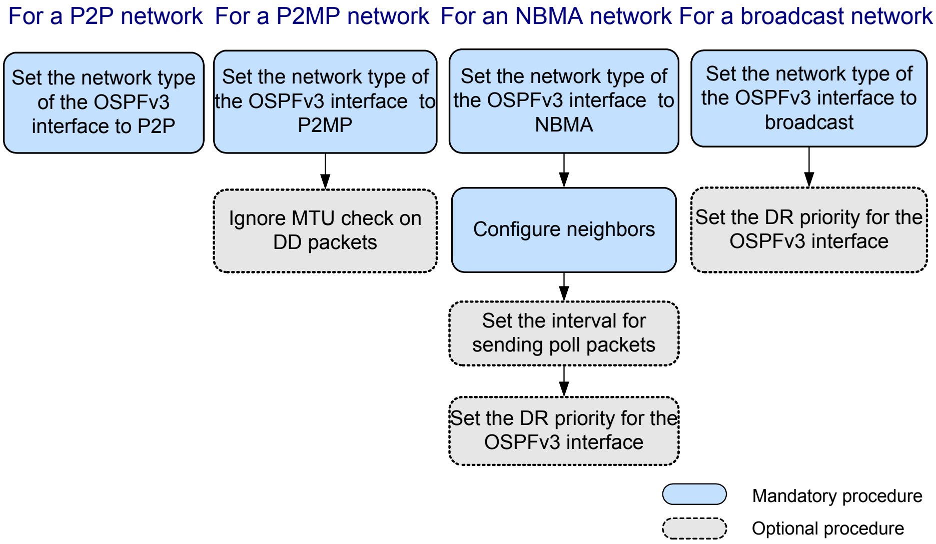
Pre-configuration Tasks

Before configuring OSPFv3 attributes in different types of networks, complete the following tasks:

- Configuring a link layer protocol
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPFv3 Functions**

Configuration Procedures

Figure 4-2 Flowchart of configuring OSPFv3 attributes in different types of networks


For a P2P network
For a P2MP network
For an NBMA network
For a broadcast network
Set the network type of the OSPFv3 interface to P2P
Set the network type of the OSPFv3 interface to P2MP
Set the network type of the OSPFv3 interface to NBMA
Set the network type of the OSPFv3 interface to broadcast
Ignore MTU check on DD packets
Configure neighbors
Set the DR priority for the OSPFv3 interface
Set the interval for sending poll packets
Set the DR priority for the OSPFv3 interface
Mandatory procedure
Optional procedure

4.4.1 Configuring Network Types of OSPFv3 Interfaces

OSPFv3 classifies networks into broadcast networks, P2P networks, P2MP networks, and NBMA networks based on link layer protocols. By configuring network types for interfaces, you can change the network types of interfaces.

Context

By default, the network type of an interface is determined by the physical interface. The network type of an Ethernet interface is **broadcast**, that of a serial interface or a POS interface (encapsulated with PPP or HDLC) is **p2p**, and that of an ATM interface or a Frame-relay interface is **nbma**.

Do as follows on the OSPFv3 router:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 network-type { broadcast | nbma | p2mp | p2p }[ instance instance-id ]
```

Network types are configured for OSPFv3 interfaces.

NOTE

Generally, the network types of two OSPFv3 interfaces on the both ends of the link must be identical. Otherwise, the two interfaces cannot set up the neighbor relationship.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.4.2 (Optional) Setting the DR Priority for the OSPFv3 Interface of the Broadcast or NBMA Network Type

When configuring a broadcast network or an NBMA network, you can specify the DR priority for each interface to change the results of DR/BDR election on the network.

Context

When configuring broadcast networks or NBMA networks, you can specify the DR priority for each interface to affect the DR/BDR election in the network. The greater the value is, the higher the priority is.

By default, the priority of the interface that candidates for the DR is 1.

Do as follows on the OSPFv3 router:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 dr-priority priority [ instance instance-id ]
```

The DR priority of the OSPFv3 interface is set.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

CAUTION

Restarting or shutting down an interface will interrupt the OSPFv3 adjacency between devices. Therefore, perform the operation with caution.

Reconfiguring the DR priority for a device does not change the DR or BDR on a network. You can re-elect a DR or BDR by using the following methods. This, however, will result in the interruption of the OSPFv3 adjacency between devices. Therefore, the following methods are used only when necessary.

- Restart the OSPFv3 processes on all the routers.
- Run the **shutdown** and then **undo shutdown** commands on the interfaces where the OSPFv3 adjacency is established.

4.4.3 (Optional) Configuring the Interval for Sending Poll Packets in NBMA Networks

On an NBMA network, after a neighbor becomes invalid, a device sends Hello packets to the neighbor at the set polling interval.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPFv3 interface view is displayed.

Step 3 Run:

```
ospfv3 timer poll interval [ instance instance-id ]
```

The interval for sending Poll packets on the NBMA interface is set.

The parameter **poll** *interval* specifies the polling interval for sending Hello packets.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.4.4 (Optional) Ignoring the MTU Check on DD Packets

By disabling an interface from checking the MTU field in the received DD packet, you can enable an OSPFv3 device to receive the packet with the MTU field being 0.

Context

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 mtu-ignore [ instance instance-id ]
```

The MTU check on DD packets is ignored.

After the command is used, the interface does not check the MTU field of a received DD packet.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.4.5 Checking the Configuration

After the attributes of OSPFv3 interfaces in different types of networks are configured, you van view information about OSPFv3 interfaces.

Prerequisite

The configurations of OSPFv3 interface attributes in different types of network are complete.

Procedure

Step 1 Run the **display ospfv3** [*process-id*] [**area** *area-id*] [*interface-type interface-number*] command in any view to view information about an OSPFv3 interface.

----End

Example

Run the **display ospfv3 interface** command, and you can view the network types of OSPFv3 interfaces. For example:

```
<HUAWEI> display ospfv3 interface
GE1/0/0 is up, line protocol is up
  Interface ID 0x102
  Interface MTU 1500
  IPv6 Prefixes
    FE80::2E0:FFF:FE4E:F101 (Link-Local Address)
    2000::1/64
  Interface Event: 1
  Interface Lsa Count: 1
  Interface Lsa Checksum: 0x9b74
  OSPFv3 Process (1), Area 0.0.0.1, Instance ID 0
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State Full, Priority 1
    No designated router on this link
    No backup designated router on this link
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
```

4.5 Configuring OSPFv3 Route Attributes

By setting OSPFv3 route attributes, you can change OSPFv3 routing policies to meet the requirements of complex networks.

Applicable Environment

In actual applications, to meet the requirements of a complicated networking environment, you can change OSPFv3 routing policies by configuring OSPFv3 route attributes. Through the following procedures, you can:

- Set the cost on the OSPFv3 interface.
- Configure load balancing among equal-cost routes.

Pre-configuration Tasks

Before configuring OSPFv3 route attributes, complete the following tasks:

- Configuring IP addresses of interfaces to make neighboring nodes reachable
- **Configuring basic OSPFv3 functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except "Checking the Configuration") according to the applicable environment.

4.5.1 Setting the Link Cost on the OSPFv3 Interface

OSPFv3 can automatically calculate the link cost for an interface based on the interface bandwidth. You can also set the link cost for the interface by using the relevant command.

Context

You can control the route cost by setting the link cost of OSPFv3 on different interfaces.

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 cost cost [ instance instance-id ]
```

The link cost is set on the OSPFv3 interface.

By default, the link cost on an OSPFv3 interface is 1.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.5.2 Setting the Maximum Number of Equal-Cost Routes

If the destinations and costs of the multiple routes discovered by one routing protocol are the same, load balancing can be performed among these routes.

Context

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
maximum load-balancing number
```

The maximum number of equal-cost routes is set.

NOTE

The range and default value of the number of equal-cost routes may vary with products and protocols. You can adjust the range and default value of the number of equal-cost routes after purchasing the license.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.5.3 Checking the Configuration

After setting OSPF route attributes, you can view information about OSPFv3 interfaces and the routing table.

Prerequisite

The configurations of OSPFv3 route attributes are complete.

Procedure

- Run the **display ospfv3** [*process-id*] [**area** *area-id*] [*interface-type interface-number*] command in any view to view information about an OSPFv3 interface.
- Run the **display ospfv3** [*process-id*] **routing** command in any view to view information about an OSPFv3 routing table.

----End

Example

Run the **display ospfv3 interface** command, and you can view priorities of OSPFv3 interfaces. For example:

```
<HUAWEI> display ospfv3 interface
GE1/0/0 is up, line protocol is up
  Interface ID 0x102
  Interface MTU 1500
  IPv6 Prefixes
    FE80::2E0:FFF:FE4E:F101 (Link-Local Address)
    2000::1/64
  Interface Event: 1
  Interface Lsa Count: 1
  Interface Lsa Checksum: 0x9b74
  OSPFv3 Process (1), Area 0.0.0.1, Instance ID 0
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State Full, Priority 1
    No designated router on this link
    No backup designated router on this link
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
```

Run the **display ospfv3 routing** command, and you can view the destination addresses, link costs, and next hops of OSPFv3 routes. For example:

```
<HUAWEI> display ospfv3 routing

    Destination                              Metric
     Next-hop
       Backup Next-hop
    45::/64                                  10
      via FE80::2000:10FF:FE03:4, Ethernet3/0/0
       backup via FE80::2000:10FF:FE03:4, Pos1/0/1, LFA LINK-NODE
```

4.6 Controlling OSPFv3 Routing Information

Before controlling OSPFv3 routing information, familiarize yourself with the applicable environment, complete pre-configuration tasks, and obtain the required data. This can help you complete the configuration task quickly and accurately.

Applicable Environment

Through the configuration in this section, you can control the advertising and receiving of OSPFv3 routing information and configure OSPFv3 to import external routes.

Routing information control is classified into the following types:

- Control of routes outside the local OSPFv3 area

  This function can be configured on any router that runs OSPFv3. For example, you can enable the router to filter the imported routes and set the maximum number of equal-cost routes.

- Control of routes inside the local OSPFv3 area

  This function can be configured only on ABRs. For example, when an ABR has multiple routes to a destination in the local area, you can configure the ABR to aggregate routes by configuring OSPFv3 route aggregation in the local area. In this manner, only an aggregate LSA is sent to the backbone area.

Pre-configuration Tasks

Before controlling OSPFv3 routing information, complete the following tasks:

- Enabling IPv6 capabilities
- **Configuring basic OSPFv3 functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except "Checking the Configuration") according to the applicable environment.

4.6.1 Configuring OSPFv3 to Import External Routes

Importing the routes discovered by other routing protocols can enrich OSPFv3 routing information.

Context

OSPFv3 is a link state-based routing protocol and hence cannot directly filter advertised LSAs. OSPFv3 must filter the routes when importing them. Then, only the routes that pass the filtering can be advertised.

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
default { cost cost | tag tag | type type } *
```

The default cost of the imported route is set.

Step 4 Run:

```
import-route protocol [ process-id ] [ cost cost | type type | tag tag |route-policy route-policy-name ] *
```

External routes are imported.

NOTE

The **import-route** command cannot be used to import the default route from another AS.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

4.6.2 Configuring OSPFv3 to Advertise the Default Route to the OSPFv3 Area

Configure OSPFv3 to advertise the default route to the OSPFv3 area, only the routes that pass the filtering can be advertised.

Context

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
default-route-advertise [ always | cost cost | type type | tag tag | route-policy route-policy-name ] *
```

Default routes are advertised to the OSPFv3 route area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.6.3 Configuring OSPFv3 to Filter the Received Routes

After receiving LSAs, OSPFv3 determines whether to add the calculated routes to the local routing table based on the filtering policy.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
filter-policy { acl6-number | acl6-name acl6-name | ipv6-prefix ipv6-prefix-name }
import
```

OSPFv3 is configured to filter the imported routes.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.6.4 Configuring OSPFv3 to Filter the Routes to Be Advertised

You can set filtering conditions for the imported routes. Only the routes that meet the conditions can be advertised.

Context

When OSPFv3 receives LSAs, it can filter the imported routes based on certain conditions before advertising them to neighbors.

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 process-id
```

The OSPFv3 view is displayed.

Step 3 (Optional) Run:

```
default-route-advertise [ always | cost cost | type type | tag tag | route-policy
route-policy-name ] *
```

Default routes are advertised in the OSPFv3 area.

Step 4 Run:

filter-policy { *acl6-number* | **acl6-name** *acl6-name* | **ipv6-prefix** *ipv6-prefix-name* } **export** [*protocol* [*process-id*]]

The routes imported with the **import-route** command are filtered, and only the filtered routes can be advertised.

- *acl-number*: specifies the number of the basic ACL.
- **acl-name** *acl-name*: specifies the name of the ACL.
- **ip-prefix** *ip-prefix-name*: specifies the name of the IP prefix list.

You can specify the parameter *protocol* [*process-id*] in the command so that OSPFv3 will filter the routes of a certain routing protocol or a certain OSPF process. If *protocol* [*process-id*] is not specified, OSPFv3 filters all imported routes.

Step 5 Run:

commit

The configuration is committed.

----End

4.6.5 Checking the Configuration

After controlling OSPFv3 routing information is configured, you can view the OSPFv3 LSDB.

Prerequisite

The configurations of controlling OSPFv3 routing information are complete.

Procedure

- Run the **display ospfv3** [*process-id*] **lsdb** command to view the OSPFv3 LSDB.

----End

Example

Run the **display ospfv3 lsdb** command, and you can view the Link State ID in the header of each LSA and information about the routers that generate or advertise the LSAs. For example:

```
<HUAWEI> display ospfv3 lsdb
               Inter-Area-Prefix-LSA (Area 0.0.0.0)

Link State ID   Origin Router    Age   Seq#       CkSum
0.0.0.1         1.1.1.1          0238  0x80000003 0xbfb1
0.0.0.2         1.1.1.1          0238  0x80000003 0xcfe8
0.0.0.3         1.1.1.1          0238  0x80000003 0xff6b
0.0.0.4         1.1.1.1          0025  0x80000001 0x445e
0.0.0.1         2.2.2.2          0121  0x80000003 0xef7a
0.0.0.2         2.2.2.2          0121  0x80000003 0xb103
0.0.0.3         2.2.2.2          0121  0x80000003 0x93d6


               Inter-Area-Router-LSA (Area 0.0.0.0)

Link State ID   Origin Router    Age   Seq#       CkSum
1.1.1.1         2.2.2.2          0028  0x80000001 0xe530
```

```
                       AS-External-LSA

Link State ID   Origin Router    Age   Seq#       CkSum  Type
0.0.0.1         1.1.1.1          0005  0x80000001 0xac60 E2
```

4.7 Configuring OSPFv3 IP FRR

With OSPFv3 IP FRR, devices can rapidly switch traffic from faulty links to backup links without interrupting traffic. This protects traffic and greatly improves the reliability of OSPFv3 networks.

Applicable Environment

With the development of networks, Voice over IP (VoIP) and on-line video services require high-quality real-time transmission. Nevertheless, if an OSPFv3 fault occurs, traffic can be switched to a new link after far more than 50 ms, which does not meet the requirement for real-time services on the network.

Normally, traffic can be switched to a new link after the following processes: fault detection in milliseconds, notifying the fault to the routing control plane in milliseconds, generating and flooding new topology information in tens of milliseconds, triggering SPF calculation in tens of milliseconds, and notifying and installing a new route in hundreds of milliseconds. As a result, all the processes take far more than 50 milliseconds.

With OSPFv3 IP FRR that calculates a backup link in advance, devices can rapidly switch traffic to the backup link without interrupting services when the primary link becomes faulty. This protects traffic and thus greatly improves the reliability of OSPFv3 networks.

OSPFv3 IP FRR is applicable to services that are sensitive to the packet delay and packet loss.

OSPFv3 IP FRR requires the lower layer to rapidly respond to the link change so that traffic can be rapidly switched to the backup link in the case of a link failure. You can associate FRR and BFD. In this manner, a link fault can be quickly detected, which wins time for the ensuing traffic switchover.

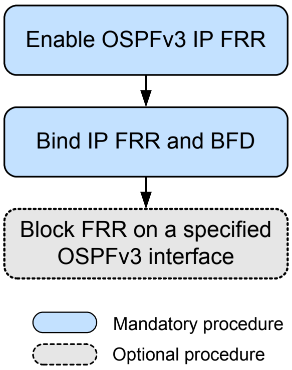
Pre-configuration Tasks

Before configuring OSPFv3 IP FRR, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic OSPFv3 Functions**

Configuration Procedures

Figure 4-3 Flowchart of configuring OSPFv3 IP FRR



Related Tasks

4.14.3 Example for Configuring OSPFv3 IP FRR

4.7.1 Enabling OSPFv3 IP FRR

Loop Free Alternate (LFA) is one of the techniques used to implement basic FRR functions. An LFA-enabled device can generate the loop-free backup link.

Context

LFA is one of the techniques used to implement FRR. Devices can generate loop-free backup links only after LFA is configured in the FRR view.

Do as follows on the router where protection is required for the traffic to be forwarded:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id | router-id router-id | vpn-instance vpn-instance-name ] *
```

An OSPFv3 process is enabled, and the OSPFv3 view is displayed.

Step 3 Run:

```
frr
```

The OSPFv3 IP FRR view is displayed.

Step 4 Run:

```
loop-free-alternate
```

LFA is enabled so that the device can generate a loop-free backup link.

NOTE

OSPFv3 can generate the loop-free backup link only when the OSPFv3 IP FRR traffic protection inequality is met.

----End

4.7.2 Associating IP FRR and BFD

After IP FRR and BFD are associated, the lower layer can rapidly respond to the link change so that traffic can be rapidly switched to the backup link in the case of a link failure.

Context

OSPFv3 IP FRR requires the lower layer to fast respond to the link change so that traffic can be rapidly switched to the backup link in the case of a link failure. After the **bfd all-interfaces frr-binding** command is run to bind the BFD status to the link status of an interface, link failures can be detected rapidly. This ensures that traffic is rapidly switched to the backup link in the case of link failures.

Do as follows on the router where IP FRR and BFD need to be associated:

Procedure

- Associate IP FRR and BFD in an OSPFv3 process.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ospfv3
     ```
     An OSPFv3 process is enabled, and the OSPFv3 view is displayed.
  3. Run:
     ```
     bfd all-interfaces frr-binding
     ```
     IP FRR and BFD are associated in the OSPFv3 process.
- Associate IP FRR and BFD on a specified OSPFv3 interface.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     An interface view is displayed.
  3. Run:
     ```
     ospfv3 bfd frr-binding
     ```
     IP FRR and BFD are associated on the interface.

----End

Follow-up Procedure

The priority of BFD configured on an interface is higher than that of BFD configured in an OSPFv3 process. If BFD is enabled on an interface, BFD sessions are established using the BFD parameters set on the interface.

4.7.3 (Optional) Blocking FRR on an OSPFv3 Interface

If FRR is not required on certain OSPFv3 interfaces, FRR needs to be blocked on these interfaces.

Context

You can run the following commands to prevent a device bearing important services from being a node on a backup link so that the services of the device will not be affected after FRR calculation.

Do as follows on the interfaces of the device where OSPFv3 IP FRR has been configured:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of the OSPFv3 interface running FRR is displayed.

Step 3 Run:

```
ospfv3 frr block
```

FRR is blocked on the OSPFv3 interface.

----End

4.7.4 Checking the Configuration

After OSPFv3 IP FRR is configured, you can view information about the primary link and backup link.

Prerequisite

The configurations of OSPFv3 IP FRR are complete.

Procedure

- Run the **display ospfv3** [*process-id*] **routing** command to view the primary link and backup link after OSPFv3 IP FRR is enabled.

----End

Example

Run the **display ospfv3** [*process-id*] **routing** command on the router running OSPFv3, and you can view information about the backup next hop. For example:

```
<HUAWEI> display ospfv3 routing 10.1.1.1
         OSPFv3 Process 1 with Router ID 1.1.1.1

 Destination      : 10.1.1.0/24
 AdverRouter      : 10.1.1.1            Area            : 0.0.0.0
 Cost             : 1                   Type            : Transit
 NextHop          : 10.1.1.2            Interface       : GE1/0/0
 Priority         : High                Age             : 17h03m33s
 Backup NextHop : 10.1.1.3          Backup Interface : GE1/0/1
 Backup Type    : LFA LINK
```

4.8 Configuring BFD for OPSFv3

If there are high requirements for data transmission, and OSPFv3 convergence needs to be speeded up when the link status changes, you can configure BFD on OSPFv3 links. After detecting a link failure, BFD notifies the routing protocol of the failure, which triggers fast convergence. When the neighbor relationship is Down, the BFD session is deleted dynamically.

Applicable Environment

To increase the convergence speed of OSPFv3 when the link status changes, you can configure BFD on OSPFv3 links.

BFD keeps track of liveliness of network links and detects any fault in the links much faster than the normal keep-alive protocols. When OSPFv3 is associated with BFD sessions, link failures are notified immediately to OSPFv3 by BFD and OSPFv3 can take actions to perform route calculation and converge in the new network topology.

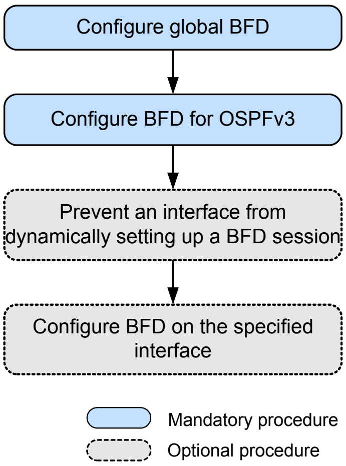
Pre-configuration Tasks

Before configuring BFD for OSPFv3, complete the following tasks:

- Configuring a link layer protocol
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring basic OSPFv3 functions**

Configuration Procedures

Figure 4-4 Flowchart of configuring BFD for OSPFv3


Configure global BFD
Configure BFD for OSPFv3
Prevent an interface from dynamically setting up a BFD session
Configure BFD on the specified interface
Mandatory procedure
Optional procedure

Related Tasks

4.14.4 Example for Configuring BFD for OSPFv3

4.8.1 Configuring Global BFD

On the two devices that need to establish a BFD session, you can configure BFD for all the interfaces in a certain OSPFv3 process.

Context

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 process-id
```

The OSPFv3 view is displayed.

Step 3 Run:

```
bfd all-interfaces enable
```

BFD for OSPFv3 is enabled to establish a BFD session.

By default, BFD is disabled in an OSPFv3 process.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.8.2 Configuring BFD for OSPFv3

After enabling BFD for OSPFv3, you need to configure BFD parameters in the OSPFv3 process.

Context

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 process-id
```

The OSPFv3 view is displayed.

Step 3 Run:

```
bfd all-interfaces { min-transmit-interval min-transmit-value | min-receive-interval min-receive-value | detect-multiplier detect-multiplier-value } *
```

OSPFv3 BFD parameters are configured.

By default, BFD parameters are not configured in an OSPFv3 process.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.8.3 (Optional) Preventing an Interface from Dynamically Setting Up a BFD Session

If you do not want certain OSPFv3 interfaces to set up dynamic BFD sessions, you can disable these interfaces from dynamically setting up BFD sessions.

Context

After the **bfd all-interfaces enable** command is used in an OSPFv3 process, the following situations occur:

- In a P2P network, all OSPFv3 interfaces whose neighbor status is Up set up dynamic BFD sessions.
- In a broadcast network, all OSPFv3 interfaces whose neighbor status is Up set up dynamic sessions between DISs and non-DISs.

If you do not want certain interfaces to set up dynamic BFD sessions, do as follows on the interfaces:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 bfd block
```

The interface is prevented from dynamically setting up a BFD session.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.8.4 (Optional) Configuring BFD on the Specified Interface

To configure BFD only on certain interfaces and not to enable OSPFv3 BFD globally, or to require certain interfaces to rapidly detect link failures after configuring OSPFv3 BFD on them, you can configure BFD on the specified interface.

Context

To configure BFD on the specified interface and not to enable OSPFv3 BFD, or to require the interface to rapidly detect link faults after configuring OSPFv3 BFD on the interface, do as follows on the interface:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 bfd enable
```

BFD is enabled on the interface to establish a BFD session.

When BFD is configured globally and the neighbor status is Full, the default values of BFD parameters used to establish a BFD session are set.

You can run the **ospfv3 bfd** { **min-transmit-interval** *min-transmit-value* | **min-receive-interval** *min-receive-value* | **detect-multiplier** *detect-multiplier-value* } * [**instance** *instance-id*] } command to specify the value for each parameter used to establish a BFD session.

NOTE

- The priority of BFD configured on an interface is higher than that of BFD configured in a process. That is, if BFD is enabled on an interface, the parameters of the interface are used to establish BFD sessions.
- If the parameters of a BFD session are set but the **ospfv3 bfd enable** command is not run, BFD cannot be enabled.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.8.5 Checking the Configuration

After BFD for OSPFv3 is configured, you can check information about the BFD session.

Prerequisite

The configurations of BFD for OSPFv3 are complete.

Procedure

- Run the **display ospfv3** [*process-id*] **bfd session** [**interface** *interface-type interface-number*] [*neighbor-id*] [**verbose**] command to check information about the BFD session.

----End

Example

Run the **display ospfv3 bfd session verbose** command. If a BFD session is correctly established, you can see that the BFD status of the local router is Up. For example:

```
<HUAWEI> display ospfv3 1 bfd session verbose

        OSPFv3 Process 1 with Router ID 1.1.1.1

* - STALE

 Neighbor-Id: 3.3.3.3            BFD Status: Up
 Interface: Eth1/0/0               Instance : 2
 IPv6-Local-Address: FE80::2E0:B1FF:FE49:8142
 IPv6-Remote-Address: FE80:23:22::

   Total UP/DOWN BFD Session Number : 1 / 0
```

4.9 Configuring OSPFv3 Fast Convergence

To achieve fast convergence, you can adjust the timer.

Pre-configuration Tasks

Before configuring OSPFv3 fast convergence, complete the following tasks:

- Configuring a link layer protocol
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring basic OSPFv3 functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

4.9.1 Setting the Interval for Sending Hello Packets

By adjusting the Hello packet transmission interval set on OSPFv3 neighbors, you can change the speed of establishing the neighbor relationship, thus changing the speed of network convergence.

Context

Hello packets are periodically sent to the neighbor router to detect and maintain the neighbor relationship and to elect the DR and the BDR. RFC 2328 requires that the Hello timer values of neighbors be consistent. The value of the Hello timer is inversely proportional to the route convergence speed and network load.

Do as follows on the router that runs OSPFv3:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 timer hello interval [ instance instance-id ]
```

The interval for sending Hello packets is set on the interface.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.9.2 Setting the Dead Time of the Neighbor Relationship

If a router does not receive a Hello packet from its neighbor within the Holddown time, the device considers the neighbor relationship invalid.

Context

Do as follows on the OSPFv3 router:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ospfv3 timer dead interval [ instance instance-id ]
```

The dead time during which the neighbor relationship becomes invalid is set.

By default, the dead interval on the same interface should be four times the interval for sending Hello packets.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.9.3 Checking the Configuration

After OSPFv3 fast convergence is configured, you can view brief information about OSPFv3.

Prerequisite

The configurations of OSPFv3 fast convergence are complete.

Procedure

- Run the **display ospfv3** [*process-id*] [**area** *area-id*] [*interface-type interface-number*] command in any view to view information about an OSPFv3 interface.

----End

Example

Run the **display ospfv3 interface** command, and you can view detailed information about OSPFv3 timers. For example:

```
<HUAWEI> display ospfv3 interface
GE1/0/0 is up, line protocol is up
  Interface ID 0x102
  Interface MTU 1500
  IPv6 Prefixes
    FE80::2E0:FFF:FE4E:F101 (Link-Local Address)
    2000::1/64
  Interface Event: 1
  Interface Lsa Count: 1
  Interface Lsa Checksum: 0x9b74
  OSPFv3 Process (1), Area 0.0.0.1, Instance ID 0
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State Full, Priority 1
    No designated router on this link
    No backup designated router on this link
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
```

4.10 Configuring the OSPFv3 GR Helper

As a neighbor of the GR restarter, the GR helper can identify GR signaling. The GR helper maintains its adjacency with the GR restarter when the restarter performs the master/slave switchover, helping the restarter recover the network topology.

Context

GR is a technology used to ensure normal traffic forwarding and non-stop forwarding of key services during the restart of routing protocols. GR is one of high availability (HA) technologies. HA technologies comprise a set of comprehensive techniques, such as fault-tolerant redundancy, link protection, faulty node recovery, and traffic engineering. As a fault-tolerant redundancy technology, GR is widely used to ensure non-stop forwarding of key services during the master/slave switchover and system upgrade.

NOTE

The NE5000E can be configured as a GR helper rather than a GR restarter.

Pre-configuration Tasks

Before configuring OSPFv3 GR, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring basic OSPFv3 functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 process-id
```

The OSPFv3 view is displayed.

Step 3 Run:

```
helper-role [ { ip-prefix ip-prefix-name | acl-number acl-number | acl-name acl-
name } | max-grace-period period | planned-only | lsa-checking-ignore ] *
```

OSPFv3 GR is enabled.

By default, the OSPFv3 GR helper function is disabled.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following command to check the preceding configuration:

- Run the **display ospfv3** [*process-id*] **graceful-restart-information** command to view the OSPFv3 GR status.

```
          OSPFv3 Router with ID (0.0.0.0) (Process 1)

 Graceful-restart capability      : enabled
 Graceful-restart support         : planned and unplanned, strict lsa check
 Grace-Period Configured          : 120 Sec
 Last Restart-exit Reason         : none

 Helper capability                : enabled
 Helper support                   : planned and unplanned, strict lsa check
 Max Grace-Period Configured      : 1800 Sec
 Last Helper-exit Reason          : none
```

4.11 Improving the Stability of an OSPFv3 Network

A stable OSPFv3 network features less route flapping, normal router performance, and good network performance.

Applicable Environment

By setting timers, you can reduce the number of unnecessary packets on networks and reduce the load on the network devices. Network performance is thus improved.

Pre-configuration Tasks

Before improving the security of an OSPFv3 network, complete the following tasks:

- Configuring a link layer protocol
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring basic OSPF3 functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

4.11.1 Setting the Preference of OSPFv3

If multiple dynamic routing protocols are running on the same device, routing information needs to be shared and which route to select needs to be determined. In this case, you can set preferences of routing protocols. In this manner, when different protocols discover routes to the same destination, the route discovered by the protocol with the highest preference will be selected.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
preference [ ase ] { preference } *
```

The preference of OSPF is set.

- If the parameter **ase** is specified, it indicates that the preference of AS external routes is set.
- The parameter **preference** specifies the preference of OSPFv3 routes. The smaller the value, the higher the preference.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.11.2 Configuring the Delay in Transmitting LSAs on the Interface

It takes time to transmit OSPFv3 packets on a link. Therefore, a certain delay is added to the aging time of an LSA before the LSA is sent. This configuration needs to be considered especially on low-speed links.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPFv3 interface view is displayed.

Step 3 Run:

```
ospfv3 trans-delay interval [ instance instance-id ]
```

The delay in transmitting LSAs is set on the interface.

By default, the delay in transmitting LSAs is 1 second.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.11.3 Configuring the Interval for Retransmitting LSAs

After a router sends an LSA to its neighbor, the router expects to receive an LSAck packet from the neighbor. If the router does not receive an LSAck packet within the LSA retransmission interval, it retransmits the LSA to the neighbor.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The OSPFv3 interface view is displayed.

Step 3 Run:

```
ospfv3 timer retransmit interval [ instance instance-id ]
```

The interval for retransmitting LSAs between adjacent routers is set.

By default, the interval for retransmitting LSAs is 5 seconds.

NOTE

The interval for retransmitting LSAs between adjacent routers should not be set too small. Otherwise, certain LSAs are retransmitted unnecessarily. Generally, the interval needs to be greater than the round trip time of a packet transmitted between two routers.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.11.4 Configuring Stub Routers

When the router has a heavy load and cannot forward any other packets, you can configure it as a stub router. After the router is configured as a stub router, other OSPFv3 routers do not use this router to forward data, but they can have a route to this stub router.

Context

A stub router is used to control traffic and instruct other OSPFv3 routers not to use it to forward data. Other OSPFv3 routers can have a route to the stub router.

The metric of links in the Router LSAs generated by the stub router is set to the maximum value (65535).

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 [ process-id ]
```

The OSPFv3 view is displayed.

Step 3 Run:

```
stub-router [ on-startup [ interval ] ]
```

A stub router is configured.

The parameter **on-startup** [*interval*] specifies the interval during which the router remains to be a stub router during master/slave switchover. By default, the interval during which the router remains to be a stub router is 500 seconds.

NOTE

There is no relationship between the stub router configured through this command and the router in a stub area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

4.11.5 Checking the Configuration

After the configurations are performed to improve the OSPFv3 network stability, you can view brief information about OSPFv3 and the IP routing table.

Procedure

- Run the **display ospfv3** [*process-id*] [**area** *area-id*] [*interface-type interface-number*] command in any view to view information about an OSPFv3 interface.
- Run the **display ospfv3** [*process-id*] **routing** command in any view to view information about an OSPFv3 routing table.

----End

Example

Run the **display ospfv3 interface** command, and you can view the values of OSPFv3 timers. For example:

```
<HUAWEI> display ospfv3 interface
GE1/0/0 is up, line protocol is up
  Interface ID 0x102
  Interface MTU 1500
  IPv6 Prefixes
    FE80::2E0:FFF:FE4E:F101 (Link-Local Address)
    2000::1/64
  Interface Event: 1
  Interface Lsa Count: 1
  Interface Lsa Checksum: 0x9b74
  OSPFv3 Process (1), Area 0.0.0.1, Instance ID 0
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State Full, Priority 1
    No designated router on this link
    No backup designated router on this link
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
```

Run the **display ospfv3 routing** command, and you can view the destination addresses, link costs, and next hops of OSPFv3 routes. For example:

```
<HUAWEI> display ospfv3 routing

    Destination                                                     Metric
     Next-hop
       Backup Next-hop
    45::/64                                                         10
      via FE80::2000:10FF:FE03:4, Ethernet3/0/0
       backup via FE80::2000:10FF:FE03:4, Pos1/0/1, LFA LINK-NODE
```

4.12 Configuring the Network Management Function of OSPFv3

OSPFv3 supports the network management function. You can configure the OSPFv3 Management Information Base (MIB) and bind it to an OSPFv3 process through the Simple Network Management Protocol (SNMP). In this manner, the OSPFv3 MIB manages multicast information exchanged between the Network Management Station (NMS) and agents.

Pre-configuration Tasks

Before configuring the network management function of OSPFv3, complete the following tasks:

- Configuring a link layer protocol
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring basic OSPFv3 functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospfv3 mib-binding process-id
```

The OSPFv3 process is bound to the MIB.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following command to check the previous configuration:

- Run the **display current-configuration** command to check whether an OSPFv3 process is bound to the OSPFv3 MIB.

4.13 Resetting OSPFv3

Restarting OSPFv3 can reset OSPFv3. In addition, you can choose to reset OSPFv3 through GR.

Context

CAUTION

The OSPFv3 adjacency is removed when you reset the OSPFv3 connection by using the **reset ospfv3** command. So, confirm the action before you use the command.

After modifying the OSPFv3 routing policy or protocol, reset the OSPFv3 connection to validate the modification. To reset OSPFv3 connections, run the **reset ospfv3** command in the user view.

Procedure

Step 1 To validate the new configuration, run the following commands:

1. **reset ospfv3** { *process-id* | **all** } [**flush-delay** *flush-delay-val*]
2. **reset ospfv3** { *process-id* | **all** } **counters** [*interface-type interface-number*]

----End

4.14 Configuration Examples

This section provides several configuration examples of OSPFv3 together with the configuration flowchart. The configuration examples explain networking requirements, configuration notes, and the configuration roadmap.

4.14.1 Example for Configuring Basic OSPFv3 Functions

This part provides an example for configuring basic OSPFv3 functions. Detailed operations include enabling OSPFv3 on each router and specifying network segments in different areas.

Networking Requirements

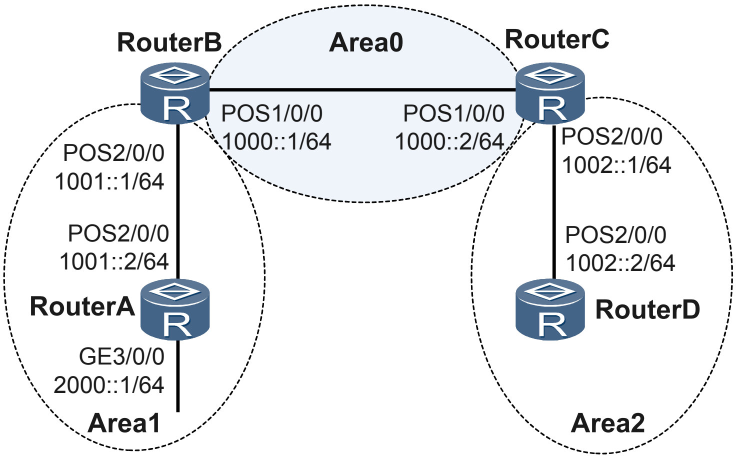
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 4-5**, all routers run OSPFv3. The entire autonomous system is divided into three areas. Router B and Router C serve as ABRs to forward the inter-area routes.

After the configuration, each router should learn the routes from the AS to all network segments.

Figure 4-5 Networking diagram of configuring OSPFv3 areas



Configuration Notes

When configuring basic OSPFv3 functions, pay attention to the following:

- The backbone area is responsible for forwarding inter-area routes. In addition, the routing information between non-backbone areas must be forwarded through the backbone area. OSPFv3 defines the following rules for the backbone area:
  - Connectivity must be available between non-backbone areas and the backbone area.
  - Connectivity must be available over the backbone area.
- The intervals for sending Hello, Dead, and Poll packets on the local interface must be the same as that on the peer interface. Otherwise, the OSPFv3 neighbor relationship cannot be established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable basic OSPFv3 functions on each router.
2. Check the routing list and LSDB.

Data Preparation

To complete the configuration, you need the following data:

Device Name	Router ID	Process ID	IPv6 Address
Router A	1.1.1.1	1	Area 1: 2000::1/64 and 1001::2/64
Router B	2.2.2.2	1	Area 0: 1000::1/64 Area 1: 1001::1/64
Router C	3.3.3.3	1	Area 0: 1000::2/64 Area 2: 1002::1/64
Router D	4.4.4.4	1	Area 2: 1002::2/64

Procedure

Step 1 Assign an IPv6 address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPFv3 functions.

# Configure Router A.

```
[~RouterA] ipv6
[~RouterA] ospfv3
[~RouterA-ospfv3-1] router-id 1.1.1.1
[~RouterA-ospfv3-1] quit
[~RouterA] interface gigabitethernet3/0/0
[~RouterA-GigabitEthernet3/0/0] ospfv3 1 area 1
```

```
[~RouterA-GigabitEthernet3/0/0] quit
[~RouterA] interface pos2/0/0
[~RouterA-Pos2/0/0] ospfv3 1 area 1
[~RouterA-Pos2/0/0] quit
[~RouterA] commit
```

# Configure Router B.

```
[~RouterB] ipv6
[~RouterB] ospfv3
[~RouterB-ospfv3-1] router-id 2.2.2.2
[~RouterB-ospfv3-1] quit
[~RouterB] interface pos1/0/0
[~RouterB-Pos1/0/0] ospfv3 1 area 0
[~RouterB-Pos1/0/0] quit
[~RouterB] interface pos2/0/0
[~RouterB-Pos2/0/0] ospfv3 1 area 1
[~RouterB-Pos2/0/0] quit
[~RouterB] commit
```

# Configure Router C.

```
[~RouterC] ipv6
[~RouterC] ospfv3
[~RouterC-ospfv3-1] router-id 3.3.3.3
[~RouterC-ospfv3-1] quit
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] ospfv3 1 area 0
[~RouterC-Pos1/0/0] quit
[~RouterC] interface pos 2/0/0
[~RouterC-Pos2/0/0] ospfv3 1 area 2
[~RouterC-Pos2/0/0] quit
[~RouterC] commit
```

# Configure Router D.

```
[~RouterD] ipv6
[~RouterD] ospfv3
[~RouterD-ospfv3-1] router-id 4.4.4.4
[~RouterD-ospfv3-1] quit
[~RouterD] interface pos 2/0/0
[~RouterD-Pos2/0/0] ospfv3 1 area 2
[~RouterD-Pos2/0/0] quit
[~RouterD] commit
```

Step 3 Verify the configuration.

# Display the OSPFv3 neighbors of Router B.

```
[~RouterB] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.1)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
1.1.1.1           1 Full/ -        00:00:34    PosS2/0/0           0
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
3.3.3.3           1 Full/ -        00:00:32    PosS1/0/0           0
```

# Display the OSPFv3 neighbors of Router C.

```
[~RouterC] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
2.2.2.2           1 Full/ -        00:00:37    Pos1/0/0           0
OSPFv3 Area (0.0.0.2)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
4.4.4.4           1 Full/ -        00:00:33    Pos2/0/0           0
```

# Display the OSPFv3 routing table of Router D.

```
[~RouterD] display ospfv3 routing
OSPFv3 Process (1)
   Destination                                   Metric
     Next-hop
  IA 1000::/64                                     2
           via FE80::1572:0:5EF4:1, Pos2/0/0
  IA 1001::/64                                     3
           via FE80::1572:0:5EF4:1, Pos2/0/0
     1002::/64                                     1
           directly-connected, Pos2/0/0
  IA 2000::/64                                     4
           via FE80::1572:0:5EF4:1, Pos2/0/0
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 ipv6
#
interface GigabitEthernet3/0/0
  undo shutdown
 ipv6 enable
 ipv6 address 2000::1/64
 ospfv3 1 area 0.0.0.1
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1001::2/64
 ospfv3 1 area 0.0.0.1
#
ospfv3 1
 router-id 1.1.1.1
 area 0.0.0.1
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
 ipv6
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1000::1/64
 ospfv3 1 area 0.0.0.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1001::1/64
 ospfv3 1 area 0.0.0.1
#
ospfv3 1
 router-id 2.2.2.2
 area 0.0.0.0
 area 0.0.0.1
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
 ipv6
#
interface Pos1/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1000::2/64
 ospfv3 1 area 0.0.0.0
#
interface Pos2/0/0
 link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1002::1/64
 ospfv3 1 area 0.0.0.2
#
ospfv3 1
 router-id 3.3.3.3
 area 0.0.0.0
 area 0.0.0.2
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
 ipv6
#
interface Pos2/0/0
link-protocol ppp
 undo shutdown
 ipv6 enable
 ipv6 address 1002::2/64
 ospfv3 1 area 0.0.0.2
#
ospfv3 1
 router-id 4.4.4.4
 area 0.0.0.2
#
return
```

Related Tasks

4.3 Configuring Basic OSPFv3 Functions

4.14.2 Example for Configuring OSPFv3 DR Election

This part provides an example for setting the DR priority on an interface for DR election on a broadcast network.

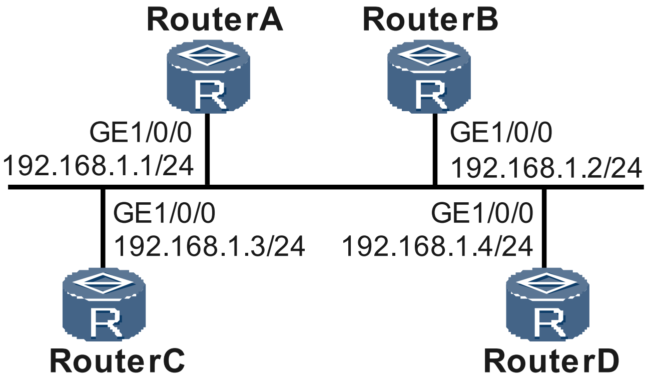
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 4-6**, Router A has the highest priority of 100 on the network and is elected as the DR; Router C has the second highest priority and is elected as the BDR. Router B has the priority of 0 and cannot be elected as a DR or a BDR; Router D is not configured with a priority and adopts the default value 1.

Figure 4-6 Networking diagram of configuring OSPFv3 DR election



Configuration Notes

Reconfiguring the DR priority for the router does not change the DR or BDR on a network. You can re-elect a DR or BDR by using the following methods. This, however, will result in the interruption of the OSPFv3 neighbor relationship between devices. Therefore, the following methods are used only when necessary.

- Restart the OSPFv3 processes on all routers.
- Configure the **shutdown** and **undo shutdown** commands on the interfaces where the OSPFv3 neighbor relationships are established.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the router ID on each router, enable OSPFv3, and specify the network segment.
2. Check the DR/BDR status with the default priority.
3. Configure the DR priority on the interface and check the DR/BDR status.

Data Preparation

To complete the configuration, you need the following data:

- Data of Router A, including the router ID (1.1.1.1) and DR priority (100)
- Data of Router B, including the router ID (2.2.2.2) and DR priority (0)
- Data of Router C, including the router ID (3.3.3.3) and DR priority (2)
- Data of Router D, including the router ID (4.4.4.4) and DR priority (1)

Procedure

Step 1 Assign an IPv6 address to each interface.

The details are not mentioned here.

Step 2 Configure basic OSPFv3 functions.

# Configure Router A, enable OSPFv3, and set its router ID to 1.1.1.1.

```
[~RouterA] ipv6
[~RouterA] ospfv3
[~RouterA-ospfv3-1] router-id 1.1.1.1
[~RouterA-ospfv3-1] quit
[~RouterA] interface GigabitEthernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] ospfv3 1 area 0
[~RouterA-GigabitEthernet1/0/0] quit
[~RouterA] commit
```

# Configure Router B, enable OSPFv3, and set its router ID to 2.2.2.2.

```
[~RouterB] ipv6
[~RouterB] ospfv3
[~RouterB-ospfv3-1] router-id 2.2.2.2
[~RouterB-ospfv3-1] quit
[~RouterB] interface GigabitEthernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] ospfv3 1 area 0
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] commit
```

# Configure Router C, enable OSPFv3, and set its router ID to 3.3.3.3.

```
[~RouterC] ipv6
[~RouterC] ospfv3
[~RouterC-ospfv3-1] router-id 3.3.3.3
[~RouterC-ospfv3-1] quit
[~RouterC] interface GigabitEthernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] ospfv3 1 area 0
[~RouterC-GigabitEthernet1/0/0] quit
[~RouterC] commit
```

# Configure Router D, enable OSPFv3, and set its router ID to 4.4.4.4.

```
[~RouterD] ipv6
[~RouterD] ospfv3
[~RouterD-ospfv3-1] router-id 4.4.4.4
[~RouterD-ospfv3-1] quit
[~RouterD] interface GigabitEthernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] ospfv3 1 area 0
[~RouterD-GigabitEthernet1/0/0] quit
[~RouterD] commit
```

# Display the neighbors of Router A. You can view the DR priority (its default value is 1) and the neighbor status. Router D is the DR and Router C is the BDR.

NOTE

The router with the greater router ID is the DR when routers have the same priority. If a certain Ethernet interface of a router becomes a DR, the other broadcast interfaces of the router have the highest priority in DR election. That is, the DR router is elected as the DR.

```
[~RouterA] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
2.2.2.2        1   2-Way/DROther    00:00:32    GE1/0/0        0
3.3.3.3        1   Full/Backup      00:00:36    GE1/0/0        0
4.4.4.4        1   Full/DR          00:00:38    GE1/0/0        0
```

# Display the neighbors of Router D, and you can view that all neighbors of Router D are in the Full state.

```
[~RouterD] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
1.1.1.1       1   Full/DROther     00:00:32    GE1/0/0         0
2.2.2.2       1   Full/DROther     00:00:35    GE1/0/0         0
3.3.3.3       1   Full/Backup      00:00:30    GE1/0/0         0
```

Step 3 Set the DR priority of the interface.

# Set the DR priority of Router A to 100.

```
[~RouterA] interface GigabitEthernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] ospfv3 dr-priority 100
[~RouterA-GigabitEthernet1/0/0] quit
[~RouterA] commit
```

# Set the DR priority of Router B to 0.

```
[~RouterB] interface GigabitEthernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] ospfv3 dr-priority 0
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] commit
```

# Set the DR priority of Router C to 2.

```
[~RouterC] interface GigabitEthernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] ospfv3 dr-priority 2
[~RouterC-GigabitEthernet1/0/0] quit
[~RouterC] commit
```

# Display the neighbors of Router A, and you can view that the DR priority is updated and the DR and BDR remain unchanged.

```
[~RouterA] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
2.2.2.2       0   2-Way/DROther    00:00:34    GE1/0/0         0
3.3.3.3       2   Full/Backup      00:00:38    GE1/0/0         0
4.4.4.4       1   Full/DR          00:00:31    GE1/0/0         0
```

# Display the neighbors of Router D, and you can view that Router D remains as the DR.

```
[~RouterD] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
1.1.1.1       100   Full/DROther  00:00:36    GE1/0/0         0
2.2.2.2       0   Full/DROther    00:00:30    GE1/0/0         0
3.3.3.3       2   Full/Backup     00:00:36    GE1/0/0         0
```

Step 4 Re-elect the DR/BDR.

# Restart all routers (or run the **shutdown** and **undo shutdown** commands on the interface that establishes the OSPFv3 neighbor relationship), and make OSPFv3 re-elect the DR/BDR.

Step 5 Verify the configuration.

# Display the neighbors of Router A, and you can view that Router C is the BDR.

```
[~RouterA] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
```

```
2.2.2.2       0   Full/DROther     00:00:31    GE1/0/0         0
3.3.3.3       2   Full/Backup      00:00:36    GE1/0/0         0
4.4.4.4       1   Full/DROther     00:00:39    GE1/0/0         0
[~RouterA]
```

# Display the neighbors of Router D, and you can view that Router A is the DR.

```
[~RouterD] display ospfv3 peer
OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID     Pri   State            Dead Time   Interface  Instance ID
1.1.1.1      100   Full/DR        00:00:39    GE1/0/0         0
2.2.2.2      0   2-Way/DROther    00:00:35    GE1/0/0         0
3.3.3.3      2   Full/Backup      00:00:39    GE1/0/0         0
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1001::1/64
 ospfv3 1 area 0.0.0.0
 ospfv3 dr-priority 100
#
ospfv3 1
 router-id 1.1.1.1
 area 0.0.0.0
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
 ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1001::2/64
 ospfv3 1 area 0.0.0.0
 ospfv3 dr-priority 0
#
ospfv3 1
 router-id 2.2.2.2
 area 0.0.0.0
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
 ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1001::3/64
 ospfv3 1 area 0.0.0.0
 ospfv3 dr-priority 2
#
ospfv3 1
 router-id 3.3.3.3
 area 0.0.0.0
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
 ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1001::4/64
 ospfv3 1 area 0.0.0.0
#
ospfv3 1
 router-id 4.4.4.4
 area 0.0.0.0
#
return
```

4.14.3 Example for Configuring OSPFv3 IP FRR

This section describes the procedure for configuring OSPFv3 IP FRR, including blocking FRR on certain interfaces and associating OSPFv3 FRR with a BFD session.

Networking Requirements

CAUTION

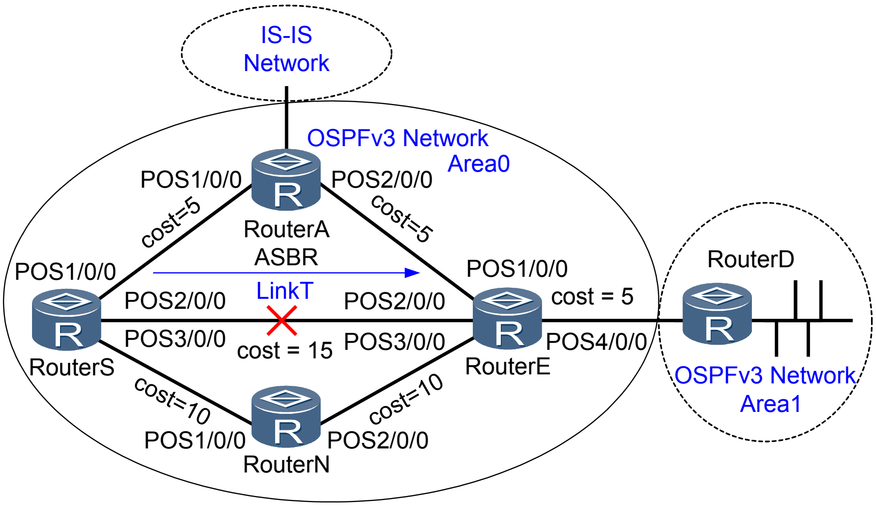
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

When a fault occurs on the network, OSPFv3 IP FRR rapidly switches traffic to the backup link without waiting for route convergence. This ensures non-stop traffic forwarding.

As shown in **Figure 4-7**,

- OSPFv3 runs on all devices.
- The link cost satisfies the OSPFv3 IP FRR traffic protection inequality.
- If a fault occurs on the primary link T, it is required that the traffic from Router S should be rapidly redirected to the backup link that passes through Router N.
- In the network planning, the link passing through Router A does not serve as a backup link.

Figure 4-7 Networking diagram for configuring OSPFv3 IP FRR



Context

When configuring OSPFv3 IP FRR, note the following:

Before configuring OSPFv3 IP FRR, block FRR on certain interfaces to prevent the links connected to these interfaces from serving as backup links. So, FRR calculation will ignore these links.

OSPFv3 IP FRR requires the lower layer to rapidly respond to the link change so that traffic can be rapidly switched to the backup link when the primary link fails. In such a case, if the **bfd all-interfaces frr-binding** command is configured, the association between the BFD session status and link status is enabled on an interface. If the BFD session on the interface becomes Down, the link status changes to Down accordingly, thus implementing fast fault sensing.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable basic OSPFv3 functions on each router.
2. Configure BFD for OSPFv3 on all devices in Area 0.
3. Configure link costs to make link T be preferred in route selection.
4. Block FRR on the specified interface on Router S.
5. Enable OSPFv3 IP FRR on Router S because the traffic forwarded by it needs to be protected.

Data Preparation

To complete the configuration, you need the following data:

Device	Router ID	Interface	IPv6 Address

Router S	1.1.1.1	POS1/0/0	1000::1/24
		POS2/0/0	1001::1/24
		POS3/0/0	1002::1/24
Router A	2.2.2.2	POS1/0/0	1000::2/24
		POS2/0/0	2000::2/24
Router N	3.3.3.3	POS1/0/0	1002::2/24
		POS2/0/0	2002::2/24
Router E	4.4.4.4	POS1/0/0	2000::1/24
		POS2/0/0	2001::1/24
		POS3/0/0	2002::1/24
		GE4/0/0	3000::1/24

Procedure

Step 1 Assign an IPv6 address to each interface. The configuration details are not mentioned here.

Step 2 Configure basic OSPFv3 functions. See **Example for Configuring Basic OSPFv3 Functions**.

Step 3 Configure BFD for OSPFv3 on all devices in Area 0. See **Example for Configuring BFD for OSPFv3**.

Step 4 Configure link costs to make link T be preferred in route selection.

# Configure Router S.

```
[~RouterS] interface pos1/0/0
[~RouterS-Pos1/0/0] ospfv3 cost 5
[~RouterS-Pos1/0/0] quit
[~RouterS] interface pos2/0/0
[~RouterS-Pos2/0/0] ospfv3 cost 20
[~RouterS-Pos2/0/0] quit
[~RouterS] interface pos3/0/0
[~RouterS-Pos3/0/0] ospfv3 cost 10
[~RouterS-Pos3/0/0] quit
[~RouterS] commit
```

# Configure Router A.

```
[~RouterA] interface pos1/0/0
[~RouterA-Pos1/0/0] ospfv3 cost 5
[~RouterA-Pos1/0/0] quit
[~RouterA] interface pos2/0/0
[~RouterA-Pos2/0/0] ospfv3 cost 5
```

```
[~RouterA-Pos2/0/0] quit
[~RouterA] commit
```

# Configure Router N.

```
[~RouterN] interface pos1/0/0
[~RouterN-Pos1/0/0] ospfv3 cost 10
[~RouterN-Pos1/0/0] quit
[~RouterN] interface pos2/0/0
[~RouterN-Pos2/0/0] ospfv3 cost 10
[~RouterN-Pos2/0/0] quit
[~RouterN] commit
```

Step 5 Block FRR on the specified interface on Router S.

```
[~RouterS] interface pos1/0/0
[~RouterS-Pos1/0/0] ospfv3 frr block
[~RouterS-Pos1/0/0] quit
[~RouterS] commit
```

Step 6 Enable OSPFv3 IP FRR on Router S.

# Enable OSPFv3 IP FRR on Router S.

```
[~RouterS] ospfv3
[~RouterS-ospfv3-1] frr
[~RouterS-ospfv3-1-frr] loop-free-alternate
[~RouterS-ospfv3] commit
```

Step 7 Verify the configuration.

# Run the **display ospfv3 routing** command on Router S to view routing information.

```
[~RouterS-ospfv3-1-frr] display ospfv3 routing 3000::1/24
   Destination                                  Metric
     Next-hop
       Backup Next-hop
   3000::1/24                                   10
       via 2001::1/24, POS2/0/0
          backup via FE80::2000:10FF:FE03:4, POS3/0/0, LFA LINK-NODE
```

The command output shows that a backup link is generated by FRR calculation on Router S.

----End

Configuration Files

- Configuration file of Router S

```
#
 sysname RouterS
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 1000::1 255.255.255.0
 ospfv3 cost 5
#
interface POS2/0/0
 link-protocol ppp
 ip address 1001::1 255.255.255.0
 ospfv3 cost 15
#
interface POS3/0/0
 link-protocol ppp
 ip address 1002::1 255.255.255.0
 ospfv3 frr block
 ospfv3 cost 10
#
```

```
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
ospfv3 1 router-id 1.1.1.1
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
  loop-free-alternate
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 10.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router A

```
#
 sysname RouterA
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 1000::2 255.255.255.0
 ospfv3 cost 5
#
interface POS2/0/0
 link-protocol ppp
 ip address 2000::2 255.255.255.0
 ospfv3 cost 5
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
ospfv3 1 router-id 2.2.2.2
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
  loop-free-alternate
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 20.1.2.0 0.0.0.255
#
return
```

- Configuration file of Router N

```
#
 sysname RouterN
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 1002::2 255.255.255.0
 ospfv3 cost 10
#
interface POS2/0/0
 link-protocol ppp
 ip address 2002::2 255.255.255.0
 ospfv3 cost 10
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
ospfv3 1 router-id 3.3.3.3
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 frr
 area 0.0.0.1
  network 10.1.3.0 0.0.0.255
  network 20.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router E

```
#
 sysname RouterE
#
 bfd
#
interface POS1/0/0
 link-protocol ppp
 ip address 2000::1 255.255.255.0
#
interface POS2/0/0
 link-protocol ppp
 ip address 2001::1 255.255.255.0
#
interface POS3/0/0
 link-protocol ppp
 ip address 2002::1 255.255.255.0
#
interface GigabitEthernet4/0/0
 link-protocol ppp
 ip address 3000::1 255.255.255.0
 ospfv3 cost 5
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
ospfv3 1 router-id 4.4.4.4
 bfd all-interfaces enable
 bfd all-interfaces frr-binding
 area 0.0.0.1
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 10.1.3.0 0.0.0.255
  network 172.17.1.0 0.0.0.255
#
return
```

Related Tasks

4.7 Configuring OSPFv3 IP FRR

4.14.4 Example for Configuring BFD for OSPFv3

This section describes how to configure BFD for OSPFv3. After BFD for OSPFv3 is configured, BFD can rapidly detect link faults and report them to OSPFv3 so that service traffic can be directed to the backup link.

Networking Requirements

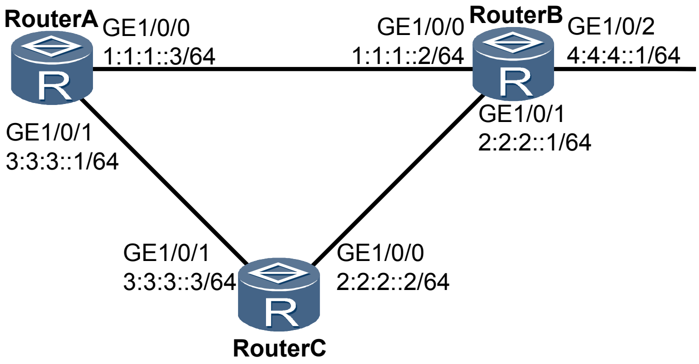
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 4-8**,

- OSPFv3 runs on Router A, Router B, and Router C.
- BFD is enabled in an OSPFv3 process on Router A, Router B, and Router C.
- Service traffic is transmitted over the primary link Router A -> Router B. The link Router A -> Router C -> Router B serves as the backup link.
- If the primary link between Router A and Router B fails, it is required that service traffic be directed to the backup link after BFD senses the fault and reports it to OSPFv3.

Figure 4-8 Networking diagram for configuring BFD for OSPFv3



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable basic OSPFv3 functions on each router.
2. Configure BFD for OSPFv3.

Data Preparation

To complete the configuration, you need the following data:

- Router ID of Router A: 1.1.1.1
- Router ID of Router B: 2.2.2.2
- Router ID of Router C: 3.3.3.3
- Minimum interval for sending BFD packets, minimum interval for receiving BFD packets, and local detection multiple on Router A and Router B

Procedure

Step 1 Assign an IPv6 address to each interface. The detailed configuration is not mentioned here.

Step 2 Configure basic OSPFv3 functions.

# Configure Router A.

```
[~RouterA] ipv6
[~RouterA] ospfv3
[~RouterA-ospfv3-1] router-id 1.1.1.1
[~RouterA-ospfv3-1] quit
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] ipv6 enable
[~RouterA-GigabitEthernet1/0/0] ospfv3 1 area 0
[~RouterA-GigabitEthernet1/0/0] quit
[~RouterA] interface gigabitethernet 1/0/1
```

```
[~RouterA-GigabitEthernet1/0/1] ipv6 enable
[~RouterA-GigabitEthernet1/0/1] ospfv3 1 area 0.0.0.0
[~RouterA-GigabitEthernet1/0/1] quit
[~RouterA] commit
```

# Configure Router B.

```
[~RouterB] ipv6 enable
[~RouterB] ospfv3 1
[~RouterB-ospfv3-1] router-id 2.2.2.2
[~RouterB-ospfv3-1] quit
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] ipv6 enable
[~RouterB-GigabitEthernet1/0/0] ospfv3 1 area 0.0.0.0
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] interface gigabitethernet 1/0/1
[~RouterB-GigabitEthernet1/0/1] ipv6 enable
[~RouterB-GigabitEthernet1/0/1] ospfv3 1 area 0.0.0.0
[~RouterB-GigabitEthernet1/0/1] quit
[~RouterB] interface gigabitethernet 1/0/2
[~RouterB-GigabitEthernet1/0/2] ipv6 enable
[~RouterB-GigabitEthernet1/0/2] ospfv3 1 area 0.0.0.0
[~RouterB-GigabitEthernet1/0/2] commit
```

# Configure Router C.

```
[~RouterC] ospfv3 1
[~RouterC-ospfv3-1] router-id 3.3.3.3
[~RouterC-ospfv3-1] quit
[~RouterC] interface gigabitethernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] ipv6 enable
[~RouterC-GigabitEthernet1/0/0] ospfv3 1 area 0.0.0.0
RouterC-GigabitEthernet1/0/0] quit
[~RouterC] interface gigabitethernet 1/0/1
[~RouterC-GigabitEthernet1/0/1] ipv6 enable
[~RouterC-GigabitEthernet1/0/1] ospfv3 1 area 0.0.0.0
[~RouterC-GigabitEthernet1/0/1] commit
```

# After the preceding configurations are complete, run the **display ospfv3 peer** command. The command output shows that neighbor relationships are set up between Router A and Router B, and between Router B and Router C. The following takes the display on Router A as an example.

```
[~RouterA] display ospfv3 peer verbose
OSPFv3 Process (1)
Neighbor 2.2.2.2 is Full, interface address FE80::E0:CE19:8142:1
    In the area 0.0.0.0 via interface GE1/0/0
    DR Priority is 1 DR is 2.2.2.2 BDR is 1.1.1.1
    Options is 0x000013 (-|R|-|-|E|V6)
    Dead timer due in 00:00:34
    Neighbour is up for 01:30:52
    Database Summary Packets List 0
    Link State Request List 0
    Link State Retransmission List 0
    Neighbour Event: 6
    Neighbour If Id : 0xe
Neighbor 3.3.3.3 is Full, interface address FE80::E0:9C69:8142:2
    In the area 0.0.0.0 via interface GE1/0/1
    DR Priority is 1 DR is 3.3.3.3 BDR is 1.1.1.1
    Options is 0x000013 (-|R|-|-|E|V6)
    Dead timer due in 00:00:37
    Neighbour is up for 01:31:18
    Database Summary Packets List 0
    Link State Request List 0
    Link State Retransmission List 0
    Neighbour Event: 6
    Neighbour If Id : 0x9
```

# View information in the OSPFv3 routing table on Router A. The routing table should contain the routes to Router B and Router C.

```
[~RouterA] display ospfv3 routing
OSPFv3 Process (1)
Destination                                                Metric
  Next-hop
1:1:1::/64                                                   1
 directly connected, GE1/0/0
2:2:2::/64                                                   2
 via FE80::E0:9C69:8142:2, GE1/0/1
 via FE80::E0:CE19:8142:1, GE1/0/0
3:3:3::/64                                                   1
 directly connected, GE1/0/1
4:4:4::1/64                                                  1
 via FE80::E0:CE19:8142:1, GE1/0/0
```

The information shows that the next hop of the route to 4:4:4::1/64 is GigabitEthernet 1/0/0. Traffic is transmitted over the primary link Router A -> RouterB.

Step 3 Configure BFD for OSPFv3.

# Enable BFD on Router A globally.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] ospfv3
[~RouterA-ospfv3-1] bfd all-interfaces enable min-transmit-interval 100 min-
receive-interval 100 detect-multiplier 4
[~RouterA-ospfv3-1] commit
```

# Enable BFD on Router B globally.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] ospfv3
[~RouterB-ospfv3-1] bfd all-interfaces enable min-transmit-interval 100 min-
receive-interval 100 detect-multiplier 4
[~RouterB-ospfv3-1] commit
```

# Enable BFD on Router C globally.

```
[~RouterC] bfd
[~RouterC-bfd] quit
[~RouterC] ospfv3
[~RouterC-ospfv3-1] bfd all-interfaces enable  min-transmit-interval 100 min-
receive-interval 100 detect-multiplier 4
[~RouterC-ospfv3-1] commit
```

# After the preceding configurations are complete, run the **display ospfv3 bfd session** command on Router A or Router B. You can view that the status of the BFD session is Up.

The following takes the display on Router B as an example.

```
<RouterB> display ospfv3 bfd session verbose
* - STALE
OSPFv3 Process (1)
   Neighbor-Id: 1.1.1.1
   BFD Status: Up
   Interface: GE1/0/0
   IPv6-Local-Address: FE80::E0:CE19:8142:1
   IPv6-Remote-Address: FE80::E0:4C3A:143:1
   BFD Module preferred timer values
      Transmit-Interval(ms): 100
      Receive-Interval(ms): 100
      Detect-Multiplier: 3
   OSPFv3 Module preferred timer values
      Transmit-Interval(ms): 100
      Receive-Interval(ms): 100
      Detect-Multiplier: 3
   Configured timer values
      Transmit-Interval(ms): 100
      Receive-Interval(ms): 100
      Detect-Multiplier: 3
   Neighbor-Id: 3.3.3.3
   BFD Status: Down
   Interface: GE1/0/1
   IPv6-Local-Address: FE80::E0:CE19:8142:2
   IPv6-Remote-Address: FE80::E0:9C69:8142:1
   BFD Module preferred timer values
      Transmit-Interval(ms): 2200
      Receive-Interval(ms): 2200
      Detect-Multiplier: 0
   OSPFv3 Module preferred timer values
      Transmit-Interval(ms): 1000
      Receive-Interval(ms): 1000
      Detect-Multiplier: 3
   Configured timer values
      Transmit-Interval(ms): 1000
      Receive-Interval(ms): 1000
      Detect-Multiplier: 3
```

Step 4 Verify the configuration.

# Run the **shutdown** command on GE 1/0/0 of Router B to simulate a failure of the primary link.

```
[~RouterB] interface gigabitethernet1/0/0
[~RouterB-GigabitEthernet1/0/0] shutdown
[~RouterB-GigabitEthernet1/0/0] commit
```

# Check the routing table on Router A. The command output shows that when the primary link fails, the next hop on the route to 4:4:4::1/64 is changed to GigabitEthernet 1/0/1. Therefore, traffic is switched to the backup link.

```
<RouterA> display ospfv3 routing
OSPFv3 Process (1)
Destination                                                 Metric
  Next-hop
 1:1:1::/64                                                    1
       directly connected, GE1/0/0
 2:2:2::/64                                                    2
       via FE80::E0:9C69:8142:2, GE1/0/1
 3:3:3::/64                                                    1
       directly connected, GE1/0/1
 4:4:4::1/64                                                   2
       via FE80::E0:9C69:8142:2, GE1/0/1
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 ipv6
#
 bfd
#
ospfv3 1
 router-id 1.1.1.1
 bfd all-interfaces enable min-transmit-interval 100 min-receive-interval 100
detect-multiplier 4
#
interface gigabitethernet1/0/0
 ipv6 enable
 ipv6 address 1:1:1::3/64
 ospfv3 1 area 0.0.0.0
```

```
#
interface gigabitethernet1/0/1
 ipv6 enable
 ipv6 address 3:3:3::1/64
 ospfv3 1 area 0.0.0.0
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
 ipv6
#
 bfd
#
ospfv3 1
 router-id 2.2.2.2
 bfd all-interfaces enable min-transmit-interval 100 min-receive-interval 100
detect-multiplier 4
#
interface gigabitethernet1/0/0
 ipv6 enable
 ipv6 address 1:1:1::2/64
 ospfv3 1 area 0.0.0.0
#
interface gigabitethernet1/0/1
 ipv6 enable
 ipv6 address 2:2:2::1/64
 ospfv3 1 area 0.0.0.0
#
interface gigabitethernet1/0/2
 ipv6 enable
 ipv6 address 4:4:4::1/64
 ospfv3 1 area 0.0.0.0
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
 ipv6
#
ospfv3 1
 router-id 3.3.3.3
 bfd all-interfaces enable min-transmit-interval 100 min-receive-interval 100
detect-multiplier 4
#
interface gigabitethernet1/0/0
 ipv6 enable
 ipv6 address 2:2:2::2/64
 ospfv3 1 area 0.0.0.0
#
interface gigabitethernet1/0/1
 ipv6 enable
 ipv6 address 3:3:3::3/64
 ospfv3 1 area 0.0.0.0
#
return
```

Related Tasks

4.8 Configuring BFD for OPSFv3

5 RIP Configuration

About This Chapter

RIP can advertise and receive routes to affect the selection of data forwarding paths, and provide the NMS (Network Management Station) function. RIP is applicable to small-scale networks.

RIP maintenance is implemented through debugging. You need to note that debugging affects system performance.

5.11 Configuration Examples

This section provides configuration examples of RIP. You can understand the configuration procedures through the configuration flowchart. Each configuration example consists of such information as the networking requirements, configuration notes, and conf

5.1 Overview of RIP

RIP is a simple Interior Gateway Protocol (IGP) and based on the distance-vector (DV) algorithm. It is used in small-scale networks.

The Routing Information Protocol (RIP) is a simple Interior Gateway Protocol (IGP), which is mainly used for small-scale networks such as campus networks and simple regional networks. The implementation, configuration, and maintenance of RIP are easier than those of the Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) protocols. Thus, RIP is widely used in the actual networking.

RIP is a distance-vector routing protocol and it exchanges routing information through User Datagram Protocol (UDP) packets by using port 520.

RIP has two versions:

- RIP-1, a classful routing protocol
- RIP-2, a classless routing protocol

To improve the performance and prevent routing loops, RIP supports both split horizon and poison reverse.

5.2 RIP Features Supported by NE5000E

When configuring RIP, you can view different RIP versions and information about the RIP multi-instance.

The NE5000E supports the following RIP features:

- RIP version 1 (RIP-1) and RIP version 2 (RIP-2)
- RIP multi-instance: It can function as an internal routing protocol of Virtual Private Networks (VPNs), and run between Customer Edge routers (CEs) and Provider Edge routers (PE)s in Multiprotocol Label Switching (MPLS) L3VPN networks.

  NOTE

  For detailed configuration of this feature, see the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN.*
- Routing policies: When advertising, receiving, or importing routes, a router implements certain policies as required to filter the routes and change the attributes of the routes.
- Importing external routes: RIP can import routes from other IGPs such as OSPF and IS-IS. Even direct, static, and Border Gateway Protocol (BGP) routes can be imported into RIP.
- Authentication: RIP supports simple, MD5, and Keychain authentication.

5.3 Configuring Basic RIP Functions

The configuration of basic RIP functions, including starting RIP and specifying the network segment and version that run RIP, is a precondition of using RIP features.

Applicable Environment

Configuring basic RIP functions is a prerequisite for building RIP networks.

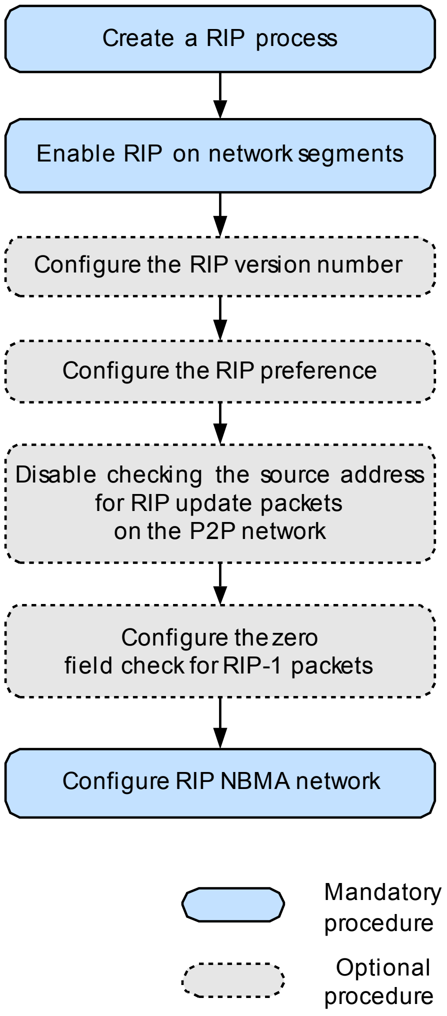
Pre-configuration Tasks

Before configuring basic RIP functions, complete the following tasks:

- Configuring the link layer protocol
- Configuring network layer addresses for interfaces to ensure that neighboring nodes are reachable at the network layer

Configuration Procedure

Figure 5-1 Flowchart of configuring basic RIP functions


Create a RIP process
Enable RIP on network segments
Configure the RIP version number
Configure the RIP preference
Disable checking the source address for RIP update packets on the P2P network
Configure the zero field check for RIP-1 packets
Configure RIP NBMA network
Mandatory procedure
Optional procedure

Related Tasks

5.11.1 Example for Configuring Basic RIP Functions

5.3.1 Creating a RIP process

The creation of RIP processes is a precondition of all RIP configurations.

Applicable Environment

Before running the RIP protocol on a device, you need to create a RIP process on the device.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

RIP supports multi-instance. To bind RIP processes to VPN instances, you can run the **rip** [*process-id*] **vpn-instance** *vpn-instance-name* command.

Step 3 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

If you run RIP-related commands in the interface view before enabling RIP, the configurations take effect only after RIP is enabled.

5.3.2 Enabling RIP on Specified Network Segments

A RIP process sends and receives RIP routes only on specified RIP-enabled network segments.

Context

You can enable RIP on specified network segments by using either of the following methods:

- Run the **network** command in the RIP process view to enable a RIP process to send and receive routes on a specified network segment.
- Run the **rip enable** command in the interface view to enable a RIP process to send and receive routes on all the network segments where a specified interface is located.

  NOTE

  The configuration of the **rip enable** command takes precedence over that of the **network** command.

Procedure

- Enable a RIP process to send and receive routes on specified network segments.
  1. Run:

     ```
     system-view
     ```

The system view is displayed.

2. Run:

   ```
   rip [ process-id ]
   ```

   A RIP process is created and the view of the RIP process is displayed.

3. Run:

   ```
   network network-address
   ```

   RIP is enabled on a specified network segment.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

   RIP runs only on interfaces on a specified network segment.

- Enable a RIP process to send and receive routes on all specified network segments where a specified interface is located:

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

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

     The interface view is displayed.

  3. Run:

     ```
     rip enable process-id
     ```

     RIP is enabled on all the network segments where the interface is located.

  4. Run:

     ```
     commit
     ```

     The configuration is committed.

----End

5.3.3 (Optional) Configuring the RIP Version Number

RIP versions include RIP-1 and RIP-2. The two versions have different functions.

Procedure

- Configuring the global RIP version number

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     rip [ process-id ]
     ```

     The RIP process is created and the RIP view is displayed.

  3. Run:

     ```
     version { 1 | 2 }
     ```

The global RIP version number is specified.

4. Run:

```
commit
```

The configuration is submitted.

- Configuring the RIP version number on an interface

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

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

The interface view is displayed.

3. Run:

```
rip version { 1 | 2 [ broadcast | multicast ] }
```

The RIP version number of the packets that are sent and received by the interface is specified.

4. Run:

```
commit
```

The configuration is submitted.

NOTE

By default, an interface receives and sends RIP-2 packets. When you configure RIP-2 for an interface, you can configure the interface to send packets in broadcast or multicast mode. If no RIP version is configured on the interface, the global version is used as the standard version.

----End

5.3.4 Configuring the RIP Preference

When multiple routing protocols are running on the same device, you can adjust the preference of the RIP protocol to enable the device to select the optimal route.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
preference [ route-policy route-policy-name ] preference
```

The RIP preference is set.

The **preference** command can be used together with the routing policy to set the preference for routes that match the routing policy.

After RIP routes are delivered to the RM(Routing Management), if the RIP preference changes, the RM updates the routing table again.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.3.5 (Optional) Disabling Checking the Source Address for RIP Packets on the P2P Network

On a P2P network, IP addresses of interfaces on the two ends of a link may belong to different networks. Therefore, the packets from neighbors can be accepted only when checking the source addresses of the incoming packets is disabled.

Context

By default, RIP checks the source addresses of the received packets to ensure that the local RIP interface receives only the packets from the same network. On a P2P network, IP addresses of interfaces on the two ends of a link may belong to different networks. Therefore, the two ends of a link can set up a neighbor relationship only when checking the source address of RIP packets is disabled.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
undo verify-source
```

The source address check is disabled for RIP packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.3.6 (Optional) Configuring the Zero Field Check for RIP-1 Packets

Certain fields in a RIP-1 packet must be 0, and these fields are called zero fields. By default, RIP-1 checks the zero fields of the received packets, and then discards the packets in which any of the zero fields is not 0.

Context

Certain fields in a RIP-1 packet must be 0, and RIP-2 packet contains no zero fields.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
checkzero
```

The zero field check is configured for RIP-1 packets.

By default, the zero field check is enabled on RIP-1 packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.3.7 Configuring NBMA Network

As the mode for sending RIP packets on a Non Broadcast Multiple Access (NBMA) network is different from those in other types of network, special configurations are required.

Applicable Environment

RIP packets are sent in unicast mode only on an NBMA network. On other types of network, interfaces send RIP packets in either broadcast or multicast mode.

Therefore, you need to perform the following configurations for an NBMA network:

- Specify RIP neighbors.
- Prevent interfaces from sending RIP packets in either broadcast or multicast mode.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
peer ip-address
```

RIP neighbors are configured on the NBMA network.

Step 4 Configure the interface to be silent as required.

- Run the **silent-interface all** command to configure all interfaces to be silent.
- Run the **silent-interface** *interface-type interface-number* command to configure a specific interface to be silent.

By default, interfaces can send RIP packets in either broadcast or multicast mode.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

5.3.8 Checking the Configuration

After basic RIP functions are successfully configured, you can view the current running status of RIP and RIP routing information.

Prerequisite

All configurations of basic RIP functions are complete.

Procedure

- Run the **display rip** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the running status and configuration of RIP.
- Run the **display rip** *process-id* **route** command to check RIP routes.

----End

Example

Run the **display rip** command, and you can view the running status and configuration of the created RIP process, such as the current RIP process, version number, whether to check the source address.

```
<HUAWEI> display rip
  Public VPN-instance
    RIP process : 1
       RIP version : 1
       Preference : 100
       Checkzero : Enabled
       Default-cost : 0
       Summary : Enabled
       Hostroutes : Enabled
       Maximum number of balanced paths : 3
       Update time   : 30 sec  Age time  : 180 sec
       Suppress time : 0 sec   Garbage-collect time : 120 sec
       Silent interfaces : None
       Default Route : Disabled
       Verify-source : Enabled
       Networks :
       172.4.0.0
       Configured peers : None
       Number of routes in database : 4
       Number of interfaces enabled : 3
       Triggered updates sent : 3
       Number of route changes : 6
       Number of replies to queries : 1
       Description : RIP
  Total count for 1 process:
     Number of routes in database : 1
     Number of interfaces enabled :  1
     Number of routes sendable in a periodic update : 1
     Number of routes sent in last periodic update : 1
```

Run the **display rip** *process-id* **route** command, and you can view all the active and inactive routes of the specified RIP process.

```
<HUAWEI> display rip 1 route
 Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
 ----------------------------------------------------------------------------
 Peer 192.4.5.1  on Pos4/0/1
      Destination/Mask        Nexthop     Cost   Tag     Flags   Sec
        172.4.0.0/16         192.4.5.1      1     0        RA      15
        192.13.14.0/24       192.4.5.1      2     0        RA      15
        192.4.5.0/24         192.4.5.1      1     0        RA      15
```

5.4 Preventing Routing Loops

RIP is a routing protocol based on the DV algorithm. RIP devices advertise their routing tables to their neighbors, so routing loops may occur.

Applicable Environment

RIP is a distance vector routing protocol, and a RIP device advertises its routing table to neighbors. In this case, routing loops may occur.

RIP avoids routing loops through the following mechanisms:

- Counting to infinity: RIP defines the cost 16 as infinity. If routing loops occur, when the cost of a route reaches 16, this route is considered unreachable.
- Split horizon: RIP does not advertise the routes learned from an interface to neighbors through that interface. This reduces bandwidth consumption and avoids routing loops.
- Poison reverse: After learning a route from an interface, RIP sets the cost of the route to 16 (indicating that the route is unreachable), and then advertises the route to neighbors through this interface. This clears unnecessary information in the routing tables of the neighbors.
- Suppression timer: A suppression timer can prevent routing loops and reduce the possibility of resulting in incorrect routing information due to the receiving of incorrect routes.
- Disabling an interface from receiving and sending RIP packets: This function has the similar effect as split horizon or poison reverse. That is, unreliable IP routes are filtered. Because neighbors cannot receive packets from the local router, routing information on the network may be incorrect.

NOTE

Counting to infinity is a basic feature of RIP and thus it does not need to be configured. Split horizon and poison reverse, however, need to be configured. When both split horizon and poison reverse are configured, only poison reverse takes effect.

For detailed description of these features, see the *HUAWEI NetEngine5000E Core Router Feature Description- RIP*.

Pre-configuration Tasks

Before configuring RIP to prevent routing loops on the network, complete the following tasks:

- Configuring IP addresses for interfaces ensure that the neighboring nodes reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Configuration Procedure

You can choose one or several of the following configuration tasks (excluding "Checking the Configuration") as required.

Related Tasks

5.11.2 Example for Preventing Routing Loops

5.4.1 Configuring Split Horizon

You can configure split horizon to prevent routing loops.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
rip split-horizon
```

Split horizon is enabled.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

If both split horizon and poison reverse are configured, only poison reverse takes effect.

If an interface is connected to an NBMA network, by default, split horizon is disabled on this interface.

5.4.2 Configuring Poison Reverse

You can configure poison reverse to prevent routing loops.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
rip poison-reverse
```

Poison reverse is enabled.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

If both split horizon and poison reverse are configured, only poison reverse takes effect.

On a Non Broadcast Multiple Access network, poison reverse is enabled on interfaces by default.

5.4.3 Configuring the Suppression Time

A suppression timer can prevent routing loops and reduce the possibility of generating incorrect routing information due to the receiving of incorrect routes.

Context

When receiving a route with the increased hop count, a device starts the Suppress timer, without updating the routes in the routing table. The device begins to accept the Update packet of this route until the Suppress timer expires.

By configuring the Suppress timer, you can delay the addition of incorrect routes to the routing table. This, however, also delays route convergence on the entire network. Therefore, you need to confirm the action before configuring the Suppress timer as required.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip process-id
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
timers rip update age suppress garbage-collect
```

The suppression time is set.

By default, the Suppress timer is 0s.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

RIP has four timers: the Update timer, Age timer, Suppress timer, and Garbage-collect timer. The relationship between the values of the four timers is as follows: *update* < *age* and *suppress* < *garbage-collect*. Changing the values of the timers affects RIP convergence speed and even causes route flapping on the network. For example, when the update time is longer than the aging time, if a RIP route changes within the update time, a device cannot inform its neighbors of the change in time.

For the configurations of the Update timer, Age timer, and Garbage-collect timer, see **5.7.2 Configuring RIP Timers**.

5.4.4 Checking the Configuration

After the prevention of routing loops is successfully configured, you can view the current running status of RIP, information about interfaces, and RIP routing information.

Prerequisite

All configurations of preventing routing loops are complete.

Procedure

- Run the **display rip** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the running status and configuration of RIP.
- Run the **display rip** *process-id* **route** command to check RIP routes.
- Run the **display rip** *process-id* **interface** command to check information about RIP interfaces.

----End

Example

Run the **display rip** command, and you can view the running status and configuration of the RIP process.

```
<HUAWEI> display rip
  Public VPN-instance
    RIP process : 1
       RIP version : 1
       Preference : 100
       Checkzero : Enabled
       Default-cost : 0
       Summary : Enabled
       Hostroutes : Enabled
       Maximum number of balanced paths : 3
       Update time   : 30 sec  Age time  : 180 sec
       Suppress time : 0 sec   Garbage-collect time : 120 sec
       Silent interfaces : None
       Default Route : Disabled
       Verify-source : Enabled
       Networks :
       172.4.0.0
       Configured peers : None
       Number of routes in database : 4
       Number of interfaces enabled : 3
       Triggered updates sent : 3
       Number of route changes : 6
       Number of replies to queries : 1
       Description  : RIP
  Total count for 1 process:
     Number of routes in database : 1
     Number of interfaces enabled :  1
     Number of routes sendable in a periodic update : 1
     Number of routes sent in last periodic update : 1
```

Run the **display rip** *process-id* **route** command, you can view all the routes learned by the RIP process.

```
<HUAWEI> display rip 1 route
 Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
 ----------------------------------------------------------------------------
 Peer 192.4.5.1  on Pos4/0/1
      Destination/Mask        Nexthop     Cost   Tag     Flags   Sec
        172.4.0.0/16         192.4.5.1      1    0         RA      15
        192.13.14.0/24       192.4.5.1      2    0         RA      15
        192.4.5.0/24         192.4.5.1      1    0         RA      15
```

Run the **display rip** *process-id* **interface verbose** command, and you can view information about the RIP interface, for example, whether the RIP interface is disabled from receiving or sending RIP packets and whether it is enabled with split horizon and poison reverse.

```
<HUAWEI> display rip 1 interface verbose
GigabitEthernet1/0/0 (1.1.1.2)
  State    : UP             MTU: 0
  Metricin : 0
  Metricout: 1
  Input    : Disabled       Output: Disabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast Packets
  Poison-reverse                : Enabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Max Packet Length             : 512
```

5.5 Adjusting RIP Route Selection

You can adjust RIP route selection in a complicated network.

Applicable Environment

The implementation of RIP is simple, and thus RIP is widely used in small and medium networks. To flexibly apply RIP on the current network to meet various requirements of users, you can change RIP route selection by setting different parameters.

Pre-configuration Tasks

Before adjusting RIP route selection, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Configuration Procedure

You can choose one or several of the following configuration tasks (excluding "Checking the Configuration") as required.

5.5.1 Disabling RIP-2 Classful Summarization

On the network where subnets are incontiguous, you can cancel the classful summarization of RIP-2 so that more accurate routing information can be obtained.

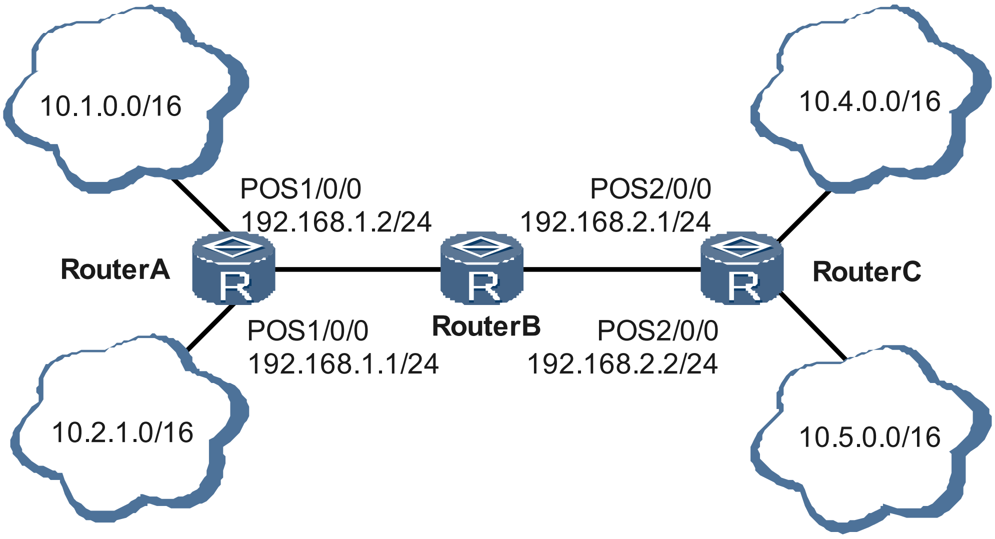
Context

Summarizing IP addresses can reduce the size of the routing table, but it shields routing information of subnets. As a result, incorrect routing information may be calculated.

On a non-contiguous subnet, you need to disable RIP-2 classful summarization. As shown in **Figure 5-2**, you need to disable split horizon on POS interfaces of Router A and Router C.

By default, RIP-2 classful summarization is enabled. Therefore, Router B and Router C send a route destined for 10.0.0.0/8 to Router A. Router A cannot differentiate 10.1.0.0/16, 10.2.0.0/16, 10.4.0.0/16, or 10.5.0.0/16. As a result, incorrect routes are calculated.

Figure 5-2 Disabling RIP-2 classful summarization



Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
undo summary
```

RIP-2 classful summarization is disabled.

By default, RIP-2 classful summarization is enabled.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.5.2 Configuring the Additional Metric on an Interface

The additional metric is a metric (number of hops) that is added to the original metric of an RIP route. You can run different commands to set additional metrics for RIP to receive and advertise routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
rip metricin value
```

The metric to be added to a received route is set.

Step 4 Run:

```
rip metricout { value | { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } value1 } *
```

The metric to be added to a sent route is set.

NOTE

When using an ACL or an IP prefix list together with the **rip metricout** command, you can specify the metric to be added to the RIP route that passes the filtering policy of the ACL or the IP prefix list. If a RIP route does not pass the filtering, its metric is increased by 1. Therefore, when an ACL or an IP prefix list is used together with the **rip metricout** command, the additional metric ranges from 2 to 15.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

After the **rip metricin** command is run, RIP adds the additional metric to the received route and then installs the route into the routing table. Therefore, after the receiving metric of an interface is increased, the metric of the RIP route received by this interface increases accordingly.

After the **rip metricout** command is run, RIP adds the additional metric to the route to be advertised. Therefore, after the sending metric of an interface is increased, the metric of the RIP route sent by this interface increases accordingly. In the routing table, however, the metric of the route is not changed.

If the metric of a route exceeds 16 after the additional metric is added, the metric is still calculated as 16.

5.5.3 Setting the Maximum Number of Equal-Cost Routes

You can set the maximum number of equal-cost RIP routes to adjust the number of routes that participate in traffic load balancing.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
maximum load-balancing  number
```

The maximum number of equal-cost routes is set.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.5.4 Checking the Configuration

After the RIP route selection is adjusted successfully, you can view the current running status of RIP, information about interfaces, and RIP routing information.

Prerequisite

All configurations of adjusting RIP route selection are complete.

Procedure

- Run the **display rip** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the running status and configuration of RIP.
- Run the **display rip** *process-id* **route** command to check RIP routes.
- Run the **display rip** *process-id* **interface** command to check information about RIP interfaces.

----End

Example

Run the **display rip** *process-id* command, and you can view the running status and configuration of the RIP process.

```
<HUAWEI> display rip
  Public VPN-instance
    RIP process : 1
       RIP version : 1
       Preference : 100
       Checkzero : Enabled
       Default-cost : 0
       Summary : Enabled
       Hostroutes : Enabled
       Maximum number of balanced paths : 3
       Update time   : 30 sec  Age time  : 180 sec
       Suppress time : 0 sec   Garbage-collect time : 120 sec
       Silent interfaces : None
       Default Route : Disabled
       Verify-source : Disabled
       Networks :
       172.4.0.0
       Configured peers : None
       Number of routes in database : 4
       Number of interfaces enabled : 3
       Triggered updates sent : 3
       Number of route changes : 6
       Number of replies to queries : 1
       Description : RIP
  Total count for 1 process:
     Number of routes in database : 1
     Number of interfaces enabled :  1
     Number of routes sendable in a periodic update : 1
     Number of routes sent in last periodic update : 1
```

Run the **display rip** *process-id* **interface verbose** command, and you can view the metric to be added to the received or sent route on the interface.

```
<HUAWEI> display rip 1 interface verbose
GigabitEthernet1/0/0 (1.1.1.2)
  State    : UP             MTU: 0
  Metricin : 2
  Metricout: 3
  Input    : Disabled       Output: Disabled
```

```
 Protocol : RIPv2 Multicast
 Send     : RIPv2 Multicast Packets
 Receive  : RIPv2 Multicast Packets
 Poison-reverse               : Enabled
 Split-Horizon                : Enabled
 Authentication type          : None
 Max Packet Length            : 512
```

Run the **display rip** *process-id* **route** command, and you can check whether there are routes for load balancing. For example, in the routing table, you can view that there are two equal-cost routes with the same destination address (10.10.10.10/32) and different next hops.

```
<HUAWEI> display rip 1 route
 Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
 -------------------------------------------------------------------------
 Peer 1.1.1.1 on GigabitEthernet3/0/2
      Destination/Mask        Nexthop     Cost   Tag     Flags   Sec
        2.2.2.0/24            1.1.1.1        1     0      RA       7
        10.10.10.10/32        1.1.1.1        1     0      RA       7
 Peer 2.2.2.2 on GigabitEthernet3/0/2
      Destination/Mask        Nexthop     Cost   Tag     Flags   Sec
        2.2.2.0/24            2.2.2.2        1     0      RA       7
        10.10.10.10/32        2.2.2.2        1     0      RA       7
```

5.6 Controlling RIP Routing Information

In practice, different protocols run on the same network. Therefore, you need to control routing information of every protocol to meet different networking requirements.

Applicable Environment

To meet the requirements of complex networking, it is required to accurately control the sending and receiving of RIP routing information.

Pre-configuration Tasks

Before configuring a router to control the receiving of RIP routing information, complete the following tasks:

- Configure IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Configuration Procedure

You can choose one or several of the following configuration tasks (excluding "Checking the Configuration") as required.

5.6.1 Configuring RIP to Import External Routes

RIP can import the routing information from other processes or other routing protocols to enrich the RIP routing table.

Context

On a large-scale network, different routing protocols are configured for devices in different ASs. In this case, you need to import routes learnt by other protocols to the devices.

If RIP needs to advertise the routing information of other routing protocols (direct, static, OSPF, IS-IS, or BGP), you can specify *protocol* to filter the specific routing information. If *protocol* is not specified, the routing information to be advertised is filtered, including the imported routes and local RIP routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 (Optional) Run:

```
default-cost cost
```

The default cost is set for the imported routes.

If no cost is specified when external routes are imported, the default cost 0 is used.

Step 4 Run:

```
import-route protocol [ process-id ] [ cost cost | route-policy route-policy-name ]
*
```

External routes are imported.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

Related Tasks

5.11.3 Example for Configuring RIP to Import External Routes

5.6.2 Configuring RIP to Advertise Default Routes

A default route is a route destined for 0.0.0.0. By default, RIP does not advertise default routes to its neighbors.

Context

In a routing table, the default route is a route to the network 0.0.0.0 (with the mask being 0.0.0.0). You can check whether the default route is configured by using the **display ip routing-table** command. When the destination address of a packet does not match any destination address in the routing table, the router uses a default route to forward this packet.

If the default route and the destination address of the packet do not exist in the routing table, the packet will be discarded by router. At the same time, an Internet Control Message Protocol (ICMP) packet is sent to report that the destination address or the destination network is unreachable.

After the **default-route originate** command is run, default routes are advertised to RIP neighbors only when there are default routes in the routing table. And the default route that is learned from a neighbor in the RIP routing table is deleted after the **default-route originate** command is run.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
default-route originate [ cost cost-value | tag tag-value ] *
```

RIP is enabled to advertise the default route.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.6.3 Configuring RIP to Filter the Received Routes

You can configure an inbound or outbound filtering policy by specifying Access Control Lists (ACLs) and IP address prefix lists to filter routes to be received and advertised. You can also configure a device to receive only the RIP packets from a specified neighbor.

Context

Devices can filter the routing information. To filter the received and advertised routes, you can configure inbound and outbound filtering policies by specifying the ACL and IP prefix list.

You can also configure a device to receive RIP packets from only a specified neighbor.

For details on how to configure RIP to filter the advertised routes, see **5.6.4 Configuring RIP to Filter the Routes to Be Advertised**.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Configure RIP to filter the received routes as required:

- Run:

  filter-policy *acl-number* **import** [*interface-type interface-number*]

  The learned routes are filtered based on the ACL.

- Run:

  filter-policy gateway *ip-prefix-name* **import**

  The routes advertised by neighbors are filtered based on the destination address prefix.

- Run:

  filter-policy acl-name *acl-name* **import** [*interface-type interface-number*]

  The routes learned by the specified interface are filtered based on the ACL name.

- Run:

  filter-policy ip-prefix *ip-prefix-name* [**gateway** *ip-prefix-name*] **import** [*interface-type interface-number*]

  The routes learned by the specified interface are filtered based on the destination address prefix and neighbors.

Step 4 Run:

commit

The configuration is submitted.

----End

5.6.4 Configuring RIP to Filter the Routes to Be Advertised

You can set conditions to filter the routes to be advertised. Only the routes that meet the conditions can be advertised.

Context

Devices can filter the routing information. To filter the advertised routes, you can configure inbound and outbound filtering policies by specifying the ACL and IP prefix list.

For details on how to configure RIP to filter the received routes, see **5.6.3 Configuring RIP to Filter the Received Routes**.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

rip [*process-id*]

The RIP process is created and the RIP view is displayed.

Step 3 Run:

filter-policy { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* } **export** [*protocol process-id* | *interface-type interface-number*]

The advertised routes are filtered based on the ACL and the destination address prefix.

Step 4 Run:

commit

The configuration is submitted.

----End

5.6.5 Disabling RIP from Receiving Host Routes

When you configure the function of disabling RIP from receiving host routes on a device, the device can reject host routes. This prevents the device from receiving a large number of unwanted routes and wasting network resources.

Context

In certain situations, the router may receive a large number of host routes from the same network segment. These routes are not required for route addressing, but consume many network resources. By disabling RIP from receiving host routes, you can configure the router to reject the received host routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
undo host-route
```

RIP is disabled from receiving host routes.

By default, RIP is allowed to receive host routes.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.6.6 Configuring the Zero Metric Check for RIP Packets

To communicate with other devices supporting packets with the zero metric, the Huawei device needs to be configured not to check the RIP packets with the zero metric.

Context

On the current networks, not all devices can receive the packets with the zero metric. By default, devices do not receive the packets with the zero metric. Therefore, RIP interfaces discard the RIP packets with the zero metric. On the current network, to make Huawei devices compatible with the devices that support the zero metric, you need to run the **undo zero-metric-check** command to allow interfaces to receive the RIP packets with the zero metric.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
undo zero-metric-check
```

The interface is allowed to receive the RIP packet with the zero metric.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

To restore the default settings, that is, to ignore the RIP packets with the zero metric, you can run the **zero-metric-check** command.

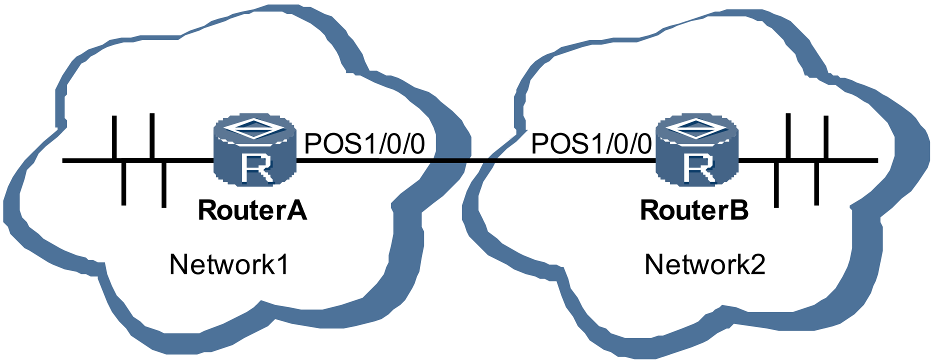
5.6.7 Disabling an Interface from Sending Packets

When you do not need to send routing information through the interface that connects to an external network, you can disable the interface from sending RIP packets.

Applicable Environment

As shown in **Figure 5-3**, RIP-enabled Network 1 is connected to Network 2 through the edge device Router A. You can disable POS 1/0/0 on Router A from sending packets.

Figure 5-3 Scenario where an interface is disabled from sending packets



Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
undo rip output
```

The interface is disabled from sending RIP packets.

By default, an interface is allowed to send RIP packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.6.8 Disabling an Interface from Receiving Packets

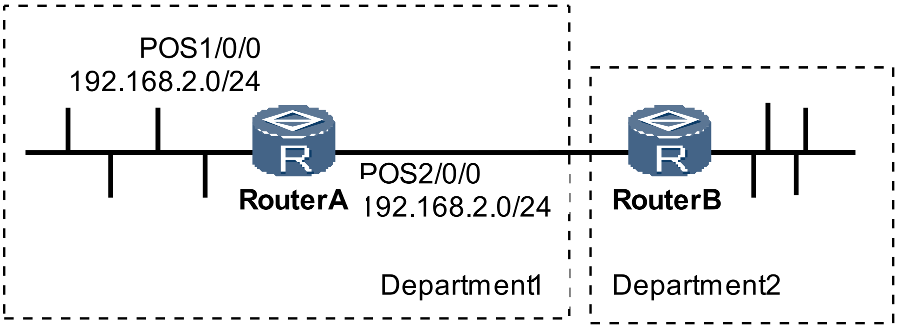
If an interface does not need to learn routing information from the peer, you can disable the interface from receiving RIP packets.

Applicable Environment

On an enterprise network where departments are not allowed to communicate with each other, you can disable interfaces from receiving packets.

As shown in **Figure 5-4**, if you do not want Department 1 to learn routing information about Department 2, you can disable POS 2/0/0 on Router A from receiving packets.

Figure 5-4 Scenario where an interface is disabled from receiving packets



Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
undo rip input
```

The interface is disabled from receiving RIP packets.

By default, an interface is allowed to receive RIP packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.6.9 Checking the Configuration

After the RIP routing information is successfully controlled, you can view all activated routes in the RIP database and the running status of RIP.

Prerequisite

All configurations of controlling the receiving of RIP routing information are complete.

Procedure

- Run the **display rip** *process-id* **database** command to check all active routes in the RIP database.
- Run the **display rip** *process-id* command to check the running status and configuration of the RIP process.

----End

Example

Run the **display rip** *process-id* **database** command, and you can view the default route and imported route. For example:

```
<HUAWEI> display rip 100 database
 0.0.0.0/0, cost 0, Default route originate
   172.4.0.0/16, cost 1, ClassfulSumm
       172.4.0.0/16, cost 1, nexthop 192.13.14.1
   20.0.0.0/8, cost 0, ClassfulSumm
       20.20.20.20/32, cost 0, Imported
```

Run the **display rip** *process-id* command, and you can check whether host routes can be received and whether there are silent interfaces. For example:

```
<HUAWEI> display rip
```

```
Public VPN-instance
  RIP process : 1
     RIP version : 1
     Preference : 100
     Checkzero : Enabled
     Default-cost : 0
     Summary : Enabled
     Hostroutes : Disabled
     Maximum number of balanced paths : 3
     Update time   : 30 sec  Age time  : 180 sec
     Suppress time : 0 sec   Garbage-collect time : 120 sec
     Silent interfaces : None
     Default Route : Disabled
     Verify-source : Disabled
     Networks :
     172.4.0.0
     Configured peers : None
     Number of routes in database : 4
     Number of interfaces enabled : 3
     Triggered updates sent : 3
     Number of route changes : 6
     Number of replies to queries : 1
     Description : RIP
```

5.7 Configuring RIP Fast Convergence

The network convergence speed is one of the key factors used to evaluate network performance.

Applicable Environment

The route convergence speed on a device is a performance index used to measure the network quality. Fast route convergence can improve the accuracy of routing information on the network.

Pre-configuration Tasks

Before configuring RIP fast convergence, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Configuration Procedure

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

5.7.1 Configuring RIP-2 Route Summarization

Configuring RIP-2 route summarization reduces the scale of the routing table, thus decreasing traffic on the network.

Context

Route summarization is the process of summarizing different subnet routes on the same natural network segment into one natural mask route when they are advertised to other network segments.

RIP-1 advertises routes with natural masks. Thus, RIP-1 does not support route summarization. RIP-2 supports Variable Length Subnet Mask (VLSM) and Classless Inter-Domain Routing

(CIDR). Thus, you can configure RIP-2 route summarization, which reduces the scale of the routing table, thus decreasing traffic on the network.

To broadcast all subnet routes, you can disable automatic route summarization of RIP-2.

Procedure

- Enabling RIP-2 automatic route summarization
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     rip [*process-id*]

     The RIP process is created and the RIP view is displayed.
  3. Run:
     summary

     RIP-2 automatic route summarization is enabled.
  4. Run:
     commit

     The configuration is submitted.
- Configuring RIP-2 to advertise the summarized address
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     interface *interface-type interface-number*

     The interface view is displayed.
  3. Run:
     rip summary-address *ip-address mask* [**avoid-feedback**]

     RIP-2 is configured to advertise the summarized local IP address.

     After **avoid-feedback** is configured, the local interface does not learn the summarized route whose destination address is the same as the advertised summary IP address, thus preventing routing loops.
  4. Run:
     commit

     The configuration is submitted.

----End

5.7.2 Configuring RIP Timers

There are four RIP timers, namely, the Update timer, Age timer, Suppress timer, and Garbage-collect timer. You can adjust the RIP convergence speed by changing the values of RIP timers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
timers rip update age suppress garbage-collect
```

RIP timers are configured.

By default, the Update timer is 30s; the Age timer is 180s; the Suppress timer is 0s; the Garbage-collect timer is four times the Update timer, namely, 120s.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

The relationship between the values of the preceding four timers is as follows: *update* < *age* and *suppress* < *garbage-collect*. Changing the values of the timers affects RIP convergence speed and even causes route flapping on the network. For example, when the update time is longer than the aging time, if a RIP route changes within the update time, a device cannot inform its neighbors of the change in time.

Configuring the Suppress timer can prevent routing loops. For details, see **5.4.3 Configuring the Suppression Time**.

5.7.3 Setting the Interval for Sending Packets and the Maximum Number of the Sent Packets

You can set the interval for sending RIP update messages and the maximum number of update messages that can be sent at a time to effectively control the memory used by a device to process RIP update messages.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
rip pkt-transmit { interval interval | number packet-count | bandwidth bandwidth-value } *
```

The interval for sending Update packets and the maximum number of packets sent each time are set on the interface.

By default, the interval for sending packets is 200 ms, and the number of packets to be sent each time is 30.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.7.4 Setting the Maximum Length of RIP packets

You can increase the maximum length of a RIP packet to enable the packet to carry more routes to be sent. This can make efficient use of bandwidth.

Context

CAUTION

You can run the **rip max-packet-length** command to set the length of a RIP packet to be greater than 512 bits only when the remote end can accept the RIP packet longer than 512 bits.

After the maximum length of a RIP packet is increased, Huawei devices may fail to communicate with non-Huawei devices. Therefore, use this command with caution.

By default, RIP can send only 25 routes in one packet.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
rip max-packet-length { value | mtu }
```

The maximum length of RIP packets is set.

mtu specifies the maximum length of a RIP packet that can be accepted.

By default, the maximum length of RIP packets is set to 512 bits.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.7.5 Checking the Configuration

After the fast convergence of a RIP network is successfully configured, you can view the current running status of RIP, RIP routing information, all activated routes in the RIP database, and information about interfaces.

Prerequisite

All configurations of RIP fast convergence are complete.

Procedure

- Run the **display rip** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the running status and configuration of RIP.
- Run the **display rip** *process-id* **route** command to check RIP routes.
- Run the **display rip** *process-id* **database** [**verbose**] command to check all active RIP routes in the database.
- Run the **display rip** *process-id* **interface** [*interface-type interface-number*] [**verbose**] command to check information about RIP interfaces.

----End

Example

Run the **display rip** command, and you can view the values of RIP timers. For example:

```
<HUAWEI> display rip
  Public VPN-instance
    RIP process : 1
       RIP version : 1
       Preference : 100
       Checkzero : Enabled
       Default-cost : 0
       Summary : Enabled
       Hostroutes : Enabled
       Maximum number of balanced paths : 3
       Update time    : 30 sec  Age time   : 180 sec
       Suppress time : 0 sec   Garbage-collect time : 120 sec
       Silent interfaces : None
       Default Route : Disabled
       Verify-source : Disabled
       Networks : 172.4.0.0
       Configured peers : None
       Number of routes in database : 4
       Number of interfaces enabled : 3
       Triggered updates sent : 3
       Number of route changes : 6
       Number of replies to queries : 1
       Description : RIP
  Total count for 1 process:
     Number of routes in database : 1
     Number of interfaces enabled :  1
     Number of routes sendable in a periodic update : 1
     Number of routes sent in last periodic update : 1
```

Run the **display rip** *process-id* **database** command, and you can view the summarized RIP route. For example:

```
<HUAWEI> display rip 1 database
 0.0.0.0/0, cost 0, Default route originate
   1.0.0.0/8, cost 0, ClassfulSumm
       1.1.1.0/24, cost 0, nexthop Rip-interface
   10.0.0.0/8, cost 1, ClassfulSumm
       10.10.0.0/16, cost NA, IfSumm
           10.10.10.10/32, cost 1, nexthop 1.1.1.1
           10.10.10.10/32, cost 1, nexthop 2.2.2.2
```

Run the **display rip** *process-id* **interface verbose** command, and you can view the maximum length of RIP packets to be received by RIP packets. For example:

```
<HUAWEI> display rip 1 interface verbose
GigabitEthernet1/0/0 (1.1.1.2)
  State    : UP              MTU: 0
  Metricin : 0
  Metricout: 1
  Input    : Disabled        Output: Disabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast Packets
  Poison-reverse                : Enabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Max Packet Length             : 500
```

5.8 Improving Security of a RIP Network

For the RIP network having higher requirements on security, you can configure RIP authentication and GTSM to enhance the security of the network.

Applicable Environment

The TCP/IP protocol suite has inherent defects and flawed implementation. Increasing network attacks have an increasingly greater impact on TCP/IP networks. Especially attacks on network devices will cause the crash of the network. Therefore, it is required to protect networks against attacks. The routing protocol algorithm (RPA) provides many techniques to ensure network security. By configuring authentication, you can improve the security of a RIP network.

RIP provides the following authentication modes in the interface view:

- Simple authentication
- MD5 authentication
- Keychain authentication

NOTE

If no authentication is configured on an interface, authentication configured on the process is adopted. For better security, it is recommended to configure different authentication modes on different interfaces.

Pre-configuration Tasks

Before improving the security of a RIP network, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Configuration Procedure

You can choose one or several of the following configuration tasks (excluding "Checking the Configuration") as required.

5.8.1 Configuring the Authentication Mode for RIP-2 Packets

RIP-2 supports authentication of protocol packets and provides two authentication modes, namely, plain text authentication and Message Digest 5 (MD5) authentication, to enhance security.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Do as follows as required:

- Run:

  ```
  rip authentication-mode simple password
  ```

  Simple authentication in the plain text is configured for RIP-2 packets.

  In simple authentication mode, the plain text password is transmitted along with authentication packets. Therefore, on a network requiring high security, it is not recommended to configure simple authentication.

- Run:

  ```
  rip authentication-mode md5 { nonstandard password-key key-id | usual password-
  key }
  ```

  MD5 authentication in the cipher text is configured for RIP-2 packets.

  In MD5 authentication mode, the MD5 password is used for packet encapsulation and decapsulation. MD5 authentication is more secure than simple authentication.

  nonstandard supports nonstandard authentication packets.

  usual supports IETF standard authentication packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.8.2 Configuring the Source Address Check for RIP Packets on the Broadcast Network

By default, RIP checks the source addresses of the received packets to ensure that the local RIP interface receives only the packets from the same network.

Context

RIP interfaces perform the source address check on the received RIP packets. If the source address of the received RIP packet is at a different network segment from the IP address of the local interface, the local interface discards this RIP packet. This improves the network security.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
verify-source
```

The source address check is configured for RIP packets on the broadcast network.

By default, the source address check is enabled for RIP packets on the broadcast network.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

5.8.3 Checking the Configuration

After the security function of RIP is successfully configured, you can view information about RIP interfaces and the current running status of RIP.

Prerequisite

All configurations of improving security of a RIP network are complete.

Procedure

- Run the **display rip** *process-id* **interface** [*interface-type interface-number*] [**verbose**] command to check information about RIP interfaces.
- Run the **display rip** *process-id* command to check the running status and configuration of the RIP process.

----End

Example

Run the **display rip** *process-id* **interface verbose** command, and you can view the authentication mode configured for RIP packets. For example:

```
<HUAWEI> display rip 1 interface verbose
GigabitEthernet1/0/0 (1.1.1.2)
  State    : UP             MTU: 0
  Metricin : 0
  Metricout: 1
  Input    : Disabled        Output: Disabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast Packets
  Poison-reverse                : Enabled
  Split-Horizon                 : Enabled
  Authentication type           : Md5 Non Standard
  Max Packet Length             : 500
```

Run the **display rip** command. If the Verify-source field is displayed as **Enabled**, it indicates that the source address check is configured successfully.

```
<HUAWEI> display rip
  Public VPN-instance
    RIP process : 1
       RIP version : 1
       Preference : 100
       Checkzero : Enabled
       Default-cost : 0
       Summary : Enabled
       Hostroutes : Enabled
       Maximum number of balanced paths : 3
       Update time   : 30 sec  Age time  : 180 sec
       Suppress time : 0 sec   Garbage-collect time : 120 sec
       Silent interfaces : All
       Default Route : Disabled
       Verify-source : Enabled
       Networks :
       172.4.0.0
       Configured peers : None
       Number of routes in database : 4
       Number of interfaces enabled : 3
       Triggered updates sent : 3
       Number of route changes : 6
       Number of replies to queries : 1
       Description : RIP
```

5.9 Configuring the Network Management Function in RIP

By binding RIP and MIBs, you can view and configure RIP through the NMS.

Applicable Environment

Through the Simple Network Management Protocol (SNMP), the RIP Management Information Base (MIB) manages multicast information exchanged between the NMS and agents.

Pre-configuration Tasks

Before controlling RIP configuration through an SNMP agent, you need to complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **5.3 Configuring Basic RIP Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

rip mib-binding *process-id*

The MIB is bound to the RIP process ID, and the ID of the RIP process that accepts SNMP requests is specified.

Step 3 Run:

commit

The configuration is submitted.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display current-configuration** command to check the set parameters that are valid on the device.

```
<HUAWEI> display current-configuration
ip route-static 129.1.0.0 255.255.0.0 nu0
 snmp-agent trap enable
 snmp-agent local-engineid 800007DB017F000001
 snmp-agent sys-info version all
 rip mib-binding 1
 interface GigabitEthernet3/0/0
 undo shutdown
 ip address 75.75.75.2 255.255.255.0
 undo rip input
 interface Ethernet3/0/1
 undo shutdown
#
ip vpn-instance public
 apply-label per-instance
rip 1
 description RIP_Network1
 network 75.0.0.0
 import-route static cost 0
```

5.10 Maintaining RIP

RIP maintenance is implemented through debugging. You need to note that debugging affects system performance.

5.10.1 Clearing RIP

After confirming that information about RIP needs to be cleared, you can run the reset commands in the user view.

Context

CAUTION

RIP information cannot be restored after you clear it. So, confirm the action before you use the command.

Procedure

- Run the **reset rip** { *process-id* | **all** } **configuration** command to reset the configuration parameters of the specific RIP process. When the RIP process is initiated, the default configuration parameters are restored to the default values.
- Run the **reset rip** { *process-id* | **all** } **imported-routes** command to clear the routes imported from other routing protocols and then import these routes into RIP again.
- Run the **reset rip** { *process-id* | **all** } **statistics** [**interface** *interface-type interface-number*] command to clear the statistics of the counter that is maintained by a particular RIP process. This command helps to re-collect statistics during debugging.

----End

5.10.2 Debugging RIP

Routers can generate associated debugging information after you enable the debugging of modules in the user view. Debugging information displays the contents of the packets sent or received by the debugged modules.

Context

CAUTION

Debugging affects the performance of the system. So, after debugging, run the **undo debugging all** command to disable it immediately.

When a RIP fault occurs, run the following **debugging** command in the user view to debug RIP and locate the fault.

For the procedure of displaying the debugging information, refer to the chapter "Information Center Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - System Management*.

Procedure

Step 1 Run the **debugging rip** *process-id* **packet** [**send** | **receive**] [**error**] [**verbose**] [**interface** *interface-type interface-number* [**peer** *peer-address*]] command in the user view to debug the RIP packets sent or received on the network.

Step 2 Run the **debugging rip** *process-id* **route** [**error** | **backup**] [**imported** | { **interface** *interface-type interface-number* [**peer** *peer-address*] }] command in the user view to debug RIP routes.

Step 3 Run the **debugging rip miscellaneous** command in the user view to debug RIP packets globally.

----End

5.11 Configuration Examples

This section provides configuration examples of RIP. You can understand the configuration procedures through the configuration flowchart. Each configuration example consists of such information as the networking requirements, configuration notes, and conf

5.11.1 Example for Configuring Basic RIP Functions

This section describes how to configure basic RIP functions, including how to enable RIP and configure the RIP version number on each device.

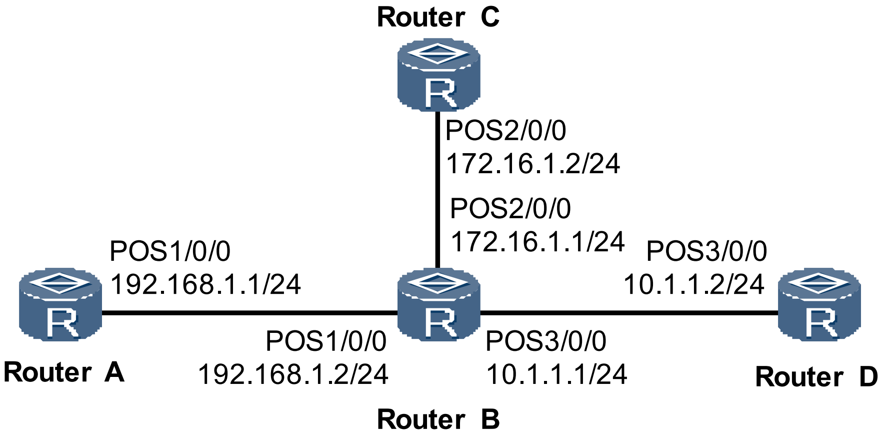
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 5-5**, it is required that RIP be enabled on all interfaces of Router A, Router B, Router C, and Router D and the Routers communicate with each other through RIP-2.

Figure 5-5 Networking diagram of configuring basic RIP functions



Configuration Notes

During the configuration, pay attention to the following:

- RIP does not support the enabling of different network segments on the same physical interface for different RIP processes.
- If a RIP process is bound to a VPN instance, the interfaces in this RIP process also need to be bound to this VPN instance.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer.
2. Enable RIP on each router and configure basic RIP functions.
3. Configure RIP-2 on each router and view the subnet masks.

Data Preparation

To complete the configuration, you need the following data:

- RIP network segment 192.168.1.0 on Router A
- RIP network segments 192.168.1.0, 172.16.0.0, and 10.0.0.0 on Router B
- RIP network segment 172.16.0.0 on Router C
- RIP network segment 10.0.0.0 on Router D
- RIP-2 on Router A, Router B, Router C, and Router D

Procedure

Step 1 Configure an IP address for each interface.

The configuration details are not mentioned here.

Step 2 Configure the network segments to be enabled with RIP.

# Configure Router A.

```
[~RouterA] rip
[~RouterA-rip-1] network 192.168.1.0
[~RouterA-rip-1] commit
[~RouterA-rip-1] quit
```

# Configure Router B.

```
[~RouterB] rip
[~RouterB-rip-1] network 192.168.1.0
[~RouterB-rip-1] network 172.16.0.0
[~RouterB-rip-1] network 10.0.0.0
[~RouterB-rip-1] commit
[~RouterB-rip-1] quit
```

# Configure Router C.

```
[~RouterC] rip
[~RouterC-rip-1] network 172.16.0.0
[~RouterC-rip-1] commit
[~RouterC-rip-1] quit
```

# Configure Router D.

```
[~RouterD] rip
[~RouterD-rip-1] network 10.0.0.0
```

```
[~RouterD-rip-1] commit
[~RouterD-rip-1] quit
```

# Check the RIP routing table of Router A.

```
[~RouterA] display rip 1 route
 Route Flags: R - RIP, T - TRIP
              P - Permanent, A - Aging, S - Suppressed, G - Garbage-collect
-------------------------------------------------------------------------
 Peer 192.168.1.2  on Pos1/0/0
      Destination/Mask        Nexthop     Cost   Tag     Flags   Sec
         10.0.0.0/8        192.168.1.2      1    0        RA      14
        172.16.0.0/16      192.168.1.2      1    0        RA      14
        192.168.1.0/24     192.168.1.2      1    0        RA      14
```

From the routing table, you can view that the routes advertised by RIP-1 use natural masks.

Step 3 Configure the RIP version number.

# Configure RIP-2 on Router A.

```
[~RouterA] rip
[~RouterA-rip-1] version 2
[~RouterA-rip-1] commit
[~RouterA-rip-1] quit
```

# Configure RIP-2 on Router B.

```
[~RouterB] rip
[~RouterB-rip-1] version 2
[~RouterB-rip-1] commit
[~RouterB-rip-1] quit
```

# Configure RIP-2 on Router C.

```
[~RouterC] rip
[~RouterC-rip-1] version 2
[~RouterC-rip-1] commit
[~RouterC-rip-1] quit
```

# Configure RIP-2 on Router D.

```
[~RouterD] rip
[~RouterD-rip-1] version 2
[~RouterD-rip-1] commit
[~RouterD-rip-1] quit
```

Step 4 Verify the configuration.

# Check the RIP routing table of Router A.

```
[~RouterA] display rip 1 route
Route Flags: R - RIP
          A - Aging, S - Suppressed, G - Garbage-collect
-------------------------------------------------------------------------
 Peer 192.168.1.2  on Pos1/0/0
      Destination/Mask        Nexthop       Cost   Tag     Flags   Sec
         10.1.1.0/24         192.168.1.2      1    0        RA      32
        172.16.1.0/24        192.168.1.2      1    0        RA      32
        192.168.1.0/24       192.168.1.2      1    0        RA      14
```

From the routing table, you can view that the routes advertised by RIP-2 contain accurate subnet masks.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
#
rip 1
 version 2
 network 192.168.1.0
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 172.16.1.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
#
rip 1
 version 2
 network 10.0.0.0
 network 172.16.0.0
 network 192.168.1.0
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 172.16.1.2 255.255.255.0
#
rip 1
 version 2
 network 172.16.0.0
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
#
rip 1
```

```
 version 2
 network 10.0.0.0
#
return
```

Related Tasks

5.3 Configuring Basic RIP Functions

5.11.2 Example for Preventing Routing Loops

This section takes split horizon as an example to describe how to prevent routing loops.

Networking Requirements

CAUTION

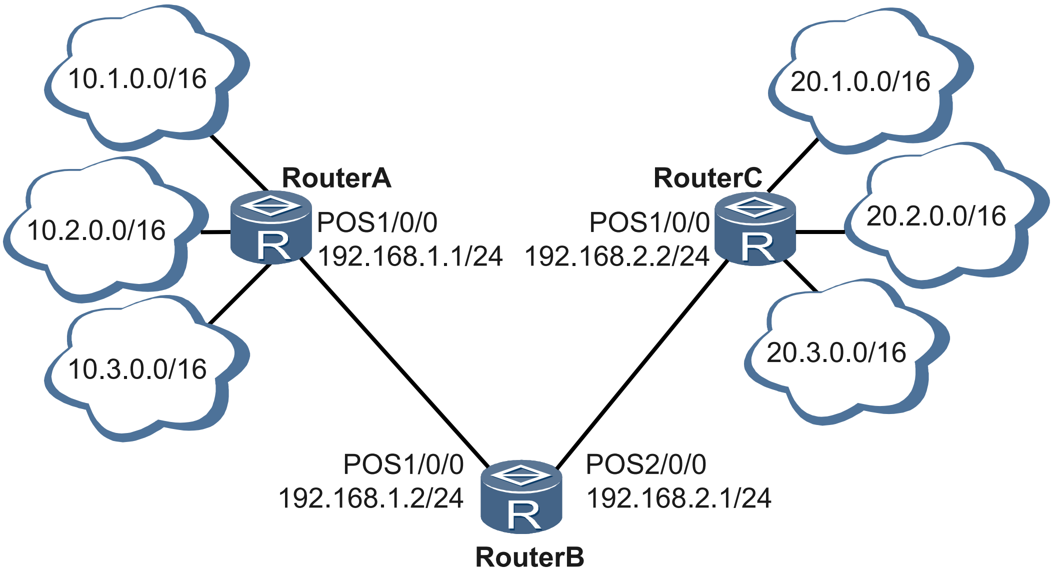
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 5-6**, IP addresses are assigned to all interfaces on each router; RIP-2 is configured on each router and the RIP service runs normally. Classful summarization is enabled on router A and router C. In this case, it is required that split horizon be configured on router A and router C.

NOTE

On a RIP-2 network, if classful summarization is configured, you need to disable split horizon and poison reverse. After classful summarization is disabled, you need to reconfigure split horizon or poison reverse to prevent routing loops. This configuration example involves the configuration of split horizon.

Figure 5-6 Networking diagram of preventing routing loops



Configuration Notes

When preventing routing loops, pay attention to the following:

- By default, split horizon is enabled.
- When both split horizon and poison reverse are configured, only poison reverse takes effect.

Configuration Roadmap

The configuration roadmap is as follows:

1. Disable route summarization.
2. Enable split horizon.

Data Preparation

None.

Procedure

Step 1 Disable route summarization.

# Disable route summarization on Router A.

```
[~RouterA] rip 1
[~RouterA-rip-1] undo summary
[~RouterA-rip-1] commit
[~RouterA-rip-1] quit
```

# Disable route summarization on Router B.

```
[~RouterB] rip 1
[~RouterB-rip-1] undo summary
[~RouterB-rip-1] commit
[~RouterB-rip-1] quit
```

Step 2 Configure split horizon.

Configure split horizon on the RIP interfaces of all routers. The configurations of Router B and Router C are the same as the configuration of Router A, and are not mentioned here.

# Configure Router A.

```
[~RouterA] interface pos1/0/0
[~RouterA-Pos1/0/0] rip split-horizon
[~RouterA-Pos1/0/0] quit
[~RouterA] interface pos1/0/1
[~RouterA-Pos1/0/1] rip split-horizon
[~RouterA-Pos1/0/1] quit
[~RouterA] interface pos1/0/2
[~RouterA-Pos1/0/2] rip split-horizon
[~RouterA-Pos1/0/2] quit
[~RouterA] interface pos1/0/3
[~RouterA-Pos1/0/3] rip split-horizon
[~RouterA-Pos1/0/3] quit
[~RouterA] commit
```

Step 3 Verify the configuration.

# Run the **display rip 1 interface verbose** command on Router A and Router C. You can check whether split horizon is enabled. Take the display on Router A as an example. If the Split-Horizon field is displayed as **Enabled**, it indicates that split horizon is enabled.

```
[~RouterA] display rip 1 interface verbose
 Pos1/0/0(192.168.1.1)
  State    : DOWN            MTU: 500
  Metricin : 0
  Metricout: 1
  Input    : Enabled         Output       : Enabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast and Broadcast Packets
  Poison-reverse                : Disabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Replay Protection             : Disabled
 Pos1/0/1(10.1.1.1)
  State    : DOWN            MTU: 500
  Metricin : 0
  Metricout: 1
  Input    : Enabled         Output       : Enabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast and Broadcast Packets
  Poison-reverse                : Disabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Replay Protection             : Disabled
 Pos1/0/2(10.2.1.1)
  State    : DOWN            MTU: 500
  Metricin : 0
  Metricout: 1
  Input    : Enabled         Output       : Enabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast and Broadcast Packets
  Poison-reverse                : Disabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Replay Protection             : Disabled
 Pos1/0/3(10.3.1.1)
  State    : DOWN            MTU: 500
  Metricin : 0
  Metricout: 1
  Input    : Enabled         Output       : Enabled
  Protocol : RIPv2 Multicast
  Send     : RIPv2 Multicast Packets
  Receive  : RIPv2 Multicast and Broadcast Packets
  Poison-reverse                : Disabled
  Split-Horizon                 : Enabled
  Authentication type           : None
  Replay Protection             : Disabled
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 ip address 192.168.1.1 255.255.255.0
 rip version 2 multicast
#
interface Pos1/0/1
 undo shutdown
 ip address 10.1.1.1 255.255.0.0
 rip version 2 multicast
#
```

```
interface Pos1/0/2
 undo shutdown
 ip address 10.2.1.1 255.255.0.0
 rip version 2 multicast
#
interface Pos1/0/3
 undo shutdown
 ip address 10.3.1.1 255.255.0.0
 rip version 2 multicast
#
rip 1
 network 10.1.0.0
 network 10.2.0.0
 network 10.3.0.0
 network 192.0.0.0
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 ip address 192.168.1.2 255.255.255.0
 rip version 2 multicast
#
interface Pos2/0/1
 undo shutdown
 ip address 192.168.2.1 255.255.255.0
 rip version 2 multicast
#
rip 1
 network 192.0.0.0
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 ip address 192.168.2.2 255.255.255.0
 rip version 2 multicast
#
interface Pos1/0/1
 undo shutdown
 ip address 20.1.1.1 255.255.0.0
 rip version 2 multicast
#
interface Pos1/0/2
 undo shutdown
 ip address 20.2.1.1 255.255.0.0
 rip version 2 multicast
#
interface Pos1/0/3
 undo shutdown
 ip address 20.3.1.1 255.255.0.0
 rip version 2 multicast
#
rip 1
 network 20.1.0.0
 network 20.2.0.0
 network 20.3.0.0
 network 192.0.0.0
return
```

Related Tasks

5.4 Preventing Routing Loops

5.11.3 Example for Configuring RIP to Import External Routes

This section describes how to configure RIP to import external routes to increase the number of routes in the RIP routing table.

Networking Requirements

CAUTION

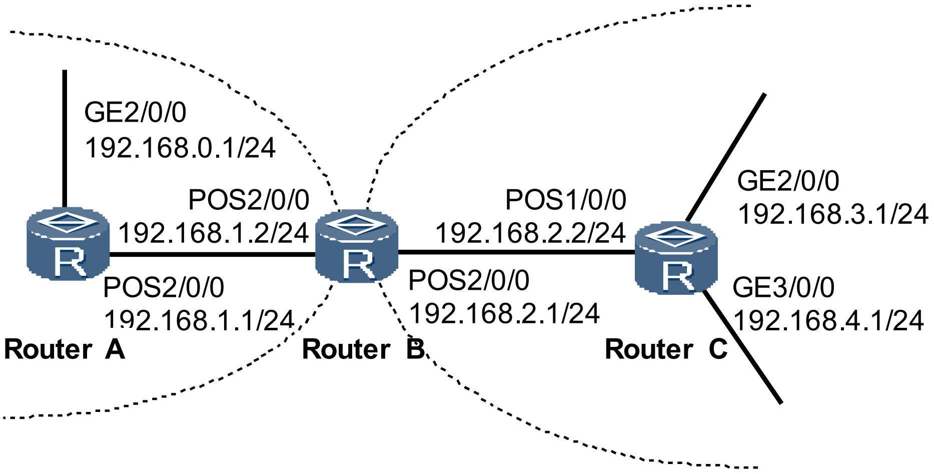
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 5-7**, two RIP processes, RIP 100 and RIP 200, run on Router B. Router B exchanges routing information with Router A through RIP 100. Router B exchanges routing information with Router C through RIP 200.

It is required that the two RIP processes of Router B import the RIP routes from each other. The cost of the routes imported from RIP 200 defaults to 3.

It is required that a filtering policy be configured on Router B to filter the route 192.168.4.0/24 imported from RIP 200, thus preventing the route from being advertised to Router A.

Figure 5-7 Networking diagram of configuring RIP to import external routes



Configuration Notes

You can run one of the following commands to set the cost of the imported route. The following commands are listed in descending order of priorities:

- Run the **apply cost** command to set the cost of a route.
- Run the **import-route** (RIP) command to set the cost of the imported route.
- Run the **default-cost** (RIP) command to set the cost of the default route.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable RIP 100 and RIP 200 on each router and specify network segments.
2. Configure the two RIP processes on Router B to import routes from each other and set the default cost of the routes imported from RIP 200 to 3.
3. Configure an ACL on Router B to filter the routes imported from RIP 200.

Data Preparation

To complete the configuration, you need the following data:

- RIP network segments 192.168.0.0 and 192.168.1.0 on Router A
- RIP network segments 192.168.1.0 and 192.168.2.0 on Router B
- RIP network segments 192.168.2.0, 192.168.3.0, and 192.168.4.0 on Router C

Procedure

Step 1 Configure an IP address for each interface.

The configuration details are not mentioned here.

Step 2 Configure basic RIP functions.

# Enable RIP process 100 on Router A.

```
[~RouterA] rip 100
[~RouterA-rip-100] network 192.168.0.0
[~RouterA-rip-100] network 192.168.1.0
[~RouterA-rip-100] commit
[~RouterA-rip-100] quit
```

# Enable the two RIP processes, RIP 100 and RIP 200, on Router B.

```
[~RouterB] rip 100
[~RouterB-rip-100] network 192.168.1.0
[~RouterB-rip-100] quit
[~RouterB] rip 200
[~RouterB-rip-200] network 192.168.2.0
[~RouterB-rip-200] commit
[~RouterB-rip-200] quit
```

# Enable RIP process 200 on Router C.

```
[~RouterC] rip 200
[~RouterC-rip-200] network 192.168.2.0
[~RouterC-rip-200] network 192.168.3.0
[~RouterC-rip-200] network 192.168.4.0
[~RouterC-rip-200] commit
[~RouterC-rip-200] quit
```

# Check the routing table of Router A.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 7       Routes : 7
Destination/Mask    Proto  Pre  Cost   Flags    NextHop        Interface
      127.0.0.0/8   Direct 0    0        D      127.0.0.1      InLoopBack0
      127.0.0.1/32  Direct 0    0        D      127.0.0.1      InLoopBack0
    192.168.0.0/24  Direct 0    0        D     192.168.0.1    GigabitEthernet2/0/0
    192.168.0.1/32  Direct 0    0        D      127.0.0.1       InLoopBack0
    192.168.1.0/24  Direct 0    0        D      192.168.1.1     Pos1/0/0
    192.168.1.1/32  Direct 0    0        D      127.0.0.1       InLoopBack0
    192.168.1.2/32  Direct 0    0        D      192.168.1.2     Pos1/0/0
```

You can view that there are no routes of other processes in the routing table of Router A.

Step 3 Configure RIP to import external routes.

# Set the default route cost to 3 on Router B and import the routes of the two RIP processes into the routing table of each other.

```
[~RouterB] rip 100
[~RouterB-rip-100] default-cost 3
[~RouterB-rip-100] import-route rip 200
[~RouterB-rip-100] quit
[~RouterB] rip 200
[~RouterB-rip-200] import-route rip 100
[~RouterB-rip-200] commit
[~RouterB-rip-200] quit
```

# Check the routing table of Router A after the routes are imported.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 10       Routes : 10
Destination/Mask    Proto  Pre  Cost   Flags    NextHop         Interface
      127.0.0.0/8   Direct 0    0       D         127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0       D         127.0.0.1       InLoopBack0
   192.168.0.0/24  Direct 0    0      D        192.168.0.1    GigabitEthernet2/0/0
    192.168.0.1/32  Direct 0    0       D          127.0.0.1       InLoopBack0
    192.168.1.0/24  Direct 0    0       D          192.168.1.1     Pos1/0/0
    192.168.1.1/32  Direct 0    0       D          127.0.0.1       InLoopBack0
    192.168.1.2/32  Direct 0    0       D          192.168.1.2     Pos1/0/0
    192.168.2.0/24  RIP    100  4       D          192.168.1.2     Pos1/0/0
    192.168.3.0/24  RIP    100  4       D          192.168.1.2     Pos1/0/0
    192.168.4.0/24  RIP    100  4       D          192.168.1.2     Pos1/0/0
```

You can view that information about routes to 192.168.2.0/24, 192.168.3.0/24, and 192.168.3.0/24 is displayed in the RIP routing table of Router A. These new routes are learnt through RIP 200 on Router B.

Step 4 Configure RIP to filter the imported routes.

# Configure an ACL on Router B and set a rule to deny the packets with the source address being 192.168.4.0/24.

```
[~RouterB] acl 2000
[~RouterB-acl-basic-2000] rule deny source 192.168.4.0 0.0.0.255
[~RouterB-acl-basic-2000] rule permit
[~RouterB-acl-basic-2000] quit
```

# Filter out the route 192.168.4.0/24 imported from RIP 200 on Router B according to the ACL rule.

```
[~RouterB] rip 100
[~RouterB-rip-100] filter-policy 2000 export
[~RouterB-rip-100] quit
[~RouterB] commit
```

Step 5 Verify the configuration.

# Check the routing table of Router A after the filtering.

```
[~RouterA] display ip routing-table
Route Flags: R - relay, D - download to fib
```

```
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 9        Routes : 9
Destination/Mask    Proto  Pre  Cost   Flags    NextHop         Interface
      127.0.0.0/8   Direct 0    0       D       127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0       D       127.0.0.1       InLoopBack0
    192.168.0.0/24  Direct 0    0       D       192.168.0.1     GigabitEthernet2/0/0
    192.168.0.1/32  Direct 0    0       D       127.0.0.1       InLoopBack0
    192.168.1.0/24  Direct 0    0       D       192.168.1.1     Pos1/0/0
    192.168.1.1/32  Direct 0    0       D       127.0.0.1       InLoopBack0
    192.168.1.2/32  Direct 0    0       D       192.168.1.2     Pos1/0/0
    192.168.2.0/24  RIP    100  4       D       192.168.1.2     Pos1/0/0
    192.168.3.0/24  RIP    100  4       D       192.168.1.2     Pos1/0/0
```

You can view that the RIP routing table of Router A is changed that the route with the source address being 192.168.4.0/24 is rejected.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.0.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
#
rip 100
 network 192.168.0.0
 network 192.168.1.0
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
acl number 2000
 rule 5 deny source 192.168.4.0 0.0.0.255
 rule 10 permit
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.1 255.255.255.0
#
rip 100
 default-cost 3
 network 192.168.1.0
 filter-policy 2000 export
 import-route rip 200
#
rip 200
 network 192.168.2.0
 import-route rip 100
#
```

```
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 192.168.3.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 192.168.4.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.2 255.255.255.0
#
rip 200
 network 192.168.2.0
 network 192.168.3.0
 network 192.168.4.0
#
return
```

Related Tasks

5.6.1 Configuring RIP to Import External Routes

6 RIPng Configuration

About This Chapter

RIPng is an extension of RIP for support of IPv6.

6.1 RIPng Overview

RIPng is a distance-vector routing protocol, which measures the distance to the destination host by the hop count.

The Routing Information Protocol Next Generation (RIPng) protocol is an extension of RIPv2 that is applied to IPv4 networks. Most RIP-related concepts are applicable to RIPng.

Extension of RIP

For IPv6 applications, RIPng extends RIP as follows:

- UDP port number: In RIPng, UDP port number 521 is used to send and receive routing information.
- Multicast group address: In RIPng, FF02::9 is used as the multicast group address of RIPng devices.
- Prefix length: In RIPng, the prefix length of a destination address is 128 bits (the mask length).
- Next-hop address: In RIPng, a next-hop address is a 128-bit IPv6 address.
- Source address: In RIPng, link-local address FE80::/10 is used as the source address to send RIPng Update packets.

Operation Principle of RIPng

RIPng employs the hop count to measure the distance to the destination. The distance is called the routing metric. In RIPng, the hop count from a router to its directly connected network is 0, and the hop count from a router to a network, which can be reached through another router, is 1. The hop count that is equal to or exceeds 16 is defined as infinity, indicating that the destination network or host is unreachable.

By default, RIPng sends an Update packet every 30 seconds. If no Update packet is received from a neighbor in 180 seconds, all the routes learned from the neighbor are marked as unreachable. After these routes are marked as unreachable, RIPng does not delete them from the routing table immediately but starts a Garbage-collect timer. If no Update packet is received in 120 seconds, RIPng deletes these routes from the routing table.

To improve the performance and prevent routing loops, RIPng supports both split horizon and poison reverse. In addition, RIPng can import routes from other routing protocols.

6.2 RIPng Features Supported by the NE5000E

In particular networking environments, you are required to configure certain RIPng features to optimize the performance of a RIPng network.

The NE5000E supports the following RIPng features:

- Routing policies: When advertising, receiving, or importing routes, a NE5000E may need to implement certain routing policies to filter routing information or change route attributes.

- Importing external routes: RIPng can import routes from other IGPs such as OSPF and IS-IS. Direct routes, static routes, and Border Gateway Protocol (BGP) routes can also be imported into RIPng.
- RIPng multi-instance: On the NE5000Es that support VPN, each RIPng process can be associated with a specified VPN instance. Then all the interfaces that run this RIPng process are associated with the VPN instance.

6.3 Configuring Basic RIPng Functions

Before using RIPng features, you need to configure basic RIPng functions, including creating RIPng processes and enabling RIPng on interfaces.

Applicable Environment

RIPng is simple and easy to use but is less powerful in functions than OSPFv3 and IS-IS. Therefore, RIPng is widely used on small-scale networks.

Configuring basic RIPng functions is a prerequisite for building RIPng networks.

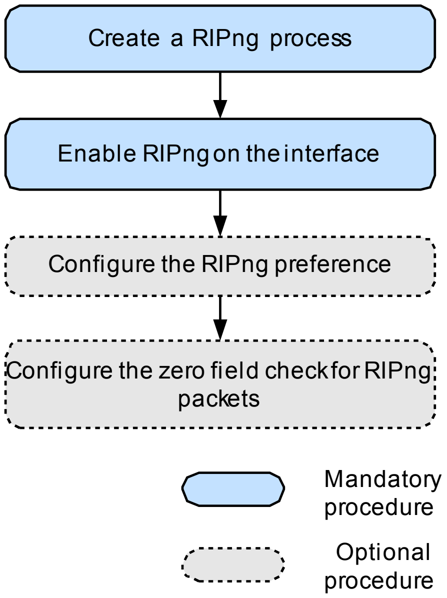
Pre-configuration Tasks

Before configuring basic RIPng functions, complete the following tasks:

- Configuring a link layer protocol
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- Enabling IPv6 in the system view of the router

Configuration Procedure

Figure 6-1 Flowchart of configuring basic RIPng functions


Create a RIPng process
Enable RIPng on the interface
Configure the RIPng preference
Configure the zero field check for RIPng packets
Mandatory procedure
Optional procedure

Related Tasks

6.9.1 Example for Configuring Basic RIPng Functions

6.3.1 Creating a RIPng process

Creating RIP processes is the prerequisite to performing RIP configurations.

Applicable Environment

Before running the RIPng protocol on a router, you need to create a RIPng process on the router.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

RIPng supports multi-instance. To bind RIPng processes to VPN instances, you can run the **ripng** [*process-id*] **vpn-instance** *vpn-instance-name* command.

NOTE

If you run RIPng-related commands in the interface view before enabling RIPng, the configurations take effect only after RIPng is enabled.

Step 3 Run:

```
commit
```

The configuration is submitted.

----End

6.3.2 Enabling RIPng on the Interface

After an interface is associated with a RIPng process, routing information on this interface can be exchanged through RIPng.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

The interface is at the network side of the router. That is, the router is connected to other devices through this interface. To enable the router to learn the routes to the network segment where this interface resides, ensure that the link status of the interface is Up.

Step 3 Run:

```
ripng process-id enable
```

RIPng is enabled on the specified interface.

NOTE

- In the interface view, this command cannot be executed if IPv6 is not enabled on the interface.
- If a router is connected to other devices through multiple interfaces, repeatedly perform Step 2 and Step 3.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.3.3 (Optional) Configuring the RIPng Preference

When there are routes discovered by multiple routing protocols on the same device, you can make the device prefer RIPng routes by setting the RIPng preference.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

Step 3 Run:

```
preference { route-policy route-policy-name | preference } *
```

The RIPng preference is set.

The **preference** command can be used together with the routing policy to set the preference for the routes that meet the matching conditions.

After RIPng routes are delivered to the Routing Management (RM), if the RIPng preference changes, the RM updates the routing table.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.3.4 (Optional) Configuring the Zero Field Check for RIPng Packets

In a RIPng packet, certain fields, which must be 0, are called zero fields. By default, RIPng checks the zero fields of the received packets, and discards the packets whose zero fields are not 0.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

Step 3 Run:

```
checkzero
```

The zero field check is configured for RIPng packets.

By default, the zero field check is enabled for RIPng packets.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.3.5 Checking the Configuration

After basic RIPng functions are configured, you can view the current operation status and routing information of RIPng.

Prerequisite

All configurations of basic RIPng functions are complete.

Procedure

- Run the **display ripng** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the operation status and configuration of RIPng.
- Run the **display ripng***process-id* **route** [**destination-address** *destination-address* [*mask-length*]] [**interface** *interface-type interface-number* [**neighbor-address** *neighbor-address*]] command to check RIPng routes.

----End

Example

Run the **display ripng** command, and you can view the operation status and configuration of RIPng, including the RIPng process ID, preference, and whether the zero field check is enabled.

```
<HUAWEI> display ripng
  Public vpn-instance name :
    RIPng process : 100
       Preference : 100
       Checkzero : Enabled
       Default Cost : 0
       Maximum number of balanced paths : 6
       Update time   : 30 sec   Age time  : 180 sec
       Suppress time : 0 sec    Garbage-Collect time : 120 sec
       Number of periodic updates sent : 0
       Number of trigger updates sent : 1
       Number of routes in database : 1
       Number of interfaces enabled : 1
```

Run the **display ripng** *process-id* **route** command, and you can view all routes of the specified RIPng process.

```
<HUAWEI> display ripng 100 route
   Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
 ----------------------------------------------------------------
 Peer FE80::200:5EFF:FE04:B602  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Dest 3FFE:C00:C18:2::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Peer FE80::200:5EFF:FE04:B601  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
 Dest 3FFE:C00:C18:3::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
```

6.4 Preventing Routing Loops

RIPng is a distance vector routing protocol. Because RIPng devices advertise their routing tables to their neighbors, routing loops may occur.

Applicable Environment

RIPng prevents routing loops by using the following mechanisms:

- Counting to infinity: RIPng defines the cost of 16 as infinity. If routing loops occur, a route is considered unreachable when its cost reaches 16.
- Split horizon: RIPng does not send the routes learned from an interface to neighbors through this interface. This reduces bandwidth consumption and prevents routing loops.
- Poison reverse: After learning a route from an interface, RIPng sets the cost of the route to 16 (indicating that the route is unreachable), and then sends the route to neighbors through this interface. This clears unnecessary information in the routing tables of the neighbors.
- Suppress timer: A Suppress timer can prevent routing loops and reduce the possibility that receiving incorrect routes results in incorrect routing information.

NOTE

If both split horizon and poison reverse are configured, only poison reverse takes effect.

Pre-configuration Tasks

Before configuring RIPng to prevent routing loops on the network, complete the following tasks:

- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer

- **6.3 Configuring Basic RIPng Functions**

Configuration Procedure

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

Related Tasks

6.9.2 Example for Configuring Split Horizon

6.4.1 Configuring Split Horizon

You can configure split horizon to prevent routing loops.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ripng split-horizon
```

Split horizon is enabled.

If both split horizon and poison reverse are configured, only poison reverse takes effect.

If an interface is connected to an NBMA network, by default, split horizon is disabled on this interface.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.4.2 Configuring Poison Reverse

You can configure poison reverse to prevent routing loops.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

If both split horizon and poison reverse are configured, only poison reverse takes effect.

On an NBMA network, poison reverse is enabled on interfaces by default.

Step 3 Run:

```
ripng poison-reverse
```

Poison reverse is enabled.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.4.3 Configuring the Suppression Time

A Suppress timer can prevent routing loops and reduce the possibility that receiving incorrect routes results in incorrect routing information.

Context

When receiving an Update packet with the increased hop count, a device starts a Suppress timer. The device does not update routes in the routing table before the Suppress timer expires. The device begins to accept the Update packet until the Suppress timer expires.

By configuring a Suppress timer, you can delay adding incorrect routes to the routing table. This, however, also delays route convergence on the entire network. Therefore, you need to confirm the action before configuring the Suppress timer.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng process-id
```

The RIPng process is created and the RIPng view is displayed.

Step 3 Run:

```
timers ripng update age suppress garbage-collect
```

The suppression time is set.

By default, the value of a Suppress timer is 0 seconds.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

RIPng has four timers: the Update timer, Age timer, Suppress timer, and Garbage-collect timer. The relationship between the values of the four timers is as follows: *update* < *age* and *suppress* < *garbage-collect*. Changing the values of the timers affects the RIPng convergence speed and even causes route flapping on the network. For example, when the update time is longer than the aging time, if a RIPng route changes within the update time, a router cannot inform its neighbors of the change in time.

For the configurations of the Update timer, Age timer, and Garbage-collect timer, see **6.7 Configuring RIPng Timers**.

6.4.4 Checking the Configuration

After the prevention of routing loops is configured, you can view the current operation status, interface information, and routing information of RIPng.

Prerequisite

All configurations of preventing routing loops are complete.

Procedure

- Run the **display ripng** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the current operation status and configuration of RIPng.
- Run the **display ripng** *process-id* **route** [**destination-address** *destination-address* [*mask-length*]] [**interface** *interface-type interface-number* [**neighbor-address** *neighbor-address*]] command to check RIPng routes.
- Run the **display ripng** *process-id* **interface** command to check information about RIPng interfaces.

----End

Example

Run the **display ripng** command, and you can view the current operation status and configuration of the RIPng process.

```
<HUAWEI> display ripng
  Public vpn-instance name :
    RIPng process : 100
       Preference : 100
       Checkzero : Enabled
       Default Cost : 0
       Maximum number of balanced paths : 6
       Update time    : 30 sec    Age time   : 180 sec
       Suppress time : 0 sec     Garbage-Collect time : 120 sec
       Number of periodic updates sent : 0
       Number of trigger updates sent : 1
       Number of routes in database : 1
       Number of interfaces enabled : 1
```

Run the **display ripng** *process-id* **route** command, and you can view all the routes learned by the RIPng process.

```
<HUAWEI> display ripng 100 route
   Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
 -----------------------------------------------------------------
 Peer FE80::200:5EFF:FE04:B602  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Dest 3FFE:C00:C18:2::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Peer FE80::200:5EFF:FE04:B601  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
 Dest 3FFE:C00:C18:3::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
```

Run the **display ripng** *process-id* **interface verbose** command, and you can view information about the RIPng interface, including whether split horizon and poison reverse are enabled.

```
<HUAWEI> display ripng 1 interface verbose
 GigabitEthernet1/0/0
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
```

6.5 Adjusting RIPng Route Selection

You can adjust RIPng route selection on a complicated network.

Applicable Environment

To flexibly apply RIPng on the current network and meet various requirements of users, you can change RIPng route selection by setting different parameters.

Pre-configuration Tasks

Before adjusting RIPng route selection, complete the following tasks:

- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **6.3 Configuring Basic RIPng Functions**

Configuration Procedure

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

6.5.1 Configuring the Additional Metric on an Interface

The additional metric is the metric (hop count) to be added to the original metric of a RIPng route. You can set additional metrics for received and sent RIPng routes by using different commands.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ripng metricin value
```

The metric to be added to a received route is set.

Step 4 Run:

```
ripng metricout { value | { acl-number | acl-name acl-name | ipv6-prefix ipv6-prefix-name } value1 } *
```

The metric to be added to a sent route is set.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

After the **ripng metricin** command is run, RIPng adds the additional metric to the received route and then adds the route to the routing table. Therefore, after the receiving metric of an interface is increased, the metric of the RIPng route received by this interface increases accordingly.

After the **ripng metricout** command is run, RIPng adds the additional metric to the route to be advertised. Therefore, after the sending metric of an interface is increased, the metric of the RIPng route sent by this interface increases accordingly. In the routing table, however, the metric of the route remains changed.

If the metric of a route exceeds 16 after the additional metric is added, the metric is still calculated as 16.

6.5.2 Setting the Maximum Number of Equal-Cost Routes

By setting the maximum number of equal-cost RIPng routes, you can change the number of routes that participate in load balancing.

Procedure

Step 1 Run:

```
interface
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

Step 3 Run:

```
maximum load-balancing number
```

The maximum number of equal-cost routes is set.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

6.5.3 Checking the Configuration

After RIPng route selection is adjusted successfully, you can view the current operation status, interface information, and routing information of RIPng.

Prerequisite

All configurations of adjusting RIPng route selection are complete.

Procedure

- Run the **display ripng** [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the current operation status and configuration of RIPng.
- Run the **display ripng** *process-id* **route** [**destination-address** *destination-address* [*mask-length*]] [**interface** *interface-type interface-number* [**neighbor-address** *neighbor-address*]] command to check RIPng routes.
- Run the **display ripng** *process-id* **interface** command to check information about RIPng interfaces.

----End

Example

Run the **display ripng** *process-id* command, and you can view the operation status and configuration of the RIPng process.

```
<HUAWEI> display ripng
  Public vpn-instance name :
    RIPng process : 100
       Preference : 100
       Checkzero : Enabled
       Default Cost : 0
       Maximum number of balanced paths : 6
       Update time   : 30 sec   Age time  : 180 sec
       Suppress time : 0 sec    Garbage-Collect time : 120 sec
       Number of periodic updates sent : 0
       Number of trigger updates sent : 1
       Number of routes in database : 1
       Number of interfaces enabled : 1
```

Run the **display ripng** *process-id* **interface verbose** command, and you can view the metric to be added to the received or sent route on the interface.

```
<HUAWEI> display ripng 1 interface verbose
 GigabitEthernet1/0/0
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
```

Run the **display ripng** *process-id* **route** command, and you can check whether there are routes for load balancing. For example, in the routing table, you can view that there are two equal-cost routes with the same destination address (10.10.10.10/32) and different next hops.

```
<HUAWEI> display ripng 100 route
   Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
 ----------------------------------------------------------------
 Peer FE80::200:5EFF:FE04:B602  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Dest 3FFE:C00:C18:2::/64,
     via FE80::200:5EFF:FE04:B602, cost  2, tag 0, A, 34 Sec
 Peer FE80::200:5EFF:FE04:B601  on GigabitEthernet6/0/0
 Dest 3FFE:C00:C18:1::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
 Dest 3FFE:C00:C18:3::/64,
     via FE80::200:5EFF:FE04:B601, cost  2, tag 0, A, 13 Sec
```

6.6 Controlling RIPng Routing Information

In practice, different protocols run on the same network. Therefore, you need to control routing information of every protocol to meet different networking requirements.

Applicable Environment

To meet the requirements of complex networking, it is required to accurately control the sending and receiving of RIPng routing information.

Pre-configuration Tasks

Before configuring a router to control the receiving of RIPng routing information, complete the following tasks:

- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **6.3 Configuring Basic RIPng Functions**

Configuration Procedure

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

6.6.1 Configuring RIPng to Import External Routes

RIPng can import routing information from other RIPng processes or other routing protocols to enrich the RIPng routing table.

Context

On a large-scale network, different routing protocols are configured for devices in different ASs. In this case, you need to configure RIPng to import the routes learned by other routing protocols.

If RIPng needs to advertise routing information of other RIPbg processes or other routing protocols (such as direct, static, RIPng, OSPFv3, IS-IS, or BGP), you can specify *protocol* to filter routing information. If *protocol* is not specified, routing information to be advertised can be filtered, including the imported routes and local RIPng routes (functioning as direct routes).

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

Step 3 (Optional) Run:

```
default-cost cost
```

The default cost is set for the imported routes.

If no cost is specified when external routes are imported, the default cost 0 is used.

Step 4 Run:

```
import-route protocol [ process-id ] [ cost cost | route-policy route-policy-name ]
*
```

External routes are imported.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

6.6.2 Configuring RIPng to Advertise the Default Routes

There are two methods of advertising RIPng default routes. You can configure a router to advertise RIPng default routes according to the actual networking. Additionally, you can specify the cost of the default routes to be advertised.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
ripng default-route { only | originate } [ cost cost-value | tag tag-value ] *
```

RIPng is configured to advertise a default route.

Configure RIPng to advertise the default routes according to the actual networking:

- **only**: advertises only IPv6 default routes (::/0) and suppress the advertisement of other routes.

- **originate**: advertises IPv6 default routes (::/0) without affecting the advertisement of other routes.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Follow-up Procedure

RIPng default routes are forcibly advertised in Update packets through the specified interface, regardless of whether these routes exist in the IPv6 routing table.

6.6.3 Checking the Configuration

After the receiving of RIPng routing information is successfully controlled, you can view all activated routes in the RIPng database.

Prerequisite

All configurations of controlling the receiving of RIPng routing information are complete.

Procedure

- Run the **display ripng** *process-id* **database** [**verbose**] [**destination-address** *destination-address* [*mask-length*]] [**interface** *interface-type interface-number* [**neighbor-address** *neighbor-address*]] command to check routes in the RIPng database.

----End

Example

Run the **display ripng** *process-id* **database** command, and you can view the default routes and imported routes. For example:

```
<HUAWEI> display ripng 100 database
   2001:7B::2:2A1:5DE/64,
        cost 4, Imported
   1:13::/120,
        cost 4, Imported
   1:32::/120,
        cost 4, Imported
   1:33::/120,
        cost 4, Imported
```

6.7 Configuring RIPng Timers

There are four RIPng timers, namely, the Update timer, Age timer, Suppress timer, and Garbage-collect timer. You can adjust the RIPng convergence speed by changing the values of the RIPng timers.

Applicable Environment

By default, the Update timer is 30s; the Age timer is 180s; the Suppress timer is 0s; and the Garbage-collect timer is 120s. The relationship between the values of the four timers is as follows: *update* < *age* and *suppress* < *garbage-collect*. Changing the values of the timers affects the RIPng convergence speed and even causes route flapping on the network. For example, when the update time is longer than the aging time, if a RIPng route changes within the update time, a device cannot inform its neighbors of the change in time. Configuring a Suppress timer can prevent routing loops. For details, see **6.4.3 Configuring the Suppression Time**.

Pre-configuration Tasks

Before configuring RIPng timers, complete the following tasks:

- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **6.3 Configuring Basic RIPng Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ripng [ process-id ]
```

The RIPng process is created and the RIPng view is displayed.

Step 3 Run:

```
timers ripng update age suppress garbage-collect
```

RIPng timers are configured.

By default, the Update timer is 30s; the Age timer is 180s; the Suppress timer is 0s; the Garbage-collect timer is four times the Update timer, namely, 120s.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display ripng** command to view the values of RIPng timers.

```
<HUAWEI> display ripng
  Public vpn-instance name :
    RIPng process : 100
       Preference : 100
       Checkzero : Enabled
       Default Cost : 0
       Maximum number of balanced paths : 6
       Update time    : 30 sec    Age time   : 180 sec
       Suppress time : 0 sec    Garbage-Collect time : 120 sec
       Number of periodic updates sent : 0
       Number of trigger updates sent : 1
       Number of routes in database : 1
       Number of interfaces enabled : 1
```

6.8 Maintaining RIPng

RIPng maintenance is implemented through debugging. Debugging, however, affects the system performance.

6.8.1 Clearing RIPng

To clear RIPng information, run the reset commands in the user view.

Context

CAUTION

RIPng information cannot be restored after you clear it. So, confirm the action before you use the command.

Procedure

- Run the **reset ripng** *process-id* **statistics** [**interface** [*interface-type interface-number* [neighbor [neighbor-ipv6-address]]]] command to clear the statistics of the counter that is maintained by a particular RIPng process. This command helps to re-collect statistics during debugging.

----End

6.8.2 Debugging RIPng

A device generates debugging information after you enable debugging of modules in the user view. Debugging information shows the contents of the packets sent or received by the debugged modules.

Context

CAUTION

Debugging affects the system performance. So, after debugging, run the **undo debugging all** command to disable it immediately.

When a RIPng fault occurs, run the following **debugging** command in the user view to debug RIPng and locate the fault.

Procedure

Step 1 Run the **debugging ripng** *process-id* [**error** | **event** | **job** | **backup** | **interface** *interface-type interface-number* | **packet** [*interface-type interface-number* [**neighbor** *neighbor-address*]]

[**route** *acl-number*] | **route-processing** [*acl-number*]] command in the user view to debug the specified RIPng process.

Step 2 Run the **debugging ripng miscellaneous** command in the user view to debug component-level RIPng information.

----End

6.9 Configuration Examples

This section provides configuration examples of RIPng, including networking requirements, configuration notes, and configuration roadmap.

6.9.1 Example for Configuring Basic RIPng Functions

This section describes how to configure basic RIPng functions.

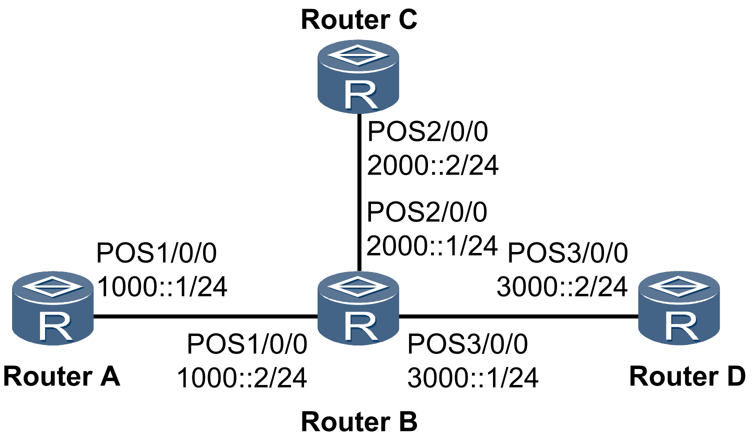
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 6-2**, it is required that RIPng be enabled on all interfaces of Router A, Router B, Router C, and Router D and the Routers communicate through RIPng.

Figure 6-2 Networking diagram of configuring basic RIPng functions



Configuration Notes

When configuring basic RIPng functions, note the following:

- RIPng takes effect only after IPv6 is enabled on interfaces.

- If a RIPng process is bound to a VPN instance, the interfaces that run this RIPng process are also bound to the VPN instance.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer.
2. Enable RIPng and configure basic RIPng functions on each router.

Data Preparation

To complete the configuration, you need the following data:

- IPv6 addresses of interfaces

Procedure

Step 1 Configure IPv6 addresses for interfaces. The configuration details are not described here.

Step 2 Configure basic RIPng functions.

# Configure Router A.

```
[~RouterA] ipv6
[~RouterA] ripng 1
[~RouterA-ripng-1] commit
[~RouterA-ripng-1] quit
[~RouterA] interface pos1/0/0
[~RouterA-Pos1/0/0] ipv6 enable
[~RouterA-Pos1/0/0] ipv6 address 1000::1/24
[~RouterA-Pos1/0/0] ripng 1 enable
[~RouterA-ripng-1] commit
[~RouterA-ripng-1] quit
```

# Configure Router B.

```
[~RouterA] ipv6
[~RouterB] ripng 1
[~RouterA-ripng-1] commit
[~RouterA-ripng-1] quit
[~RouterA] interface pos1/0/0
[~RouterA-Pos1/0/0] ipv6 enable
[~RouterA-Pos1/0/0] ipv6 address 1000::2/24
[~RouterA-Pos1/0/0] ripng 1 enable
[~RouterA-Pos1/0/0] commit
[~RouterA-Pos1/0/0] quit
[~RouterA] interface pos2/0/0
[~RouterA-Pos2/0/0] ipv6 enable
[~RouterA-Pos2/0/0] ipv6 address 2000::1/24
[~RouterA-Pos2/0/0] ripng 1 enable
[~RouterA-Pos2/0/0] commit
[~RouterA-Pos2/0/0] quit
[~RouterA] interface pos3/0/0
[~RouterA-Pos3/0/0] ipv6 enable
[~RouterA-Pos3/0/0] ipv6 address 3000::1/24
[~RouterA-Pos3/0/0] ripng 1 enable
[~RouterA-Pos2/0/0] commit
[~RouterA-Pos3/0/0] quit
```

# Configure Router C.

```
[~RouterA] ipv6
```

```
[~RouterC] ripng 1
[~RouterA-ripng-1] commit
[~RouterA-ripng-1] quit
[~RouterA] interface pos2/0/0
[~RouterA-Pos2/0/0] ipv6 enable
[~RouterA-Pos2/0/0] ipv6 address 2000::2/24
[~RouterA-Pos2/0/0] ripng 1 enable
[~RouterA-Pos2/0/0] commit
[~RouterA-Pos2/0/0] quit
```

# Configure Router D.

```
[~RouterA] ipv6
[~RouterD] ripng 1
[~RouterA-ripng-1] commit
[~RouterA-ripng-1] quit
[~RouterA] interface pos3/0/0
[~RouterA-Pos3/0/0] ipv6 enable
[~RouterA-Pos3/0/0] ipv6 address 3000::2/24
[~RouterA-Pos3/0/0] ripng 1 enable
[~RouterA-Pos3/0/0] commit
[~RouterA-Pos3/0/0] quit
```

Step 3 Verify the configuration.

# View RIPng neighbors of Router A.

```
[~RouterA] display ripng 1 neighbor
Neighbor : FE80::A0A:201:1 Pos1/0/0
Protocol : RIPNG
```

The command output shows that Router A has established the neighbor relationship with other devices on the network.

# View RIPng routing information of Router B.

```
[~RouterB] display ripng 1 route
Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
-----------------------------------------------------------
Peer FE80::F54C:0:9FDB:1  on Pos1/0/0
Dest 1000::1/24,
    via FE80::F54C:0:9FDB:1, cost  1, tag 0, A, 3 Sec
Peer FE80::D472:0:3C23:1  on Pos2/0/0
Dest 2000::2/24,
    via FE80::D472:0:3C23:1, cost  1, tag 0, A, 4 Sec
Peer FE80::D472:0:3C23:1  on Pos3/0/0
Dest 3000::2/24,
    via FE80::D472:0:3C23:1, cost  1, tag 0, A, 4 Sec
```

The command output shows that Router B has learned routing information on the network.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 1000::1/24
 ripng 1 enable
#
```

```
ripng 1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 1000::2/24
 ripng 1 enable
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 2000::1/24
 ripng 1 enable
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 3000::1/24
 ripng 1 enable
#
ripng 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
ipv6
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 2000::2/24
 ripng 1 enable
#
ripng 1
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
ipv6
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 3000::2/24
 ripng 1 enable
#
ripng 1
#
return
```

Related Tasks

6.3 Configuring Basic RIPng Functions

6.9.2 Example for Configuring Split Horizon

This section describes how to configure split horizon to prevent routing loops.

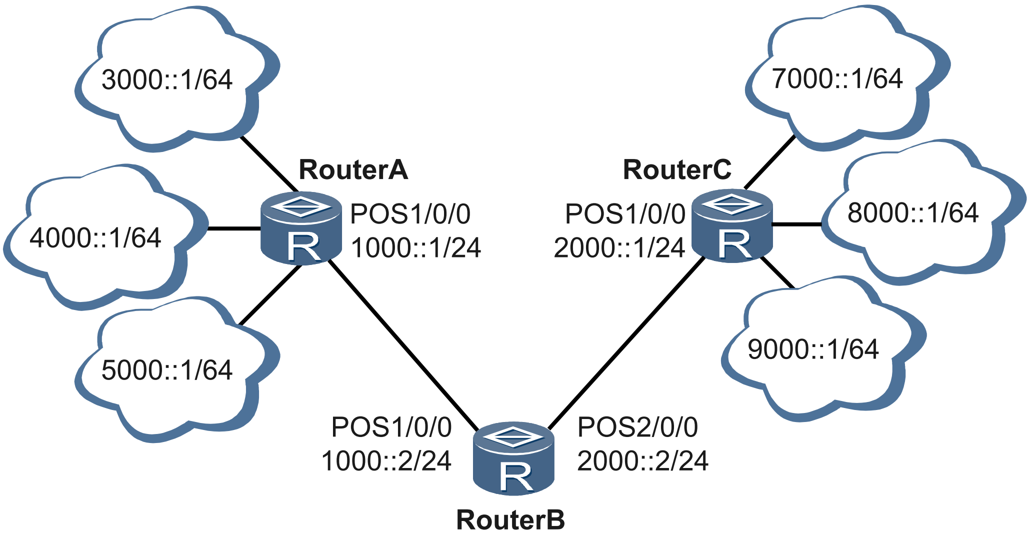
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 6-3**, IP addresses are assigned to all interfaces on each router; RIPng is configured on each router and RIPng services run normally. It is required that split horizon be configured on router A and router C.

Figure 6-3 Networking diagram of preventing routing loops



Configuration Notes

When configuring split horizon to prevent routing loops, note the following:

- Split horizon is enabled by default.
- If both split horizon and poison reverse are configured, only poison reverse takes effect.

Configuration Roadmap

The configuration roadmap is as follows:

1. Enable split horizon.

Data Preparation

None.

Procedure

Step 1 Configure split horizon.

Configure split horizon on the RIPng interfaces of all routers. The configurations of Router B and Router C are the same as the configuration of Router A, and are not described here.

# Configure Router A.

```
[~RouterA] interface pos1/0/0
[~RouterA-Pos1/0/0] ripng split-horizon
[~RouterA-Pos1/0/0] quit
[~RouterA] interface pos1/0/1
[~RouterA-Pos1/0/1] ripng split-horizon
[~RouterA-Pos1/0/1] quit
[~RouterA] interface pos1/0/2
[~RouterA-Pos1/0/2] ripng split-horizon
[~RouterA-Pos1/0/2] quit
[~RouterA] interface pos1/0/3
[~RouterA-Pos1/0/3] ripng split-horizon
[~RouterA-Pos1/0/3] quit
[~RouterA] commit
```

Step 2 Verify the configuration.

# Run the **display ripng 1 interface verbose** command on Router A and Router C to check whether split horizon is enabled. Take the display on Router A as an example. If the **Split-Horizon** field is displayed as **Enabled**, it indicates that split horizon is enabled.

```
[~RouterA] display ripng 1 interface verbose
 Pos1/0/0
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
 Pos1/0/1(10.1.1.1)
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
 Pos1/0/2
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
 Pos1/0/3
    FE80::A0A:200:1
    State : UP, Protocol : RIPNG, MTU : 1440
    Metricin : 0 , Metricout : 1
    Default Route : Disabled
    Poison Reverse : Disabled
    Split Horizon : Enabled
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1000::1/24
 ripng 1 enable
#
interface Pos1/0/1
 undo shutdown
 ipv6 enable
 ipv6 address 3000::1/64
 ripng 1 enable
#
interface Pos1/0/2
 undo shutdown
 ipv6 enable
 ipv6 address 4000::1/64
 ripng 1 enable
#
interface Pos1/0/3
 undo shutdown
 ipv6 enable
 ipv6 address 5000::1/64
 ripng 1 enable
#
ripng 1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1000::2/24
 ripng 1 enable
#
interface Pos2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2000::2/24
 ripng 1 enable
#
ripng 1
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2000::1/24
 ripng 1 enable
```

```
#
interface Pos1/0/1
 undo shutdown
 ipv6 enable
 ipv6 address 7000::1/64
 ripng 1 enable
#
interface Pos1/0/2
 undo shutdown
 ipv6 enable
 ipv6 address 8000::1/64
 ripng 1 enable
#
interface Pos1/0/3
 undo shutdown
 ipv6 enable
 ipv6 address 9000::1/64
 ripng 1 enable
#
ripng 1
#
return
```

Related Tasks

6.4 Preventing Routing Loops

7 IS-IS Configuration

About This Chapter

This chapter describes the principle, configuration procedures, and configuration examples of IS-IS.

BFD can provide channel fault detection featuring light load and high speed (at the millisecond level). With dynamic BFD, routing protocols can dynamically trigger the establishment of BFD sessions.

BFD can provide link failure detection featuring light load and high speed (at the millisecond level). With dynamic BFD, routing protocols can dynamically trigger the establishment of BFD sessions.

By configuring SPF parameters and LSP fast flooding, you can speed up the convergence of IS-IS networks.

You can perform the following configurations to improve the stability of an IS-IS network.

This section describes how to enable the IPv6 capability for IS-IS and adjust IS-IS IPv6 route selection.

This section describes how to configure IS-IS Auto FRR.

On a network that requires high security, you can configure IS-IS authentication to improve the security of the IS-IS network.

Maintaining IS-IS involves resetting IS-IS, clearing the IS-IS statistics, and debugging IS-IS.

This section describes IS-IS configuration examples. You can understand the configuration procedures through the configuration flowchart. Each configuration example consists of such information as the networking requirements, configuration notes, and configuration roadmap.

7.1 Basic Concepts of IS-IS

As an Interior Gateway Protocol (IGP), Intermediate System to Intermediate System (IS-IS) is used within an AS. IS-IS is a link state protocol. It uses the Shortest Path First (SPF) algorithm to calculate routes.

IS-IS is a dynamic routing protocol initially designed by the International Organization for Standardization (ISO) for its Connectionless Network Protocol (CLNP).

The IETF extends and modifies IS-IS in RFC 1195 to support IP routing. This enables IS-IS to be applied to TCP/IP and OSI environments. This type of IS-IS is called Integrated IS-IS or Dual IS-IS.

IS-IS is an IGP and is used within an AS. IS-IS is a link state protocol and uses the SPF algorithm to calculate routes. IS-IS is similar to OSPF in many aspects.

IS-IS Areas

To support a network with a large number of routes, IS-IS adopts a two-level structure in an IS-IS area. A large IS-IS area can be divided into one area or multiple areas.

- Routes within an area are managed by Level-1 devices.
- Routes between areas are managed by Level-2 devices.

Figure 7-1 shows a network topology that runs IS-IS. The network is similar to an OSPF topology with multiple areas. Area 1 is a backbone area. All devices in the area are Level-2 devices. The other four areas are non-backbone areas. They are connected to the backbone area (Area 1) through Level-1-2 devices.

Figure 7-1 IS-IS topology I

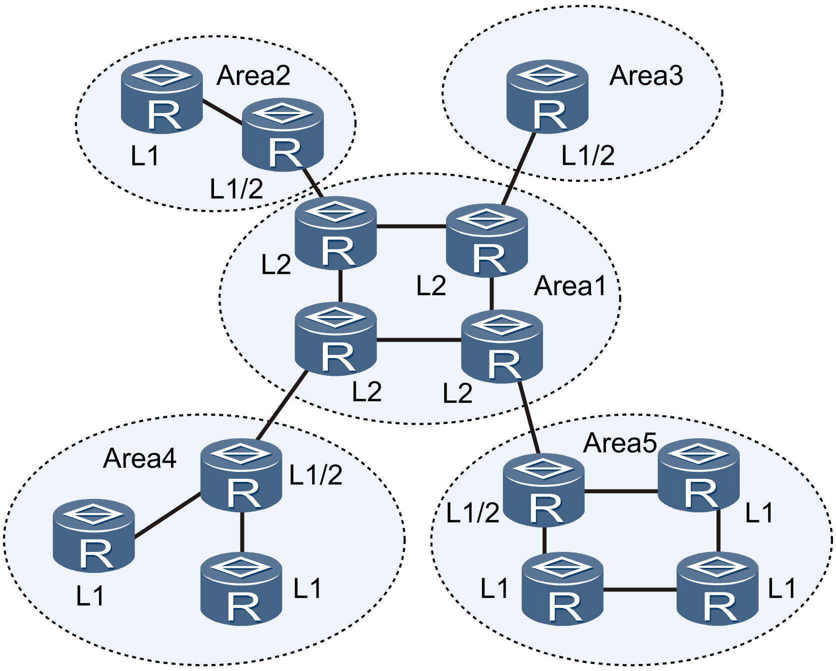
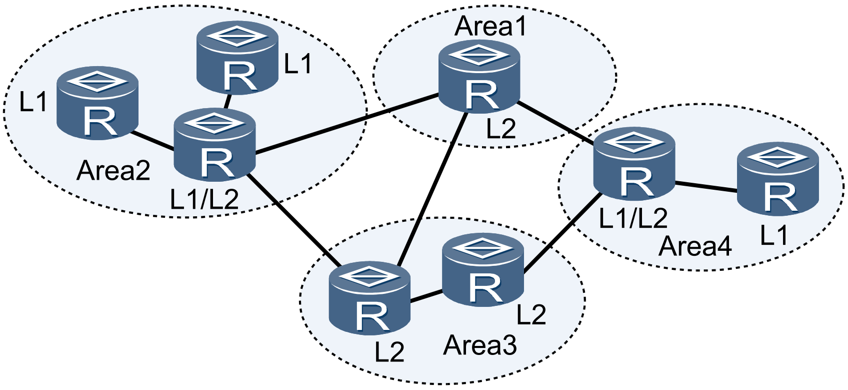


Figure 7-2 shows another network topology that runs IS-IS. The Level-1-2 devices are not only used to connect the Level-1 and Level-2 devices but also connect to Level-2 devices to form an IS-IS backbone network. In this topology, all Level-2 devices are connected to form an IS-IS backbone network. These devices may belong to different areas but must be contiguous.

Figure 7-2 IS-IS topology II



NOTE

The IS-IS backbone network does not refer to a specific area.

This is one of the differences between IS-IS and OSPF. In OSPF, inter-area routes must be forwarded through a backbone area, and only the devices within the same area adopt the SPF algorithm. In IS-IS, both Level-1 and Level-2 devices adopt the SPF algorithm to generate Shortest Path Trees (SPTs) for Level-1 areas and Level-2 areas respectively.

Network Types

IS-IS supports only two types of network. According to physical links, IS-IS networks can be classified into the following types:

- Broadcast network: such as Ethernet and Token-Ring
- Point-to-point (P2P) network: such as PPP and HDLC

7.2 IS-IS Features Supported by the NE5000E

The IS-IS features supported by the NE5000E include hot standby (HSB), TE,administrative tags, LSP fragment extension, dynamic hostname exchange, domain-wide prefix distribution, IS-IS authentication, fast convergence, BFD, and 3-way handshake.

IS-IS HSB

The device of a distributed architecture supports IS-IS HSB. In the IS-IS HSB process, IS-IS backs up data from the Active Main Board (AMB) to the Standby Main Board (SMB). Whenever the AMB fails, the SMB becomes active. In this manner, IS-IS is not affected and functions normally.

In the IS-IS HSB process, IS-IS configurations on the AMB and the SMB are consistent. When the switchover of the AMB and SMB is performed, Non-Stop Routing (NSR) supported by IS-IS can ensure uninterrupted traffic forwarding.

IS-IS GR

Graceful restart (GR) is a function used to restart a device gracefully. It ensures uninterrupted traffic forwarding during the restart of a device in a short time.

If IS-IS restarts in a non-graceful mode, IS-IS sessions are reset and Link State Protocol Data Units (LSPs) are regenerated and flooded. This triggers the SPF calculation in the entire area, which causes route flapping and forwarding interruption in the area. The IETF defines IS-IS GR in RFC 3847, in which the specifications of protocol restart with FIB tables reserved and unreserved are stated.

IS-IS GR involves two roles, namely, GR restarter and GR helper.

- GR restarter

  The GR restarter refers to the device that restarts in GR mode.

- GR helper

  The GR helper refers to another GR device that helps the restarter to complete the GR process. The GR restarter must have the capability of the GR helper.

The NE5000E can function as only the GR helper.

IS-IS TE

IS-IS Traffic Engineering (TE) supports the establishment and maintenance of the TE Label Switched Paths (LSPs).

When establishing Constraint-based Routed LSPs (CR-LSPs), MPLS needs to learn the traffic attributes on all the links in the local area. CR-LSPs can acquire the TE information of the links using IS-IS.

NOTE

For detailed configuration of IS-IS TE, refer to the *Configuration Guide - MPLS*.

All the other LSPs referred to in this chapter are link state protocol data units. You must differentiate the two acronyms that are the same.

Administrative Tag

The use of administrative tags simplifies management. Administrative tags can be used to control the IP prefix advertisement in an IS-IS routing domain. The administrative tag carries the administrative information about an IP address prefix. It is used to control the routes of different levels and routes imported from different areas, various routing protocols, multiple IS-IS instances running on a device, and carrying of tags.

Each administrative tag is associated with certain attributes. If the prefix of the reachable IP address to be advertised by IS-IS has this attribute, IS-IS adds the administrative tag to the reachability TLV in the prefix. In this manner, the tag is advertised throughout the entire IS-IS area.

LSP Fragment Extension

When more information is carried in an LSP to be advertised by IS-IS, IS-IS advertises multiple LSP fragments. Each LSP fragment is identified by the LSP identifier field of an LSP. The LSP identifier field is 1 byte long. Thus, the maximum number of fragments that can be generated by an IS-IS device is 256.

The IS-IS fragment extension feature allows an IS-IS device to generate more LSP fragments. To implement this feature, you can enable network manager and configure additional system IDs for each device. Each system ID represents a virtual system, which can generate 256 LSP fragments. With more additional system IDs (up to 99 virtual systems), an IS-IS device can generate a maximum of 25600 LSP fragments excluding the LSP fragments on the pseudonode.

Dynamic Hostname Exchange Mechanism

The dynamic hostname exchange mechanism is introduced to conveniently manage and maintain IS-IS networks. The mechanism provides a service of mapping hostnames to system IDs for the IS-IS devices. The dynamic hostname information is advertised in the form of a dynamic hostname TLV in an LSP.

The dynamic hostname exchange mechanism also provides a service to associate a host name with the Designated IS (DIS) on a broadcast network. Then, LSPs of pseudo nodes advertise this association in the form of a dynamic hostname TLV.

It is easier to identify and memorize the hostname than the system ID. After this function is configured, the **display** commands used on a device display the host name of the device instead of the system ID.

Domain-wide Prefix Distribution

Domain-wide prefix distribution indicates that an Level-1-2 IS-IS device can advertise the known routing information of one level to the other level. That is, the routing information can be advertised from Level 2 to Level 1 or from Level 1 to Level 2 (default).

IS-IS Authentication

IS-IS authentication encrypts IS-IS packets by adding the authentication field to packets to ensure network security. When a local device receives IS-IS packets from a remote device, the local device discards the packets if finding that the authentication passwords do not match. This protects the local device.

📖 **NOTE**

For details about IS-IS authentication, refer to the chapter "Improving IS-IS Network Security" in the *Configuration Guide – IP Routing*.

IS-IS Fast Convergence

- Incremental SPF (I-SPF)

  Incremental SPF calculates only changed routes at a time rather than all routes.

  In ISO-10589, it is defined that the Dijkstra algorithm is used to calculate routes. When a node is added to or removed from a network topology, using the Dijkstra algorithm calculates the routes of all nodes. As a result, it takes a long time and occupies too many resources, thus affecting the convergence speed of the entire network.

  I-SPF improves this algorithm. Except for the first time, only changed nodes instead of all nodes are involved in calculation. The SPT generated at last is the same as that generated by the Dijkstra algorithm. This decreases the CPU usage and speeds up network convergence.

- Partial Route Calculation (PRC)

  Similar to I-SPF, only changed nodes are involved in PRC. PRC, however, does not calculate the shortest path but updates leaf routes based on the SPT calculated by I-SPF.

In route calculation, a leaf represents a route, and a node represents a device. If the SPT calculated through I-SPF changes, PRC only calculates all the leaves on the changed node; if the SPT calculated through I-SPF does not change, PRC only calculates the changed leaf.

For example, if a node is enabled with an IS-IS interface, the SPT of the entire network remains unchanged. In this case, PRC only updates the routes on the interface of this node, thus reducing CPU usage.

PRC working with I-SPF further improves the convergence performance of the network. As an improvement of the original SPF algorithm, RPC and I-SPF replace the original algorithm.

NOTE

On the NE5000E, only I-SPF and PRC are used to calculate routes.

- LSP fast flooding

  According to the RFC, when IS-IS receives LSPs from other devices and the LSPs are more updated than those in its own LSDB, IS-IS uses a timer to flood out the LSPs in the LSDB at specified intervals. Therefore, the LSDB synchronization is slow.

  LSP fast flooding addresses the problem. When the device configured with this feature receives one or more new LSPs, it floods out the LSPs less than the specified number before route calculation. Thus, LSDB can be synchronized quickly. LSP fast flooding speeds up network convergence to the greatest extent.

- Intelligent timer

  Although the route calculation algorithm is improved, the long interval for triggering the route calculation also affects the convergence speed. Using a millisecond timer can shorten the interval, however, CPU resources will be consumed too much if the network topology changes frequently. An SPF intelligent timer can respond quickly to certain emergent events and also avoid excessive CPU usage.

  An IS-IS network running normally is stable. The network seldom changes frequently, and an IS-IS device does not calculate routes frequently. Thus, you can set a short interval (in milliseconds) for triggering the route calculation for the first time. If the network topology changes frequently, the value of the intelligent timer increases with the calculation times, and thus the interval for route calculation becomes longer. This avoids consuming too much CPU resources.

  The LSP generation intelligent timer is similar to the SPF intelligent timer. In IS-IS, when the LSP generation timer expires, the system regenerates its own LSP according to the current topology. In the original implementation mechanism, a timer with a fixed interval is used, which, however, cannot meet the requirements on fast convergence and low CPU usage. Therefore, the LSP generation timer is designed as an intelligent timer so that it can respond quickly to some emergent events (such as interface going Up or Down) to speed up network convergence. For example, when the network topology changes frequently, the interval of the intelligent timer automatically prolongs, thus preventing too much use of CPU resources.

  NOTE

  You need to configure an intelligent timer according to the actual conditions of the network and device performance.

BFD for IS-IS

The NE5000E supports Bidirectional Forwarding Detection (BFD), which is used to detect IS-IS neighbor relationships. BFD can fast detect the faults on links between IS-IS neighbors and report them to IS-IS. The fast convergence of IS-IS is thus implemented.

NOTE

In IS-IS, only one-hop neighbors can be set up. Thus, BFD detects only the one-hop link between IS-IS neighbors.

- Static BFD

  To configure static BFD, command lines are used to manually configure the parameters of BFD sessions, including the local identifier and remote identifier, and then requests for setting up BFD sessions are delivered manually.

  The disadvantage of static BFD is that you need to manually set up or delete BFD sessions, which lacks flexibility and may cause configuration errors. Moreover, if the local or remote identifier is incorrectly configured, BFD sessions cannot work normally.

- Dynamic BFD

  Dynamic BFD refers to the dynamic establishment of BFD sessions through routing protocols.

  When setting up the new neighbor relationship, the routing protocol sends parameters of neighbors and detection parameters (including source and destination IP addresses) to BFD. BFD then sets up sessions according to the received parameters. Dynamic BFD is more flexible than static BFD.

  Normally, the interval for sending Hello packets is 10s and the IS-IS neighbor holdtime (the timeout period of the neighbor relationship) is three times the interval for sending Hello packets. Therefore, the neighbor fault detection time is in seconds, which may lead to the loss of a large number of packets in a high-speed network.

  Dynamic BFD provides channel fault detection of light load and high speed (in milliseconds). Dynamic BFD does not take the place of the Hello mechanism of IS-IS, but helps IS-IS to detect the faults on neighboring devices or links more quickly, and instructs IS-IS to recalculate routes to correctly guide packet forwarding.

  NOTE

  For details on BFD for IS-IS, refer to the chapter "IS-IS" in the *Feature Description - IP Routing*.

IS-IS 3-Way Handshake

A reliable link layer protocol is required when IS-IS runs on a P2P link. According to ISO-10589, the 2-way handshake mechanism of IS-IS uses Hello packets to set up P2P neighbor relationships. In the 2-way handshake mechanism, once a device receives a Hello packet from its neighbor, it regards the status of the neighbor as Up and sets up an adjacency with the neighbor.

This mechanism has obvious defects. For example, the unstable link status causes the loss of Complete Sequence Number Packets (CSNPs) once an adjacency is set up. As a result, the LSDB fails to be synchronized during an update period of LSPs. If two or more links exist between two neighbors, an adjacency can still be set up when one link is Down in one direction and the other is Up in the same direction. The parameters of the other link in the Up state, however, are also used in SPF calculation. The device does not detect the Down state of the link and still tries to forward packets over this link.

The 3-way handshake mechanism addresses the problem on the unreliable P2P link. In this mechanism, a device regards the neighbor as Up only after confirming that the neighbor receives the packet that it sends, and then sets up an adjacency with the neighbor. In addition, a 32-bit circuit ID is used in the 3-way handshake mechanism, which is an extension of the local 8-bit circuit ID that defines 255 P2P links.

7.3 Configuring Basic IS-IS Functions

An IS-IS network can be set up only after basic IS-IS functions are configured.

Applicable Environment

When configuring IS-IS, you need to start an IS-IS process, specify a NET, and enable IS-IS on interfaces, and then other functions can be configured or take effect.

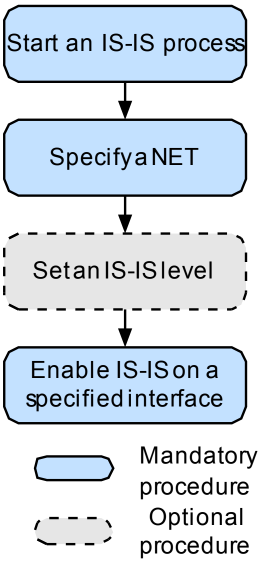
Pre-configuration Tasks

Before configuring basic IS-IS functions, complete the following tasks:

- Configuring a link layer protocol on each interface
- Assigning an IP address to each interface so that neighboring devices are reachable at the network layer

Configuration Procedures

Figure 7-3 Flowchart of configuring basic IS-IS functions



7.3.1 Starting an IS-IS Process

Starting an IS-IS process is the first step before configuring IS-IS functions.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The specified IS-IS process is started and the IS-IS view is displayed.

process-id specifies an IS-IS process. If *process-id* is not specified, the IS-IS process ID of the system is 1 by default.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

7.3.2 Specifying a NET

A Network Entity Title (NET) specifies the IS-IS area address and the system ID of a device. An IS-IS process can be configured with up to three NETs, of which the area addresses can be different but the system IDs must be the same.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The specified IS-IS process is started and the IS-IS view is displayed.

Step 3 Run:

```
network-entity net
```

A NET is specified.

CAUTION

Converting the address of a loopback network to a NET is recommended to ensure that the NET is unique on the network. If a NET is not unique, route flapping may occur. Therefore, plan the network properly.

During the establishment of the Level-2 neighbor relationship, IS-IS does not check whether area addresses are the same. During the establishment of the Level-1 neighbor relationship, area addresses must be the same; otherwise, the Level-1 neighbor relationship cannot be established.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.3.3 (Optional) Setting an IS-IS Level

You can set an IS-IS level for each IS-IS neighbor.

Procedure

- Set a level for an IS-IS process.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     isis [ process-id ]
     ```
     The specified IS-IS process is started and the IS-IS view is displayed.
  3. Run:
     ```
     is-level { level-1 | level-1-2 | level-2 }
     ```
     A level is set for an IS-IS process.

     By default, the level of an IS-IS process is Level-1-2.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.
- Set an IS-IS level for an interface.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     The interface view is displayed.
  3. Run:
     ```
     isis circuit-level [ level-1 | level-1-2 | level-2 ]
     ```
     An IS-IS level is set for an interface.

     NOTE

     The IS-IS level of an interface can be set and take effect only when the level of an IS-IS process is Level-1-2; otherwise, the IS-IS level of an interface is determined by the level of an IS-IS process.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.

----End

7.3.4 Enabling IS-IS on a Specified Interface

By enabling IS-IS on a specified interface, you can bind the interface to an IS-IS process.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis enable [ process-id ]
```

An IS-IS process is started on a specified interface.

By default, IS-IS process 1 is started.

NOTE

Before starting an IS-IS process, you need to make sure that the interface is Up.

If the interface is Down, you need to enable the IS-IS process in the interface view and then run the **undo shutdown** command to enable the interface to go Up.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.3.5 Checking the Configuration

After basic IS-IS functions are successfully configured, you can view information about the interface enabled with IS-IS, LSDB information, neighbor information, routing information, and interface statistical information.

Prerequisite

The configurations of basic IS-IS functions are complete.

Procedure

- Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view information about the interfaces enabled with IS-IS.
- Run the **display isis lsdb** [**verbose**] [**level-1** | **level-2**] [**local** | *lsp-id*] [*process-id*] command to view the LSDB information of IS-IS.
- Run the **display isis peer** [*process-id*] [**verbose**] command to view information about IS-IS neighbors.
- Run the **display isis route** [**level-1** | **level-2**] [*process-id*] [[**ipv4**] [*ip-address* [*mask* | *mask-length*]] | [**ipv6**] [*ipv6-address* [*prefix-length*]]] [**verbose**] command to view the IS-IS routing information.
- Run the **display isis statistics** [**level-1** | **level-2** | **level-1-2**] [*process-id*] command to view statistics on an IS-IS process.

----End

Example

Run the **display isis interface** command, and you can view information about all interfaces enabled with IS-IS.

```
<HUAWEI> display isis interface
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State      MTU  Type  DIS
 GE1/0/0        001         Up                  Down        1497 L1/L2 No/No
```

7.4 Establishing or Maintaining IS-IS Neighbor Relationships or Adjacencies

You can set up or maintain IS-IS neighbor relationships or IS-IS adjacencies by configuring timers of IS-IS packets and setting LSP parameters.

Applicable Environment

You can perform configurations as required when setting up or maintaining IS-IS neighbor relationships or IS-IS adjacencies.

- IS-IS packet timers and LSP parameters can be set to adjust the holdtime of IS-IS neighbor relationships and adjust the interval for sending IS-IS packets.
- The problem that the IP addresses used to set up IS-IS neighbor relationships are not in the same network segment is addressed.
- The Mesh-Group feature can be adopted to restrict the IS-IS neighbor range.

Pre-configuration Tasks

Before setting up or maintaining IS-IS neighbor relationships or IS-IS adjacencies, complete the following tasks:

- Configuring a link layer protocol on each interface
- Assigning an IP address to each interface so that neighboring devices are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

7.4.1 Configuring Timers of IS-IS Packets

You can configure timers of IS-IS packets, including the Hello timer, CSNP timer, and LSP timer.

Procedure

- Set the interval for sending Hello packets.
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     interface *interface-type interface-number*

The interface view is displayed.

3. Run:

   ```
   isis timer hello hello-interval [ level-1 | level-2 ]
   ```

   The interval for sending Hello packets on an interface is set. By default, the interval for sending Hello packets is 10 seconds.

   - The Hello packets over a P2P link have no IS-IS level. Therefore, the parameters **level-1** and **level-2** are valid only on broadcast interfaces.
   - The Hello packets sent by a broadcast interface have IS-IS levels, and you can set an interval for sending Hello packets with a specified IS-IS level. If Hello packets are not specified with IS-IS levels, the same interval is set for sending Level-1 Hello packets and Level-2 Hello packets by default.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Set the number of Hello packets that are used for declaring a neighbor Down.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

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

   The interface view is displayed.

3. Run:

   ```
   isis timer holding-multiplier number [ level-1 | level-2 ]
   ```

   The number of Hello packets for IS-IS to determine whether a neighbor goes Down is set. The value indicates the number of consecutive Hello packets that IS-IS fails to receive from its neighbor. The default value is 3.

   IS-IS maintains neighbor relationships by sending and receiving Hello packets. IS-IS declares a neighbor Down when it fails to receive specified number of Hello packets consecutively in a time period.

   You can adjust the holdtime of the IS-IS neighbor relationship by setting the product of two values, that is, the interval for sending Hello packets multiplied by the number of Hello packets for declaring a neighbor Down.

   - The Hello packets over a P2P link have no IS-IS level. Therefore, the parameters **level-1** and **level-2** are valid only on broadcast interfaces.
   - The Hello packets sent by a broadcast interface have IS-IS levels, and you can set the number of Hello packets declaring a neighbor Down with a specified IS-IS level. If Hello packets are not specified with IS-IS levels, the number of Level-1 Hello packets declaring a neighbor Down is the same as that of Level-2 Hello packets declaring a neighbor Down.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Set the interval for sending CSNPs.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

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

   The interface view is displayed.

3. Run:

   ```
   isis timer csnp csnp-interval [ level-1 | level-2  ]
   ```

   The interval for sending CSNPs on an interface is set. By default, the interval for sending CSNPs is 10 seconds.

   CSNPs are sent by a Designated IS (DIS) on a broadcast network to synchronize LSDBs.

   If no IS-IS level is specified, the interval for sending CSNPs by the IS-IS process with the current IS-IS level is set by default.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Set the interval for retransmitting LSPs.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

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

   The interface view is displayed.

3. Run:

   ```
   isis timer lsp-retransmit retransmit-interval
   ```

   The interval for retransmitting LSPs over a P2P link is set. By default, the interval for retransmitting LSPs over a P2P link is 5 seconds.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

On a P2P link, if a device does not receive a response to a sent LSP from the peer within a certain period, it considers that the sent LSP is lost or discarded. To ensure reliable transmission, the local device retransmits the LSP at a specified interval. By default, the interval for retransmitting LSPs over a P2P link is 5s.

LSPs transmitted on broadcast links do not require any response. Therefore, the interval for retransmitting LSPs can be configured on the interfaces of broadcast networks but does not take effect.

- Set the minimum interval for sending LSPs on an interface.

1. Run:

   ```
   system-view
   ```

The system view is displayed.

2. Run:

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

   The interface view is displayed.

3. Run:

   ```
   isis timer lsp-throttle throttle-interval [ count count  ]
   ```

   The minimum interval for sending LSPs is set. By default, the minimum interval for sending LSPs is 50 milliseconds and a maximum of 10 LSPs can be sent each time.

   count: specifies the maximum number of LSPs that can be sent within the interval specified by *throttle-interval*. The value is an integer that ranges from 1 to 1000.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

The minimum interval for sending LSPs on an IS-IS interface is the delay between two consecutive LSPs and also the interval for sending fragments of a CSNP.

----End

7.4.2 Setting LSP Parameters

You can configure an LSP generation timer to adjust the time to generate LSPs in an IS-IS network, and you can also configure the receiving of LSPs by setting the sizes of the LSPs to be generated and received on both ends.

Procedure

- Set the LSP refresh interval.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     isis [ process-id ]
     ```

     The IS-IS view is displayed.
  3. Run:

     ```
     timer lsp-refresh refresh-interval
     ```

     The LSP refresh interval is set.
  4. Run:

     ```
     commit
     ```

     The configuration is committed.

  To synchronize all LSPs in the entire area, IS-IS regularly refresh LSPs.
- Set the maximum lifetime for an LSP.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

2. Run:

   ```
   isis [ process-id ]
   ```

   The IS-IS view is displayed.

3. Run:

   ```
   timer lsp-max-age age-time
   ```

   The maximum lifetime for an LSP is set.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

When a device generates the system LSP, it fills in the maximum lifetime for this LSP. After this LSP is received by other devices, the lifetime of the LSP is reduced gradually. If the device does not receive any more update LSPs and the lifetime of the LSP is reduced to 0, the LSP will be deleted from the LSDB 60s later if no more updated LSPs are received.

CAUTION

By default, the LSP refresh interval is 900s, and the maximum lifetime of an LSP is 1200s. Make sure that the LSP refresh interval is more than 300s shorter than the maximum LSP lifetime. In this manner, new LSPs can reach all devices in an area before existing LSPs expire.

- Set the intelligent timer used for generating LSPs.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   isis [ process-id ]
   ```

   The IS-IS view is displayed.

3. Run:

   ```
   timer lsp-generation max-interval [ initial-interval [ incremental-
   interval ] ] [ level-1 | level-2 ]
   ```

   The intelligent timer used for generating LSPs is set.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

If no IS-IS level is specified, IS-IS levels are Level-1 and Level-2 by default.

The delay of generating an LSP or an LSP fragment for the first time is *initial-interval*; the delay of generating an LSP or an LSP fragment for the second time is *incremental-interval*. From the third time on, the delay for generating an LSP increases twice every time until the delay reaches *max-interval*. After the delay remains at *max-interval* for three times, or the IS-IS process is restarted, the delay decreases to *initial-interval* again.

If *incremental-interval* is not specified, the delay of generating the first LSP or LSP fragment for the first time is *initial-interval*. From the second time on, the delay of generating an LSP is *max-interval*. After the delay remains at *max-interval* for three times, or the IS-IS process is restarted, the delay decreases to *initial-interval* again.

If only *max-interval* is specified, the intelligent timer is reduced to an ordinary timer.

- Set the size of an LSP.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     isis [ process-id ]
     ```
     The IS-IS view is displayed.
  3. Run:
     ```
     lsp-length originate max-size
     ```
     The size of an LSP to be generated is set.
  4. Run:
     ```
     lsp-length receive max-size
     ```
     The size of an LSP to be received is set.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.

  NOTE

  When setting *max-size*, note that *max-size* of an LSP to be generated must be equal to or smaller than *max-size* of an LSP to be received.

  The value of *max-size* set through the **lsp-length** command must meet the following requirements; otherwise, the MTU status on the interface is considered to be Down.
  - The MTU of an Ethernet interface must be greater than or equal to the sum of *max-size* and 3.
  - The MTU of a P2P interface must be greater than or equal to the value of *max-size*.
- Configure LSP fragment extension.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     isis [ process-id ]
     ```
     The IS-IS view is displayed.
  3. Run:
     ```
     lsp-fragments-extend [ mode-1 | mode-2 ] [ level-1 | level-2 | level-1-2 ]
     ```
     LSP fragment extension for an IS-IS process is enabled.
  4. Run:
     ```
     virtual-system virtual-system-id
     ```
     A virtual system is configured.
  5. Run:

```
commit
```

The configuration is committed.

At least one virtual system ID must be configured for a device to generate extended LSP fragments. The virtual system ID must be unique in the entire IS-IS area.

An IS-IS process can be configured with a maximum of 99 virtual system IDs.

If the mode or level is not specified during the configuration of LSP fragment extension, mode-1 and level-1-2 are adopted by default.

----End

7.4.3 Configuring IS-IS Not to Check the IP Addresses of the Received Hello Packets

You can configure IS-IS not to check the IP addresses of the received Hello packets so that an IS-IS neighbor relationship can be set up between two interfaces whose IP addresses are on different network segments.

Context

In general, IS-IS checks the IP address of each received Hello packet. The IS-IS neighbor relationship can be set up only when the IP address of the Hello packet and the address of the interface that receives the Hello packet belong to the same network segment. If the IP addresses of the local interface and peer interface are in different network segments, you can configure IS-IS not to check the IP addresses of the received Hello packets. In this manner, the IS-IS neighbor relationship can be set up between the interfaces on both ends of a link. The routes of the two different network segments exist in the routing table, but the two routes cannot ping each other successfully.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 (Optional) Run:

```
isis circuit-type p2p
```

The network type of an interface is set to P2P.

Step 4 Run:

```
isis peer-ip-ignore
```

The IP addresses of the received Hello packets are not to be checked.

NOTE

- On a broadcast interface, you need to run the **isis circuit-type p2p** command in the interface view before running the **isis peer-ip-ignore** command. The **isis circuit-type p2p** command takes effect only on broadcast interfaces.
- On a P2P interface or an NBMA interface, you can run the **isis peer-ip-ignore** command without having to run the **isis circuit-type p2p** command first.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.4.4 Checking the Configuration

After establishing or maintaining IS-IS neighbor relationships or adjacencies, you can view information about the IS-IS interfaces.

Prerequisite

The configurations of setting up or maintaining IS-IS neighbor relationships or adjacencies are complete.

Procedure

- Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view information about the interfaces enabled with IS-IS.
- Run the **display isis statistics** [**level-1** | **level-2** | **level-1-2**] [*process-id*] command to view statistics on an IS-IS process.

----End

Example

On Gigabit Ethernet 6/0/0, set the interval for sending Hello packets to 15, number of Hello packets for declaring a neighbor Down to 10, interval for sending Level-1 CSNPs to 123, and minimum interval for sending LSPs to 159. Run the **display isis interface verbose** command, and you can view information about interfaces enabled with IS-IS.

```
<HUAWEI> display isis interface verbose
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
 GE6/0/0         1           Up            Mtu:Up/Lnk:Dn/IP:Dn   1497 L1/L2 No/No
  Circuit MT State           : standard
  SNPA Address               : 00e0-095b-4201
  IP Address                 : 123.1.1.1
  Csnp Timer Value           :  L1      123   L2       10
  Hello Timer Value          :  L1    15000   L2    15000 <ms>
  DIS Hello Timer Value      :  L1     5000   L2     5000 <ms>
  Hello Multiplier Value     :  L1       10   L2       10
  LSP-Throttle Timer         :  L12  159
  Cost                       :  L1    10  L2    10
  Ipv6 Cost                  :  L1    10  L2    10
  Priority                   :  L1    64  L2    64
  Retransmit Timer Value     :  L1    5   L2    5
  Bandwidth-Value            :  Low  1000000000  High  0
  Static Bfd                 :  NO
```

```
Dynamic Bfd                 :  NO
Static IPv6 Bfd             :  NO
Dynamic IPv6 Bfd            :  NO
Extended-Circuit-Id Value   :  0000000000
Circuit State               : OSI:Up / IP:Up / Mtu:Up / IpBorrow:Up /
                            : BandWidth:Up / IsEnable:Up / IfNet:Up
Circuit Ipv6 State          : OSI:Up / IP:Down / Mtu:Up / IpBorrow:Up /
                            : BandWidth:Up / IsEnable:Down / IfNet:Down
```

7.5 Configuring IS-IS Attributes in Different Types of Network

You can configure the network type and DIS priority on an IS-IS interface and configure the neighbor negotiation mode on a P2P link.

Applicable Environment

IS-IS attributes vary with the types of network. This section describes how to configure IS-IS attributes in different types of network, including:

- Simulating a P2P interface on an Ethernet interface by changing the link type of the Ethernet interface to P2P
- Controlling DIS election
- Checking the OSI network negotiation status over a PPP link
- Configuring IS-IS not to check IP addresses when the neighbor relationship is set up between the P2P interfaces on two nodes (In this manner, the neighbor relationship can be set up between two P2P interfaces that reside at different network segments.)

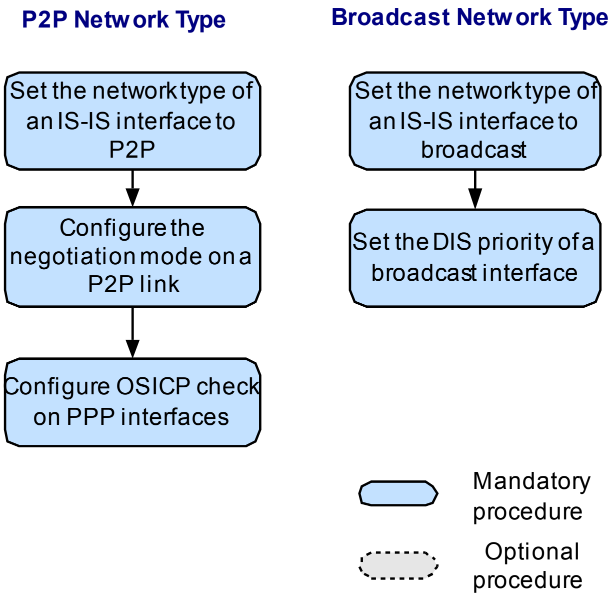
Pre-configuration Tasks

Before configuring IS-IS attributes in different types of network, complete the following tasks:

- Assigning an IP address to each interface so that neighboring devices are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

Figure 7-4 Flowchart of configuring IS-IS attributes in different types of network


P2P Network Type
Broadcast Network Type
Set the network type of an IS-IS interface to P2P
Configure the negotiation mode on a P2P link
Configure OSICP check on PPP interfaces
Set the network type of an IS-IS interface to broadcast
Set the DIS priority of a broadcast interface
Mandatory procedure
Optional procedure

7.5.1 Setting the Network Type of an IS-IS Interface to P2P

You can set the network type of an IS-IS interface to P2P so that the IS-IS neighbor relationship can be set up between two interfaces of different network types.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of a specified interface is displayed.

Step 3 Run:

```
isis circuit-type p2p
```

The network type of an IS-IS interface is set to P2P.

By default, the network type of an interface is determined by the physical type of the interface.

NOTE

The network types of the interfaces on both ends of a link must be the same; otherwise, the IS-IS neighbor relationship cannot be set up between the two interfaces.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.5.2 Setting the DIS Priority of a Broadcast Interface

You can set the DIS priority of a broadcast interface for DIS election.

Context

Level-1 and Level-2 DISs are elected separately and can be set with different priorities. If no level is specified in the command, the same priority is set for both Level-1 and Level-2 DISs.

DIS election is based on DIS priorities. The IS where the interface with the highest DIS priority resides is elected as the DIS. In the case of equal priorities, the interface with the highest MAC address is selected. An interface with the DIS priority of 0 still participates in the DIS election, which is different from OSPF.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of a specified interface is displayed.

Step 3 Run:

```
isis dis-priority priority [ level-1 | level-2 ]
```

The DIS priority used for DIS election is set. The greater the value, the higher the DIS priority.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

If you run the **isis circuit-type** command to simulate a broadcast interface as a P2P interface, the **isis dis-priority** command does not take effect on the simulated interface.

7.5.3 Configuring the Negotiation Mode on a P2P Link

You can use the 3-way handshake mechanism to detect faults on a unidirectional link and detect the unreliable peer status on P2P interfaces of a link with other faults.

Context

This configuration specifies the negotiation mode in which neighbor relationships can be set up on P2P links. If the 3-way handshake mechanism is adopted as the negotiation mode, the faults on a unidirectional link can be detected.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis ppp-negotiation { 2-way | 3-way [ only ] }
```

The negotiation mode used on an interface is specified.

By default, **3-way** is adopted as the negotiation mode.

The **isis ppp-negotiation** { **2-way** | **3-way** [**only**] } command can only be used for the establishment of the neighbor relationships on P2P links. In the case of broadcast links, you can run the **isis circuit-type p2p** command to set the link type to P2P, and then set the negotiation mode for the establishment of the neighbor relationship.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.5.4 Configuring OSICP Negotiation Check on PPP Interfaces

After OSICP negotiation check is configured on PPP interfaces, the OSI network negotiation status of PPP will affect the IS-IS interface status.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of a specified interface is displayed.

Step 3 Run:

```
isis ppp-osicp-check
```

The OSICP negotiation status is checked on a PPP interface.

By default, the OSICP negotiation status of a PPP interface does not affect the status of an IS-IS interface.

After this command is run, the OSICP negotiation status of a PPP interface affects the status of an IS-IS interface. When PPP detects that the OSI network fails, the link status of the IS-IS interface goes Down and the route to the network segment where the interface resides is not advertised through LSPs.

CAUTION

The **isis ppp-osicp-check** command is applicable to only PPP interfaces. For other point-to-point (P2P) interfaces, this command is invalid.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.5.5 Checking the Configuration

After configuring IS-IS attributes in different types of network, you can view information about the interfaces enabled with IS-IS.

Prerequisite

The configurations of IS-IS attributes in different types of network are complete.

Procedure

Step 1 Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view information about the interfaces enabled with IS-IS.

----End

Example

Run the **display isis interface verbose** command, and you can view that Gigabit Ethernet 6/0/0 is simulated as a P2P interface and the priority of Gigabit Ethernet 6/0/0 is set to 100.

```
<HUAWEI> display isis interface verbose
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
 GE6/0/0          2         Up            Mtu:Up/Lnk:Dn/IP:Dn   1497 L1/L2 --
  Circuit MT State            : standard
  Circuit Parameters          : p2p
  SNPA Address                : 00e0-095b-4201
  IP Address                  : 123.1.1.1
  Csnp Timer Value            :  L1    10  L2    10
  Hello Timer Value           :      15000 <ms>
  DIS Hello Timer Value       :
  Hello Multiplier Value      :        3
  LSP-Throttle Timer          :  L12   50
  Cost                        :  L1    10  L2    10
  Ipv6 Cost                   :  L1    10  L2    10
  Priority                    :  L1    100 L2    100
  Retransmit Timer Value      :  L1    5   L2    5
  Bandwidth-Value             :  Low  1000000000  High  0
  Static Bfd                  :  NO
  Dynamic Bfd                 :  NO
  Static IPv6 Bfd             :  NO
  Dynamic IPv6 Bfd            :  NO
  Extended-Circuit-Id Value   :  0000000001
  Circuit State               : OSI:Up / IP:Up / Mtu:Up / IpUnnumber:Up
                              : BandWidth:Up / IsEnable:Up / IfNet:Up
```

```
    Circuit Ipv6 State          : OSI:Up / IP:Down / Mtu:Up / IpUnnumber:Up
                                : BandWidth:Up / IsEnable:Down / IfNet:Down
```

7.6 Adjusting IS-IS Route Selection

You can adjust IS-IS route selection in a network with a complicated environment.

Applicable Environment

This section describes how to optimize and adjust the configurations of IS-IS networks. including:

- Configuring the link cost and IS-IS priority on an IS-IS interface
- Adjusting specified route preferences

Pre-configuration Tasks

Before adjusting and optimizing IS-IS route selection, complete the following tasks:

- Assigning an IP address to each interface so that neighboring devices are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

7.6.1 Setting the Link Cost of an IS-IS Interface

You can set the cost on an IS-IS interface to change the cost of an IS-IS route, thus affecting IS-IS route selection.

Context

The link cost of an IS-IS interface can be calculated in the following modes with priorities in descending order:

- Interface cost: calculates the link cost of a specified interface.
- Global cost: calculates the link costs of all interfaces.
- Automatically-calculated cost: automatically calculates the link cost based on the interface bandwidth.

If none of the preceding configurations is performed, the default cost of an IS-IS interface is 10, and the default cost style is narrow.

Procedure

- Set the IS-IS cost style.
  1. Run:
     system-view

     The system view is displayed.

2. Run:

   ```
   isis [ process-id ]
   ```

   The IS-IS view is displayed.

3. Run:

   ```
   cost-style { narrow | wide | wide-compatible | { { narrow-compatible |
   compatible } [ relax-spf-limit ] } }
   ```

   The IS-IS cost style is set.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

The cost range of an interface and the cost range of a route received by the interface vary with the cost style.

- If the cost type is narrow, the cost of an interface ranges from 1 to 63. The maximum cost of a route received by the interface is 1023.
- If the cost style is narrow-compatible or compatible, the cost of an interface ranges from 1 to 63. The cost of a route received by the interface is related to the parameter **relax-spf-limit**.
  - If **relax-spf-limit** is not specified:

    If the cost of the route is not greater than 1023 and the link cost of every interface that the route passes through is equal to or smaller than 63, the cost of the route received by the interface is the actual one.

    If the cost of the route is not greater than 1023 but the link cost of a certain interface that the route passes through is greater than 63, the device can learn only the routes to the network segment where the interface resides and the routes imported by the interface. The cost of the route received by the interface is the actual one. Subsequent routes forwarded by the interface are discarded.

    If the cost of the route is greater than 1023, the device can learn only the route to the interface whose link cost exceeds 1023 for the first time. That is, the link cost of each interface before this interface is not greater than 63. The routes to the network segment where the interface resides and the routes imported by the interface can all be learnt. The cost of the routes is 1023. Subsequent routes forwarded by the interface are discarded.
  - If **relax-spf-limit** is specified:

    There is no limit on link costs of interfaces or route costs. The cost of the route received by an interface is the actual one.
- If the cost style is wide-compatible or wide, the cost of the interface ranges from 1 to 16777215. When the cost is 16777215, the neighbor TLV generated on the link cannot be used for route calculation but for the transmission of TE information. The maximum cost of a received route is 0xFFFFFFFF.

- Set the cost of a specified interface.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

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

The interface view is displayed.

3. Run:

   ```
   isis cost cost [ level-1 | level-2 ]
   ```

   The cost of an IS-IS interface is set.

   NOTE

   To change the cost of a loopback interface, you can run the **isis cost** command only in the interface view.

   You can use the **isis cost** *cost* [**level-1** | **level-2**] command to set the cost of a specified interface.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Set the global IS-IS cost.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     isis [ process-id ]
     ```

     The IS-IS view is displayed.

  3. Run:

     ```
     circuit-cost cost [ level-1 | level-2 ]
     ```

     The global IS-IS cost of the interface is set.

     You can use the **circuit-cost** *cost* [**level-1** | **level-2**] command to set the link costs of all interfaces at a time.

  4. Run:

     ```
     commit
     ```

     The configuration is committed.

- Enable IS-IS to automatically calculate the interface cost.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     isis [ process-id ]
     ```

     The IS-IS view is displayed.

  3. Run:

     ```
     bandwidth-reference value
     ```

     The bandwidth reference value is set.

     By default, the bandwidth reference value is 100.

  4. Run:

     ```
     auto-cost enable
     ```

The cost of the interface is automatically calculated.

5. Run:

```
commit
```

The configuration is committed.

The configuration of the bandwidth reference value in Step 3 takes effect only when the cost style is wide or wide-compatible. In this case, Cost of each interface = (Bandwidth-reference/Interface bandwidth) x 10.

If the cost style is narrow, narrow-compatible, or compatible, the cost of each interface is based on **Table 7-1**.

Table 7-1 Relationship between interface costs and interface bandwidths

Cost	Bandwidth Range
60	Interface bandwidth =< 10 Mbit/s
50	10 Mbit/s < Interface bandwidth =< 100 Mbit/s
40	100 Mbit/s < Interface bandwidth =< 155 Mbit/s
30	155 Mbit/s < Interface bandwidth =< 622 Mbit/s
20	622 Mbit/s < Interface bandwidth =< 2.5 Gbit/s
10	2.5 Gbit/s < Interface bandwidth

----End

7.6.2 Setting the Preference of the IS-IS Protocol

You can set the preference of the IS-IS protocol so that the preference of the IS-IS protocol is different from that of other protocols. Therefore, route selection is affected accordingly.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
preference preference
```

The preference of the IS-IS protocol is set.

The smaller the configured value, the higher the preference.

By default, the preference of the IS-IS protocol is 15.

Step 4 Run:

```
commit
```

The configuration is committed.

A device can run multiple routing protocols at the same time. When the multiple routing protocols discover routes to the same destination, the route discovered by the protocol with the highest preference is selected.

----End

7.6.3 Setting the Preference of Specific IS-IS Routes

You can set the preference of specific IS-IS routes for route selection.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
preference route-policy route-policy-name
```

The preference of specific IS-IS routes is set according to a routing policy.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.6.4 Setting the Maximum Number of Equal-cost IS-IS Routes

You can set the maximum number of equal-cost IS-IS routes for traffic load balancing. Maximum load balance value depends on Paf value.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
maximum load-balancing number
```

The maximum number of equal-cost IS-IS routes is set.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.6.5 Checking the Configuration

After adjusting IS-IS route selection, you can view the link costs of IS-IS interfaces, IS-IS protocol preference, preferences of various IS-IS routes, and information about IS-IS routes.

Prerequisite

The configurations of IS-IS route selection is complete.

Procedure

Step 1 Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view information about the interfaces enabled with IS-IS.

Step 2 Run the **display isis route** [**level-1** | **level-2**] [*process-id*] [[**ipv4**] [*ip-address* [*mask* | *mask-length*]] | [**ipv6**] [*ipv6-address* [*prefix-length*]]] [**verbose**] command to view the preferences of IS-IS routes.

----End

Example

Set the IPv4 cost on Gigabit Ethernet 6/0/0 to 20. The configuration is displayed as follows:

```
<HUAWEI> display isis interface verbose
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
 GE6/0/0         1          Up            Mtu:Up/Lnk:Dn/IP:Dn   1497  L1/L2 No/No
  Circuit MT State            : standard
  SNPA Address                : 00e0-095b-4201
  IP Address                  : 123.1.1.1
  Csnp Timer Value            :  L1      123   L2       10
  Hello Timer Value           :  L1    15000   L2    15000 <ms>
  DIS Hello Timer Value       :  L1     5000   L2     5000 <ms>
  Hello Multiplier Value      :  L1       10   L2       10
  LSP-Throttle Timer          :  L12  159
  Cost                        :  L1    20  L2    20
  Ipv6 Cost                   :  L1    10  L2    10
  Priority                    :  L1    64  L2    64
  Retransmit Timer Value      :  L1    5   L2    5
  Bandwidth-Value             :  Low  1000000000  High  0
  Static Bfd                  :  NO
  Dynamic Bfd                 :  NO
  Static IPv6 Bfd             :  NO
  Dynamic IPv6 Bfd            :  NO
  Extended-Circuit-Id Value   :  0000000000
```

```
Circuit State              : OSI:Up / IP:Up / Mtu:Up / IpBorrow:Up /
                           : BandWidth:Up / IsEnable:Up / IfNet:Up
Circuit Ipv6 State         : OSI:Up / IP:Down / Mtu:Up / IpBorrow:Up /
                           : BandWidth:Up / IsEnable:Down / IfNet:Down
```

7.7 Controlling IS-IS Routing Information

A device can control the receiving and advertising of IS-IS routing information by generating default routes, leaking routes, filtering routes, and importing external routes, thus affecting the IS-IS routing information that is added to the IS-IS routing table and IP routing table.

Applicable Environment

A device can control the receiving and advertising of IS-IS routing information by generating default routes, leaking routes, filtering routes, and importing external routes, thus affecting the IS-IS routing information that is added to the IS-IS routing table and IP routing table.

Pre-configuration Tasks

Before controlling IS-IS routing information, complete the following tasks:

- Configuring a link layer protocol on each interface
- Assigning an IP address to each interface so that neighboring devices are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or several configuration tasks (excluding "Checking the Configuration") as required.

7.7.1 Configuring IS-IS to Generate Default Routes

You can configure IS-IS to generate default routes to control the advertising of IS-IS routing information.

Context

The IS-IS level of a device determines the IS-IS level of the generated default routes. The generated default routes are advertised to only other devices of the same level. Based on the routing policy, you can forcibly configure IS-IS to generate default routes only when there are matched routes in the routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
default-route-advertise [ always | match default | route-policy route-policy-name ]
[ cost cost ] [ tag tag ] [ level-1 | level-1-2 | level-2 ] [ avoid-learning ]
```

IS-IS is configured to generate default routes.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.2 Controlling IS-IS Route Leaking from the Level-2 Area into the Level-1 Area

By controlling IS-IS route leaking (Level-2 to Level-1), you can enable the routing information in the Level-2 area and in other Level-1 areas to be leaked into the Level-1 area.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
import-route isis level-2 into level-1 [ filter-policy { acl-number | acl-name acl-
name | ip-prefix ip-prefix-name | route-policy route-policy-name } ] [ tag tag ]
```

IS-IS route leaking is configured. The routes in the Level-2 area and other Level-1 areas are leaked into the Level-1 area where the local device resides.

The command is run on the Level-1-2 device that is connected to the external area. By default, the routes in the level-2 area are not leaked into the Level-1 area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.3 Controlling IS-IS Route Leaking from the Level-1 Area into the Level-2 Area

By controlling IS-IS route leaking (Level-1 to Level-2), you can control the IS-IS routing information in the Level-1 area not to be leaked into the Level-2 area so that the routing information in the Level-2 area can be controlled.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
import-route isis level-1 into level-2 [ filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name | route-policy route-policy-name } [ tag tag ]
```

A routing policy is configured to prevent the routes in the Level-1 area from being leaked into the Level-2 area.

The **undo import-route isis level-1 into level-2** command can be used to control the routing information in the Level-1 area not to be leaked into the Level-2 area.

This command is run on the Level-1-2 device that is connected to the external area. By default, the routes (except the default routes) in the Level-1 area are leaked into the Level-2 area.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.4 Configuring IS-IS to Import External Routes

By configuring IS-IS to import routes, you can enable IS-IS to learn routing information of other protocols or other IS-IS processes.

Context

IS-IS regards the routes discovered by other routing protocols or other IS-IS processes as external routes. When routes of other protocols are being imported, you can specify their default costs.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Configure IS-IS to import external routes.

Before setting costs for the imported routes, you can run the **import-route** *protocol* [*process-id*] [**cost-type** { **external** | **internal** } | **cost** *cost* | **tag** *tag* | **route-policy** *route-policy-name* | [**level-1** | **level-2** | **level-1-2**]] * command to import external routes.

Before retaining the original costs of the imported routes, you can run the **import-route** { { **rip** | **isis** | **ospf** } [*process-id*] | **bgp** } **inherit-cost** [**tag** *tag* | **route-policy** *route-policy-name* | [**level-1** | **level-2** | **level-1-2**]] * command to import external routes.

If you do not specify a level for the imported routes, the level of the imported routes is Level-2.

NOTE

When the cost of a route to be imported is modified by using the routing policy, the following situations occur:

- If both route-policy route-policy-name and inherit-cost are configured, the cost modified by using the routing policy does not take effect.
- If both route-policy route-policy-name and cost cost are configured, the cost modified by using the routing policy is preferred.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.5 Configuring IS-IS to Filter the Received Routes

By configuring IS-IS to filter the received routes, you can control the number of IS-IS routes to be added to the IP routing table, and thus reduce the size of the IP routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name | route-policy route-policy-name } import
```

IS-IS is configured to filter the received routes that need to be added to the IP routing table.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.6 Configuring IS-IS to Filter the Routes to Be Advertised

By configuring IS-IS to filter the routes to be advertised, you can effectively control the number of IS-IS routes on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix -name | route-policy route-policy-name } export [ protocol [ process-id ] ]
```

IS-IS is configured to filter the imported routes that need to be advertised.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.7.7 Checking the Configuration

After controlling IS-IS routing information is configured, you can view IS-IS routing tables and routing information.

Prerequisite

The configurations of controlling IS-IS routing information are complete.

Procedure

- Run the **display isis route** [**level-1** | **level-2**] [*process-id*] [[**ipv4**] [*ip-address* [*mask* | *mask-length*]] | [**ipv6**] [*ipv6-address* [*prefix-length*]]] [**verbose**] command to view the IS-IS routing information.
- Run the **display isis lsdb** [**verbose**] [**level-1** | **level-2**] [**local** | *lsp-id*] [*process-id*] command to view the IS-IS LSDB.

----End

Example

Enable IS-IS process 1 to leak routes from a Level-2 area to a Level-1 area. Run the **display isis lsdb verbose level-1 local 1** command on the device, and you can view the leaked routes. The command output is as follows:

```
<HUAWEI> display isis lsdb verbose level-1 local 1
                        Database information for ISIS(1)
                        --------------------------------
                          Level-1 Link State Database
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0002.00-00* 0x00000006   0x6ed8        1110          110     0/0/0
 SOURCE       0000.0000.0002.00
 NLPID        IPV4
```

```
AREA ADDR    10
INTF ADDR    200.1.1.1
INTF ADDR    100.1.1.2
NBR  ID      0000.0000.0003.01 COST: 10
IP-Internal  200.1.1.0      255.255.255.0    COST: 10
IP-Internal  100.1.1.0      255.255.255.0    COST: 10
IP-Internal* 5.5.5.5        255.255.255.255  COST: 10
IP-External* 1.1.1.1        255.255.255.255  COST: 74
   *(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
          ATT-Attached, P-Partition, OL-Overload
```

7.8 Configuring Static BFD for IS-IS

BFD can provide channel fault detection featuring light load and high speed (at the millisecond level). Static BFD needs to be configured manually.

Applicable Environment

To accelerate the IS-IS convergence when the status of a link changes, you can configure BFD on IS-IS links. To configure a static BFD session, you need to set parameters for the BFD session, including the local discriminator and remote discriminator through command lines, and then enable BFD.

NOTE

BFD detects only the one-hop link between IS-IS neighbors. This is because IS-IS establishes only one-hop neighbors.

Pre-configuration Tasks

Before configuring static BFD for IS-IS, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

Global BFD is enabled on the local end.

Step 3 Run:

```
quit
```

Return to the system view.

Step 4 Run:

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

The interface view is displayed.

BFD can be enabled on physical interfaces only.

Step 5 Run:

```
isis bfd static
```

Static BFD is enabled on the interface.

Step 6 Run:

```
quit
```

Return to the system view.

Step 7 Run:

```
bfd cfg-name bind peer-ip ip-address [ interface interface-name ]
```

A BFD binding is created.

If the peer IP address and local interface are specified, it indicates that BFD is configured to detect the one-hop link, that is, a specific route with this interface as the outbound interface and with the peer IP address as the next-hop address.

Step 8 Run the following commands as required:

- To set the local discriminator, run the **discriminator local** *discr-value* command.
- To set the remote discriminator, run the **discriminator remote** *discr-value* command.

The local discriminator and remote discriminator of devices at both ends of the BFD session should be correctly associated; otherwise, the session cannot be established. After being configured, the local discriminator and remote discriminator cannot be modified.

NOTE

The local discriminator of the local device corresponds to the remote discriminator of the peer device, and the remote discriminator of the local device corresponds to the local discriminator of the peer device.

Step 9 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following commands to check the previous configuration.

- Run the **display isis bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **session** { **peer** *ip-address* | **all** } command to view the information about the BFD session.
- Run the **display isis interface verbose** command to view the configurations of BFD for IS-IS on an interface.

You can check the information about a BFD session only after parameters of the BFD session are set and the BFD session is established. If the configurations are correct, the value of the Fast-Sense field is displayed as YES.

Using the **display isis interface verbose** command, you can find that the value indicating the status of static BFD for IS-IS process 1 is YES.

```
<HUAWEI> display isis interface verbose
                       Interface information for ISIS(1)
                       ---------------------------------
```

```
Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
GE6/0/0          1          Up          Mtu:Up/Lnk:Dn/IP:Dn    1497 L1/L2 No/No
 Circuit MT State            : standard
 SNPA Address                : 00e0-c72d-da01
 IP Address                  : 123.1.1.1
 Csnp Timer Value            :  L1    10  L2    10
 Hello Timer Value           :  L1    10  L2    10
 DIS Hello Timer Value       :  L1     3  L2     3
 Hello Multiplier Value      :  L1     3  L2     3
 LSP-Throttle Timer          :  L12   50
 Cost                        :  L1    20  L2    20
 Ipv6 Cost                   :  L1    10  L2    10
 Priority                    :  L1    64  L2    64
 Retransmit Timer Value      :  L1    5   L2    5
 Bandwidth-Value             :  Low  1000000000  High  0
 Static Bfd                  :  YES
 Dynamic Bfd                 :  NO
 Static IPv6 Bfd             :  NO
 Dynamic IPv6 Bfd            :  NO
 Extended-Circuit-Id Value   :  0000000000
 Circuit State               : OSI:Up / IP:Up / Mtu:Up / IpUnnumber:Up
                             : BandWidth:Up / IsEnable:Up / IfNet:Up
 Circuit Ipv6 State          : OSI:Up / IP:Down / Mtu:Up / IpUnnumber:Up
                             : BandWidth:Up / IsEnable:Down / IfNet:Down
```

7.9 Configuring Dynamic BFD for IS-IS

BFD can provide channel fault detection featuring light load and high speed (at the millisecond level). With dynamic BFD, routing protocols can dynamically trigger the establishment of BFD sessions.

Applicable Environment

If the requirement for data transmission is high and IS-IS convergence needs to be accelerated when the link status changes, you can configure dynamic BFD on IS-IS links.

Dynamic BFD needs to be configured according to the actual network environment. If the time parameters are set improperly, network flapping may occur.

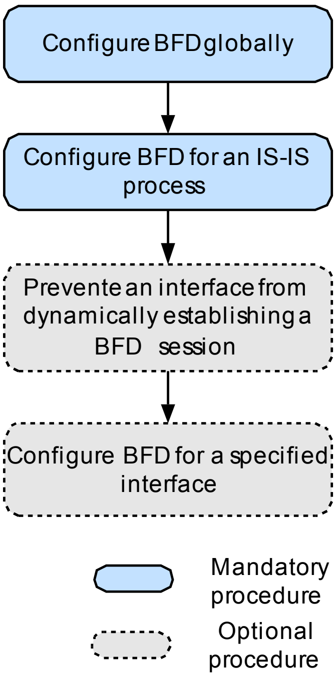
Pre-configuration Tasks

Before configuring dynamic BFD for IS-IS, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

Figure 7-5 Flowchart of configuring basic IS-IS functions


Configure BFD globally
Configure BFD for an IS-IS process
Prevente an interface from dynamically establishing a BFD session
Configure BFD for a specified interface
Mandatory procedure
Optional procedure

7.9.1 Configuring BFD Globally

Before configuring dynamic BFD for IS-IS, you need to enable BFD globally.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

BFD is configured globally.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

7.9.2 Configuring BFD for an IS-IS Process

By configuring BFD for an IS-IS process, you can set parameters for dynamic BFD sessions and enable dynamic BFD for IS-IS on all IS-IS interfaces.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis process-id
```

The IS-IS view is displayed.

Step 3 Run:

```
bfd all-interfaces enable
```

BFD for an IS-IS process is enabled and a BFD session is established.

When global BFD is enabled and the neighbor status is Up, IS-IS adopts default BFD parameters to establish BFD sessions on all the interfaces.

Step 4 (Optional) Run:

```
bfd all-interfaces { min-rx-interval receive interval | min-tx-interval transmit-interval | detect-multiplier multiplier-value | frr-binding } *
```

The parameters for establishing a BFD session are set.

- **min-rx-interval** *receive interval*: specifies the expected minimum interval for receiving BFD packets from the peer.
- **min-tx-interval** *transmit-interval*: specifies the minimum interval for sending BFD packets to the peer.
- **detect-multiplier** *multiplier-value*: specifies the local BFD detection multiplier.
- **frr-binding**: enables the binding of IS-IS auto FRR and an interface's BFD session status.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.9.3 (Optional) Preventing an Interface from Dynamically Establishing a BFD Session

If you do not want certain IS-IS interfaces to establish dynamic BFD sessions, you can disable these interfaces from dynamically establishing BFD sessions.

Context

After the **bfd all-interfaces enable** command is used for an IS-IS process on a P2P network, all IS-IS interfaces whose neighbor relationship is Up establish dynamic BFD sessions; all IS-IS interfaces whose neighbor relationship is Up on a broadcast network establish dynamic sessions between DISs and non-DISs. If you do not want certain IS-IS interfaces to establish dynamic BFD sessions, you can disable these interfaces from dynamically establishing BFD sessions. Do as follows to disable the specified interface from dynamically establishing BFD sessions:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis bfd block
```

The interface is prevented from dynamically establishing a BFD session.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.9.4 (Optional) Configuring BFD for a Specified Interface

You can configure different dynamic BFD session parameters for certain interfaces. The priority of BFD configured on an interface is higher than that of BFD configured for a process.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis bfd enable
```

BFD is enabled on the interface and a BFD session is established.

When global BFD is configured and the neighbor relationship is Up (In the case of broadcast network, DIS is in the Up state), the default BFD parameters are used for dynamically establishing a BFD session.

Step 4 Run:

```
isis bfd { min-rx-interval receive-interval | min-tx-interval transmit-interval |
detect-multiplier multiplier-value | frr-binding } *
```

The parameters for establishing a BFD session are set a specified interface.

NOTE

The priority of BFD configured on an interface is higher than that of BFD configured for a process. That is, if BFD is also enabled on an interface, the parameters on the interface are used to establish a dynamic BFD session.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.9.5 Checking the Configuration

After configuring dynamic BFD for IS-IS, you can check information about the BFD session and dynamic BFD for IS-IS on an interface.

Prerequisite

The configurations of dynamic BFD for IS-IS are complete.

Procedure

- Run the **display isis bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **session** { **peer** *ip-address* | **all** } command to view the information about the BFD session.
- Run the **display isis bfd** [*process-id*] **interface** command to view the information about BFD on an interface.

----End

Example

When BFD is enabled on both ends of the link, run the **display isis bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **session** { **peer** *ip-address* | **all** } command to view the status of the BFD session. The command output shows that the BFD session is Up. For example:

```
<HUAWEI> display isis bfd session all
                      BFD session information for ISIS(1)
                      -----------------------------------
Peer System ID : 2222.2222.2222        Interface : GE1/0/0
BFD State : up              Type : L1
Peer IP Address : 192.168.1.1
Local IP Address: 192.168.1.2
Total BFD session(s): 1
```

Using the **display isis bfd** [*process-id*] **interface** command, you can view all the interfaces enabled with BFD and the values of the BFD session parameters on these interfaces.

```
<HUAWEI> display isis bfd interface
                    BFD information of interface for ISIS(1)
                   -----------------------------------------
 Interface          BFD.State         Min-Tx          Min-Rx         Mul
Frr-Binding
 Pos1/0/0           enable            1000            1000           3
enable
 Total interfaces: 1                          Total bfd enabled interfaces: 1
```

7.10 Configuring Dynamic IPv6 BFD for IS-IS

BFD can provide link failure detection featuring light load and high speed (at the millisecond level). With dynamic BFD, routing protocols can dynamically trigger the establishment of BFD sessions.

Applicable Environment

If the requirement for data transmission is high and IS-IS convergence needs to be accelerated when the link status changes, you can configure dynamic BFD on IS-IS links.

Dynamic BFD needs to be configured based on the actual network environment. If the time parameters are set improperly, network flapping may occur.

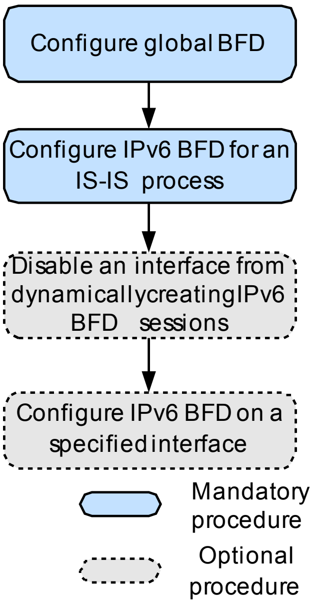
Pre-configuration Tasks

Before configuring dynamic IPv6 BFD for IS-IS, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring IS-IS IPv6 Features**

Configuration Procedures

Figure 7-6 Flowchart for configuring dynamic IPv6 BFD for IS-IS


Configure global BFD
Configure IPv6 BFD for an IS-IS process
Disable an interface from dynamicallycreatingIPv6 BFD sessions
Configure IPv6 BFD on a specified interface
Mandatory procedure
Optional procedure

7.10.1 Configuring BFD Globally

Before configuring dynamic BFD for IS-IS, you need to enable BFD globally.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

BFD is configured globally.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

7.10.2 Configuring IPv6 BFD for IS-IS Processes

By configuring IPv6 BFD for an IS-IS process, you can set parameters for dynamic BFD sessions and enable dynamic IPv6 BFD for IS-IS on all IS-IS interfaces.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis process-id
```

The IS-IS view is displayed.

Step 3 Run:

```
ipv6 bfd all-interfaces enable
```

IPv6 BFD is enabled in the IS-IS process to establish BFD sessions.

When global IPv6 BFD is enabled in the IS-IS process and the neighbor IPv6 status is Up, IS-IS adopts default BFD parameters to establish BFD sessions on all the interfaces.

Step 4 (Optional) Run:

```
ipv6_bfd all-interfaces { min-rx-interval receive interval | min-tx-interval
transmit-interval | detect-multiplier multiplier-value | frr-binding } *
```

IPv6 BFD parameters are configured for setting up BFD sessions.

- **min-rx-interval** *receive interval*: specifies the minimum interval at which BFD packets are received from the peer end.
- **min-tx-interval** *transmit-interval*: specifies the minimum interval at which BFD packets are sent to the peer end.
- **detect-multiplier** *multiplier-value*: specifies the local detection time multiplier, which determines the neighbor holdtime.

- **frr-binding**: indicates that the status of the IPv6 BFD session is bound to IPv6 IS-IS Auto FRR.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.10.3 (Optional) Preventing an Interface from Dynamically Establishing an IPv6 BFD Session

You can disable certain IS-IS interfaces from dynamically establishing IPv6 BFD sessions.

Context

After the **ipv6_bfd all-interfaces enable** command is used for an IS-IS process on a P2P network, all IS-IS interfaces whose neighbors are Up establish dynamic BFD sessions; all IS-IS interfaces whose neighbors are Up on a broadcast network establish BFD sessions between DISs and non-DISs. If you do not expect certain IS-IS interfaces to establish dynamic BFD sessions, you can disable these interfaces from dynamically establishing BFD sessions. Do as follows to disable the specified interface from dynamically establishing BFD sessions:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis ipv6 bfd block
```

The interface is prevented from dynamically establishing a BFD session.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.10.4 (Optional) Configuring IPv6 BFD for a Specified Interface

You can configure IPv6 BFD parameters on a specified interface. The priority of IPv6 BFD parameters on an interface is higher than that of IPv6 BFD parameters in the process.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis ipv6 bfd enable
```

IPv6 BFD is enabled on the interface to establish IPv6 BFD sessions.

After IPv6 BFD is globally enabled in an IS-IS process, default IPv6 BFD parameters are used for establishing IPv6 BFD sessions.

Step 4 Run:

```
isis ipv6 bfd { min-rx-interval receive-interval | min-tx-interval transmit-
interval | detect-multiplier multiplier-value | frr-binding } *
```

IPv6 BFD parameters are configured for setting up BFD sessions.

NOTE

The priority of IPv6 BFD configured on an interface is higher than that of IPv6 BFD configured for a process. That is, if BFD is also enabled on an interface, the parameters on the interface are preferentially used to establish a dynamic BFD session.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.10.5 Checking the Configuration

After configuring dynamic IPv6 BFD for IS-IS, you can check information about the BFD session and dynamic BFD for IS-IS on an interface.

Prerequisite

The configurations of dynamic IPv6 BFD for IS-IS are complete.

Procedure

- Run the **display isis ipv6 bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **session** { **all** | **peer** *ipv6-address* } command to check information about BFD sessions.
- Run the **display isis ipv6 bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **interface** command to check the BFD configurations on an interface.

----End

Example

When IPv6 BFD for IS-IS is enabled for both ends of a link, you can run the **display isis ipv6 bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **session** { **all** | **peer** *ipv6-address* } command to view the BFD state. The command output shows that the BFD state is Up. For example:

```
<HUAWEI> display isis ipv6 bfd 1 session all
                    IPv6 BFD session information for ISIS(1)
                    ----------------------------------------
Peer System ID : 0000.0000.0022        Interface : GE1/0/0
IPv6 BFD State : up  Type : L2
Peer IPv6 Address : FE80::360C:8DFF:FE10:301
Local IPv6 Address: FE80::360B:8DFF:FE10:301

Total BFD session(s): 1
```

Run the **display isis ipv6 bfd** [*process-id* | **vpn-instance** *vpn-instance-name*] **interface** command to view the BFD session parameters on all BFD-capable interfaces.

```
<HUAWEI> display isis ipv6 bfd interface
        IPV6 BFD information of interface for ISIS(1)
         ---------------------------------------------
Interface       BFD6.State       Min-Tx     Min-Rx     Mul     Frr-Binding
Pos1/0/0           enable         1000       1000      10      disable
Total interfaces: 1 Total IPv6 bfd enabled interfaces: 1
```

7.11 Configuring Fast Convergence for IS-IS

By configuring SPF parameters and LSP fast flooding, you can speed up the convergence of IS-IS networks.

Applicable Environment

This section describes how to configure IS-IS fast convergence, including:

- Adjusting SPF parameters to avoid the problem of resource consumption caused by the frequent changes on the network
- Configuring LSP fast flooding to accelerate the network convergence

Pre-configuration Tasks

Before configuring fast convergence for IS-IS, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

7.11.1 Configuring SPF Parameters

By setting SPF parameters, you can adjust the interval for calculating IS-IS routes, and thus avoid network flappings caused by frequent route calculation.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
timer spf max-interval [ init-interval [ incr-interval ] ]
```

The SPF intelligent timer is set.

Step 4 Run:

```
commit
```

The configuration is committed.

The intelligent timer changes as follows:

- The delay for the first SPF calculation is *init-interval*; the delay for the second SPF calculation is *incr-interval*. From the third time on, the delay for generating an LSP doubles every time until the delay reaches *max-interval*. After the delay of SPF calculation remains *max-interval* for three times, or the IS-IS process is restarted, the delay decreases to *init-interval*.
- When *incr-interval* is not specified, the delay of the first SPF calculation is *init-interval*; the delay of the following SPF calculations is *max-interval*. When the delay of SPF calculation is *max-interval* for three times or the IS-IS process is restarted, the delay decreases to *init-interval*.
- When only *max-interval* is specified, the intelligent timer functions as an ordinary one-time triggering timer.

----End

7.11.2 Configuring LSP Fast Flooding

By configuring LSP fast flooding, you can speed up the convergence of IS-IS networks.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
flash-flood [ lsp-count ] [ max-timer-interval interval ] [ level-1 | level-2 ]
```

LSP fast flooding is enabled.

You can use the **flash-flood** command to speed up the LSP flooding. The parameter *lsp-count* specifies the number of LSPs flooded each time, which is applicable to all interfaces. If the number of LSPs to be sent is greater than *lsp-count*, the value *lsp-count* takes effect. If a timer is configured and the configured timer does not time out before the route calculation, the LSPs are flooded immediately when being received; otherwise, the LSPs are sent when the timer times out.

When LSP fast flooding is configured, Level-1 LSPs and Level-2 LSPs are fast flooded by default if neither Level-1 nor Level-2 is specified.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.11.3 Checking the Configuration

After configuring IS-IS fast convergence, you can view the convergence priority of IS-IS routes, parameters of the SPF intelligent timer, and duration for the SPF calculation.

Prerequisite

The configurations of IS-IS fast convergence are complete.

Procedure

- Run the **display isis route verbose** command to view the convergence priority of IS-IS routes.

----End

Example

After the convergence priority of route 10.0.0.0/24 is set to Medium, run the **display isis route verbose** command to view the convergence priority of an IS-IS route. The command output shows that the convergence priority of route 10.0.0.0/24 is displayed as Medium.

```
<HUAWEI> display isis route verbose

                         Route information for ISIS(1)
                         -----------------------------

                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 10.0.0.0/24        Int. Cost : 10            Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 2             Flags     : D/-/L/-
 Priority   : Medium
 NextHop    :                    Interface :               ExitIndex :
    Direct                             GE1/0/0                   0x00000000

 IPV4 Dest  : 20.0.0.0/24        Int. Cost : 10            Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 2             Flags     : D/-/L/-
 Priority   : -
 NextHop    :                    Interface :               ExitIndex :
    Direct                             GE2/0/0                   0x00000000

 IPV4 Dest  : 30.0.0.0/24        Int. Cost : 20            Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 2             Flags     : A/-/L/-
 Priority   : Low
 NextHop    :                    Interface :               ExitIndex :
    20.0.0.2                           GE2/0/0                   0x00000003
    10.0.0.2                           GE1/0/0                   0x00000005
 IPV4 Dest   : 1.1.1.1/32         Int. Cost : 10            Ext. Cost : NULL
 Admin Tag   : -                  Src Count : 1             Flags     : A/-/L/-
 Priority    : Medium
 NextHop     :                    Interface :               ExitIndex :
    10.0.0.2                            GE1/0/0                   0x00000005
 (B)20.0.0.2                            GE2/0/0                   0x00000003

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set


                       ISIS(1) Level-2 Forwarding Table
                       --------------------------------

 IPV4 Dest   : 10.0.0.0/24        Int. Cost : 10            Ext. Cost : NULL
 Admin Tag   : -                  Src Count : 3             Flags     : D/-/L/-
 Priority    : -
 NextHop     :                    Interface :               ExitIndex :
    Direct                              GE1/0/0                   0x00000000

 IPV4 Dest   : 20.0.0.0/24        Int. Cost : 10            Ext. Cost : NULL
 Admin Tag   : -                  Src Count : 3             Flags     : D/-/L/-
 Priority    : -
 NextHop     :                    Interface :               ExitIndex :
    Direct                              GE2/0/0                   0x00000000

 IPV4 Dest   : 30.0.0.0/24        Int. Cost : 20            Ext. Cost : NULL
 Admin Tag   : -                  Src Count : 2             Flags     : -/-/-/-
 Priority    : Low

 IPV4 Dest   : 1.1.1.1/32         Int. Cost : 10            Ext. Cost : NULL
 Admin Tag   : -                  Src Count : 2             Flags     : -/-/-/-
 Priority    : Medium

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set
```

7.12 Improving the Stability of an IS-IS Network

You can perform the following configurations to improve the stability of an IS-IS network.

Applicable Environment

This section describes how to improve the stability of an IS-IS network, including:

- Suppressing an IS-IS interface to reduce the flooding of broadcast packets on the network
- Configuring the overload bit to protect against route calculation errors caused by faults on a single device

Pre-configuration Tasks

Before establishing or maintaining IS-IS neighbor relationships or adjacencies, complete the following tasks:

- Configuring the link layer protocol on the interfaces
- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

7.12.1 Suppressing an Interface from Receiving or Sending IS-IS Packets

By suppressing an IS-IS interface, you can configure the interface to advertise only direct routes and not to receive or send IS-IS routes. In this manner, the transmission of unnecessary packets on IS-IS networks is minimized.

Context

When an IS-IS network is connected to other ASs, it is required to enable IS-IS on the border interfaces to let the devices within the AS learn the routes destined for the border interfaces. This interface, however, unnecessarily advertises IS-IS Hello packets on its network segment. In this case, you can run the **isis silent** command to suppress the IS-IS interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis silent
```

The IS-IS interface is suppressed.

When the IS-IS interface is suppressed, the interface no longer sends or receives any IS-IS packet. The routes of the network segment where the interface resides, however, can still be advertised to other devices within the AS.

NOTE

If the IS-IS protocol on the interface in the AS is Down, the devices within the AS cannot learn the outbound routes.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.12.2 Configuring the LSDB Overload Bit

LSPs with the overload bit are flooded on the network, but the LSPs are not used for calculating routes that pass through a device configured with the overload bit are calculated.

Context

After a device is configured with the overload bit, other devices ignore the device when performing the SPF calculation. The direct routes between the device and other devices, however, are still calculated.

If a device in an IS-IS domain is faulty, it results in incorrect calculation of routes in the entire domain. To troubleshoot this problem, you can set the overload bit for this device to isolate it from the IS-IS network temporarily for convenient troubleshooting.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
set-overload [ on-startup [ timeout1 | start-from-nbr system-id [ timeout1
[ timeout2 ] ] | wait-for-bgp [ timeout1 ] ] ] [ allow { interlevel | external }
* ]
```

The overload bit is set.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.12.3 Checking the Configuration

After improving the stability of an IS-IS network, you can view the IS-IS interface status and the overload bit of the LSDB in the IS-IS process.

Prerequisite

The configurations of improving the stability of an IS-IS network are complete.

Procedure

- Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view the IS-IS interface status.
- Run the **display isis lsdb** [**verbose**] [**level-1** | **level-2**] [**local** | *lsp-id*] [*process-id*] command to view the IS-IS LSDB.

----End

Example

Run the **display isis lsdb 100** command, and you can view that the value of the overload bit in the locally generated LSP is set to 1.

```
<HUAWEI> display isis lsdb 100

                        Database information for ISIS( 100 )
                        --------------------------------
                          Level-1 Link State Database

-----------------------------------------------------------------------------
LSPID                  Seq Num     Checksum   Holdtime       Length   ATT/P/OL
-----------------------------------------------------------------------------
0000.0000.0032.00-00*  0x00000012  0x37b9     1142           55       0/0/1


    *(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload


-----------------------------------------------------------------------------
                           Level-2 Link State Database

-----------------------------------------------------------------------------
LSPID                  Seq Num     Checksum   Holdtime       Length   ATT/P/OL
-----------------------------------------------------------------------------
0000.0000.0032.00-00*  0x00000011  0x39b8     1137           55       0/0/1


    *(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
```

7.13 Configuring IS-IS IPv6 Features

This section describes how to enable the IPv6 capability for IS-IS and adjust IS-IS IPv6 route selection.

Applicable Environment

IS-IS supports multiple network layer protocols, including IPv6. On an IPv6 network, you can implement network interconnection by configuring IS-IS.

Most IS-IS IPv6 features are similar to IS-IS IPv4 features in terms of functions and configurations. This section briefly lists the configuration procedures.

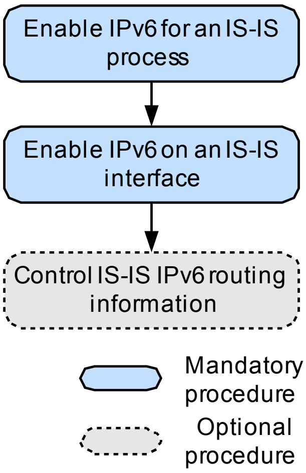
Context

Before configuring IS-IS IPv6 features, complete the following tasks:

- Enabling IPv6 forwarding in the system view
- Configuring IPv6 addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

Figure 7-7 Flowchart of configuring IS-IS IPv6 features


Enable IPv6 for an IS-IS process
Enable IPv6 on an IS-IS interface
Control IS-IS IPv6 routing information
Mandatory procedure
Optional procedure

7.13.1 Enabling IPv6 for an IS-IS Process

Before configuring IS-IS IPv6, you need to enable the IPv6 capability for an IS-IS process.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
ipv6 enable
```

IPv6 is enabled for the IS-IS process.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.13.2 Enabling IPv6 on an IS-IS Interface

After enabling IPv6 for an IS-IS process, you need to enable IPv6 on an IS-IS interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis ipv6 enable [ process-id ]
```

IPv6 is enabled on the specified IS-IS interface.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.13.3 (Optional) Controlling IS-IS IPv6 Routing Information

You can control IS-IS route selection by configuring IS-IS IPv6 features.

Procedure

- Configure the preference of an IS-IS IPv6 route:
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     isis [ process-id ]
     ```
     The IS-IS view is displayed.
  3. Run:
     ```
     ipv6 preference { preference | route-policy route-policy-name } *
     ```
     The preference of an IS-IS route is set.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.
- Do as follows to configure IS-IS to generate default IPv6 routes:
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     isis [ process-id ]
     ```
     The IS-IS view is displayed.

3. Run:

   ipv6 default-route-advertise [**always** | **match default** | **route-policy** *route-policy-name*] [**cost** *cost*] [**tag** *tag*] [**level-1** | **level-2** | **level-1-2**] [**avoid-learning**]

   IS-IS is configured to generate default IPv6 routes.

   After this command is used, IS-IS generates default IPv6 routes and advertises them to IS-IS devices of the related levels.

   When IS-IS is configured to generate default IPv6 routes, the level of the default IPv6 routes is Level-2 if no level is specified.

4. Run:

   commit

   The configuration is committed.

- Do as follows to configure IS-IS to filter the received IPv6 routes before adding the routes that match the filter policy to the IPv6 routing table:

1. Run:

   system-view

   The system view is displayed.

2. Run:

   isis [*process-id*]

   The IS-IS view is displayed.

3. Run:

   ipv6 filter-policy { **acl6-name** *acl6-name-string* | *acl6-number* | **ipv6-prefix** *ipv6-prefix-name* | **route-policy** *route-policy-name* } **import**

   IS-IS is configured to filter the received IPv6 routes before adding the routes that match the filter policy to the IPv6 routing table.

4. Run:

   commit

   The configuration is committed.

- Do as follows to configure IS-IS to import external IPv6 routes:

1. Run:

   system-view

   The system view is displayed.

2. Run:

   isis [*process-id*]

   The IS-IS view is displayed.

3. Run:

   ipv6 import-route *protocol* [*process-id*] [**cost** *cost*] [**tag** *tag*] [**route-policy** *route-policy-name*] [**level-1** | **level-2** | **level-1-2**]

   External IPv6 routes are imported.

4. Run:

   ipv6 filter-policy { **acl6-name** *acl6-name-string* | *acl6-number* | **ipv6-prefix** *ipv6-prefix-name* | **route-policy** *route-policy-name* } **export** [*protocol* [*process-id*]]

IS-IS is configured to filter the imported IPv6 routes before advertising them to other devices.

The **filter-policy export** command usually works with the **ipv6 import-route** command. It filters only the imported routes, which are to be advertised to other devices. If the parameter *protocol* is not specified, the command filters the routes imported from all the protocols. If the parameter *protocol* is specified, the command filters only the routes imported from the specific protocol.

If no level is specified in the **import-route** command, routes are imported to the Level-2 routing table by default.

5. Run:

```
commit
```

The configuration is committed.

- Do as follows to configure IS-IS IPv6 route leaking:

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

3. Run:

```
ipv6 import-route isis level-2 into level-1 [ filter-policy { acl6-number
| acl6-name acl6-name | ipv6-prefix ipv6-prefix-name | route-policy route-
policy-name } ] [ tag tag ]
```

IS-IS route leaking is enabled. Then, IPv6 routes in Level-2 area and other Level-1 areas can be leaked to the Level-1 area where Level-1-2 devices reside. This command is used on Level-1-2 devices connected to external areas.

4. Run:

```
ipv6 import-route isis level-1 into level-2 [ filter-policy { acl6-number
| acl6-name acl6-name | ipv6-prefix ipv6-prefix-name | route-policy route-
policy-name } ] [ tag tag ]
```

Domain-wide prefix distribution is enabled. In this manner, Level 1 intra-area IPv6 routes can be advertised to the Level 2 area where the Level-1-2 device resides. This command is used on Level-1-2 devices connected to Level-2 areas.

NOTE

By default, IS-IS IPv6 routes are advertised from Level-1 areas to Level-2 areas.

5. Run:

```
commit
```

The configuration is committed.

- Do as follows to configure the maximum number of equal-cost IS-IS IPv6 routes:

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

3. Run:

   ipv6 maximum load-balancing *number*

   The maximum number of equal-cost IS-IS IPv6 routes is configured.

4. Run:

   commit

   The configuration is committed.

----End

7.13.4 Checking the Configuration

After configuring IS-IS IPv6, you can view information about the IS-IS interfaces, LSDBs, neighbors, routes, and statistics about the IS-IS process.

Prerequisite

The configurations of IS-IS IPv6 are complete.

Procedure

- Run the **display isis interface** [**traffic-eng**] [**verbose**] [*process-id*] command to view the information about the interface enabled with IS-IS.
- Run the **display isis lsdb** [**verbose**] [**level-1** | **level-2**] [**local** | *lsp-id*] [*process-id*] command to view the IS-IS LSDB.
- Run the **display isis peer** [*process-id*] [**verbose**] command to view the information about IS-IS neighbors.
- Run the **display isis route** [**level-1** | **level-2**] [*process-id*] [[**ipv4**] [*ip-address* [*mask* | *mask-length*]] | [**ipv6**] [*ipv6-address* [*prefix-length*]]] [**verbose**] command to view IS-IS routing information.
- Run the **display isis statistics** [**level-1** | **level-2** | **level-1-2**] [*process-id*] command to view statistics on an IS-IS process.

----End

Example

Run the **display isis interface verbose** command, and you can view that the cost of the IPv6 route of GigabitEthernet 6/0/0 is 20.

```
<HUAWEI> display isis interface verbose
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id        IPV4.State          IPV6.State      MTU  Type  DIS
 GE6/0/0          1    Mtu:Up/Lnk:Dn/IP:Dn         Up          1497 L1/L2 No/No
  Circuit MT State            : Standard
  SNPA Address                : 00e0-c72d-da01
  IP Address                  :
  IPV6 Link Local Address     : FF80::FFE0:FFFF:FE2D:DA01
  IPV6 Global Address(es)     : 2001::1/64
  Csnp Timer Value            :  L1       10  L2       10
  Hello Timer Value           :  L1    10000  L2    10000 <ms>
  DIS Hello Timer Value       :  L1     3333  L2     3333 <ms>
  Hello Multiplier Value      :  L1      3  L2      3
  LSP-Throttle Timer          :  L12   50
```

```
Cost                       :  L1     10  L2     10
Ipv6 Cost                 :  L1     20  L2     20
Priority                   :  L1     64  L2     64
Retransmit Timer Value     :  L1     5   L2     5
Bandwidth-Value            :  Low  1000000000  High  0
Static Bfd                 :  NO
Dynamic Bfd                :  NO
Static IPv6 Bfd            :  NO
Dynamic IPv6 Bfd           :  NO
Circuit State              : OSI:Up / IP:Up / Mtu:Up / IpUnnumber:Up
                           : BandWidth:Up / IsEnable:Up / IfNet:Up
Circuit Ipv6 State         : OSI:Up / IP:Down / Mtu:Up / IpUnnumber:Up
                           : BandWidth:Up / IsEnable:Up / IfNet:Up
```

7.14 Configuring IS-IS Auto FRR

This section describes how to configure IS-IS Auto FRR.

Applicable Environment

At present, the VoIP and on-line video services require high-quality real-time transmission. Nevertheless, if an IS-IS fault occurs, multiple processes, including fault detection, LSP update, LSP flooding, route calculation, and FIB entry delivery, must be performed to switch the traffic to a new link. As a result, it takes much more than 50 ms to recover the link from the fault, which cannot meet the requirement for real-time services on the network.

IS-IS Auto FRR ensures fast switchover of traffic to the backup link before the network convergence, avoiding traffic interruption. This protects traffic and improves reliability of an IS-IS network. The NE5000E supports IPv4 and IPv6 IS-IS Auto FRR.

IS-IS Auto FRR is suitable for IP services that require a low delay and low packet loss ratio.

Pre-configuration Tasks

Before configuring IS-IS Auto FRR, complete the following tasks:

- Configuring IP addresses for interfaces to make neighboring nodes reachable at the network layer
- **Configuring basic IS-IS functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS process is enabled and the IS-IS view is displayed.

Step 3 Run:

```
frr
```

The IS-IS FRR view is displayed.

Step 4 Run:

```
loop-free-alternate [ level-1 | level-2 | level-1-2 ]
```

IS-IS Auto FRR is enabled and the loop-free backup route is created.

If the IS-IS level is not specified, IS-IS Auto FRR is enabled on Level-1 and Level-2 to create the backup route.

For detailed information about IS-IS Auto FRR, refer to the *Feature Description - IP Routing*.

NOTE

IS-IS can create the loop-free backup route only if the interface cost is in compliance with the traffic protection inequality of IS-IS Auto FRR.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

All IS-IS Auto FRR configurations are complete.

- Run the **display isis route** [**level-1** | **level-2**] [*process-id* | **vpn-instance** *vpn-instance-name*] [[**ipv4**] [*ip-address* [*mask* | *mask-length*]] | [**ipv6**] [*ipv6-address* [*prefix-length*]]] [**verbose**] command to check information about the primary link and backup link after IS-IS Auto FRR is enabled.
- Run the **display isis spf-tree** [[**level-1** | **level-2**] | **ipv6** | **verbose**] * [*process-id* | **vpn-instance** *vpn-instance-name*] command to check the traffic protection type of IS-IS Auto FRR.

Enable IS-IS Auto FRR, and then check and find information about the backup outbound interface and the backup next hop of the route with the destination address as 100.1.1.0/24. The following is the check result:

```
<HUAWEI> display isis route verbose

                         Route information for ISIS(1)
                         -----------------------------

                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 1            Flags     : A/-/L/-/-
 Priority   : Medium
 NextHop    :                    Interface :              ExitIndex :
      1.0.0.2                            GE1/0/0                  0x00000003
     (B)2.0.0.2                         GE2/0/0                  0x00000004

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set


                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 3            Flags     : -/-/-/-/-

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set
```

7.15 Improving IS-IS Network Security

On a network that requires high security, you can configure IS-IS authentication to improve the security of the IS-IS network.

Applicable Environment

On a network that requires high security, you can configure IS-IS authentication to improve the security of the IS-IS network. IS-IS authentication consists of area or domain authentication and interface authentication.

Pre-configuration Tasks

Before configuring IS-IS authentication, complete the following tasks:

- Configuring IP addresses for interfaces to ensure that neighboring nodes are reachable at the network layer
- **Configuring Basic IS-IS Functions**

Configuration Procedures

You can choose one or more configuration tasks (excluding "Checking the Configuration") as required.

7.15.1 Configuring the Area or Domain Authentication

After IS-IS area or domain authentication is configured, authentication information can be encapsulated into LSPs or SNPs to ensure the security of packet transmission.

Context

Generally, the IS-IS packets to be sent are not encapsulated with authentication information, and the received packets are not authenticated.

If the area authentication is required, the area authentication password is encapsulated into Level-1 LSPs, CSNPs, and PSNPs in a specified mode. The area authentication modes and passwords of the devices in the same area must be consistent; otherwise, IS-IS packets cannot be flooded normally.

Similarly, in domain authentication, the password is also encapsulated into the Level-2 LSPs, CSNPs, and PSNPs in a specified mode. The authentication procedures and parameters of domain authentication are the same as those of area authentication. The area authentication modes and passwords of the devices in the same area must be consistent; otherwise, IS-IS packets cannot be flooded normally.

Whether IS-IS packets can pass area or domain authentication does not affect the establishment of Level-1 or Level-2 neighbor relationships.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run the following command as required:

- To set area authentication, run the **area-authentication-mode** { **simple** *password* | **md5** *password-key* } [**ip** | **osi**] [**snp-packet** { **authentication-avoid** | **send-only** } | **all-send-only**] command.
- To set domain authentication, run the **domain-authentication-mode** { **simple** *password* | **md5** *password-key* } [**ip** | **osi**] [**snp-packet** { **authentication-avoid** | **send-only** } | **all-send-only**] command.

The authentication involves the following situations:

- The device encapsulates the authentication mode into LSPs and SNPs to be sent and checks whether the received packets pass authentication. Then, the device discards the packets that do not pass the authentication. In this case, the parameter **snp-packet** or **all-send-only** is not specified.
- The device encapsulates authentication information into LSPs to be sent and checks whether the received LSPs pass the authentication; the device neither encapsulates the SNPs to be sent with authentication information nor checks whether the received SNPs pass the authentication. In this case, the parameter **snp-packet authentication-avoid** needs to be specified.
- The device encapsulates the LSPs and SNPs to be sent with authentication information; the device, however, checks the authentication mode of only the received LSPs rather than the received SNPs. In this case, the parameter **snp-packet send-only** needs to be specified.
- The device encapsulates the LSPs and SNPs to be sent with authentication information, but does not check whether the received LSPs or SNPs pass the authentication. In this case, the parameter **all-send-only** needs to be specified.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.15.2 Configuring the Interface Authentication

After the IS-IS interface authentication is configured, authentication information can be encapsulated into the Hello packet to confirm the validity and correctness of neighbor relationships.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
isis authentication-mode { simple password | md5 password-key } [ level-1 |
level-2 ] [ ip | osi ] [ send-only ]
```

The authentication mode and password are set for an IS-IS interface.

- If the parameter **send-only** is specified, it indicates that the device only encapsulates the Hello packets to be sent with authentication information rather than checks whether the received Hello packets pass the authentication. When the Hello packets do not need to be authenticated on the local device and pass the authentication on the remote device, the two devices can establish the neighbor relationship.
- If the parameter **send-only** is not specified, ensure that passwords of all interfaces with the same level on the same network are the same.

When IS-IS interfaces are Level-1-2 interfaces and **level-1** or **level-2** is not configured in the command, authentication modes and passwords are configured for both Level-1 and Level-2 Hello packets.

When you configure the interface authentication, whether **ip** and **osi** is configured in independent of the actual networking environment.

NOTE

Parameters **level-1** and **level-2** are applicable to only the Ethernet interface that is enabled with IS-IS.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.15.3 Checking the Configuration

After improving the security of an IS-IS network, you can view the information about IS-IS neighbors to determine whether the IS-IS authentication succeeds.

Prerequisite

The configurations of improving the security of an IS-IS network are complete.

Procedure

Step 1 Run the **display isis peer** [*process-id*] [**verbose**] command to view the information about IS-IS neighbors.

----End

Example

On GigabitEthernet 2/0/0, set the authentication mode to **simple** and password to **123**. Neighbor relationship can be set up when authentication information on both ends are the same. The following is the check result:

```
[~HUAWEI] display isis peer 1 verbose
                          Peer information for ISIS(1)
                          ----------------------------
  System Id    Interface           Circuit Id       State HoldTime Type     PRI
0000.0000.0040 GE2/0/0           0000.0000.0040.04  Up    7/s      L1       64
  MT IDs Supported   : 0(Up)
  Local MT IDs       : 0
  Area Address(es)   : 10
  Peer IP Address(es): 12.40.41.1
  Uptime             : 00:01:08
  Adj Protocol       :  IPV4
0000.0000.0040 GE2/0/0           0000.0000.0040.04  Up    8/s      L2       64
  MT IDs Supported   : 0(Up)
  Local MT IDs       : 0
  Area Address(es)   : 10
  Peer IP Address(es): 12.40.41.1
  Uptime             : 00:01:14
  Adj Protocol       :  IPV4
Total Peer(s): 2
```

7.16 Maintaining IS-IS

Maintaining IS-IS involves resetting IS-IS, clearing the IS-IS statistics, and debugging IS-IS.

7.16.1 Resetting IS-IS

Resetting an IS-IS process clears all the data of the IS-IS process and re-establishes the adjacency.

Procedure

- Run the **reset isis all** [*process-id* | **vpn-instance** *vpn-instance-name*] command to reset IS-IS.

CAUTION

Resetting an IS-IS process may cause service interruption. Therefore, exercise caution before using this command.

----End

7.16.2 Suppressing IS-IS

You can disable an IS-IS process temporarily by suppressing IS-IS without affecting IS-IS configurations.

Procedure

Step 1 Run the **system-view** command to enter the system view.

Step 2 Run the **isis** [*process-id*] command to start the specified IS-IS process and enter the IS-IS view.

Step 3 Run the **shutdown** command to disable the IS-IS process temporarily.

After the IS-IS process is disabled temporarily, you can still perform IS-IS configurations but the configurations do not take effect. You can run the **undo shutdown** command to cancel the suppression.

Step 4 Run the **commit** command to commit the configuration.

----End

7.17 Configuration Examples

This section describes IS-IS configuration examples. You can understand the configuration procedures through the configuration flowchart. Each configuration example consists of such information as the networking requirements, configuration notes, and configuration roadmap.

7.17.1 Example for Configuring Basic IS-IS Functions

This section describes how to configure basic IS-IS functions, including specifying the NET, configuring the IS-IS level, and enabling IS-IS on each device.

Networking Requirements

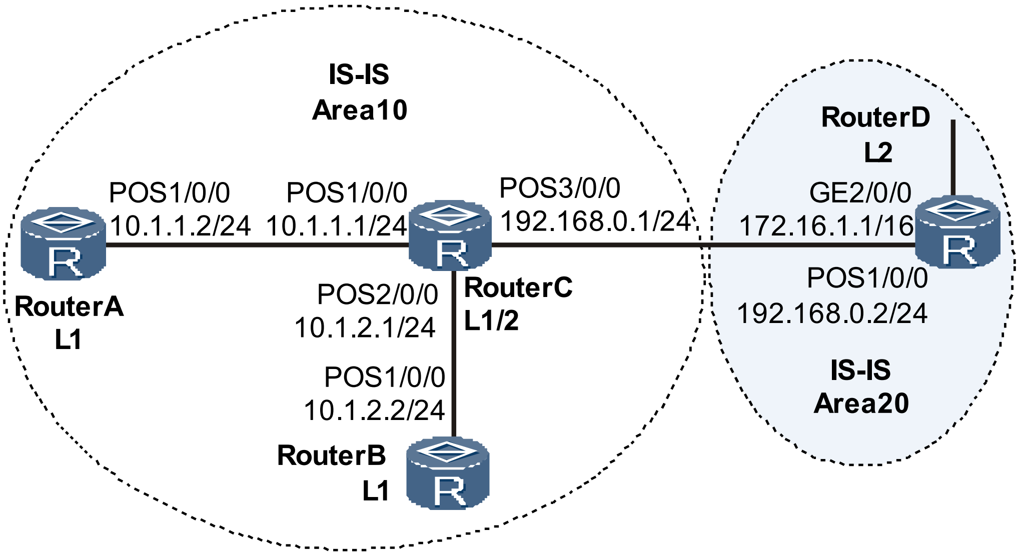
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/ interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-8**:

- Router A, Router B, Router C, and Router D run IS-IS for IP interworking.
- Router A, Router B, and Router C belong to Area 10, and Router D belongs to Area 20.
- Router A and Router B are Level-1 devices; Router C is a Level-1-2 device; Router D is a Level-2 device.

Figure 7-8 Networking diagram for configuring basic IS-IS functions



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IS-IS, configure the level, and specify the NET on each router.
2. Configure Router A and Router C to authenticate Hello packets in a specified mode and with the specified password.
3. View the IS-IS LSDB and the routing table of each router.

Data Preparation

To complete the configuration, you need the following data:

- Area addresses of Router A, Router B, Router C, and Router D
- Levels of Router A, Router B, Router C, and Router D

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure basic IS-IS functions.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] is-level level-1
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] quit
[~RouterA] interface pos 1/0/0
[~RouterA-Pos1/0/0] isis enable 1
[~RouterA-Pos1/0/0] commit
[~RouterA-Pos1/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
```

```
[~RouterB-isis-1] is-level level-1
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface pos 1/0/0
[~RouterB-Pos1/0/0] isis enable 1
[~RouterA-Pos1/0/0] commit
[~RouterB-Pos1/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis 1
[~RouterC-isis-1] network-entity 10.0000.0000.0003.00
[~RouterC-isis-1] quit
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] isis enable 1
[~RouterC-Pos1/0/0] quit
[~RouterC] interface pos 2/0/0
[~RouterC-Pos2/0/0] isis enable 1
[~RouterC-Pos2/0/0] quit
[~RouterC] interface pos 3/0/0
[~RouterC-Pos3/0/0] isis enable 1
[~RouterC-Pos3/0/0] commit
[~RouterC-Pos3/0/0] quit
```

# Configure Router D.

```
[~RouterD] isis 1
[~RouterD-isis-1] is-level level-2
[~RouterD-isis-1] network-entity 20.0000.0000.0004.00
[~RouterD-isis-1] quit
[~RouterD] interface gigabitethernet 2/0/0
[~RouterD-GigabitEthernet2/0/0] isis enable 1
[~RouterD-GigabitEthernet2/0/0] quit
[~RouterD] interface pos 1/0/0
[~RouterD-Pos1/0/0] isis enable 1
[~RouterD-Pos1/0/0] commit
[~RouterD-Pos1/0/0] quit
```

Step 3 Configure the authentication mode and password used by Router A and Router C to authenticate Hello packets.

# Configure Router A.

```
[~RouterA] interface pos 1/0/0
[~RouterA-Pos1/0/0] isis authentication-mode md5 huawei
[~RouterA-Pos1/0/0] commit
[~RouterA-Pos1/0/0] quit
```

# Configure Router C.

```
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] isis authentication-mode md5 huawei
[~RouterC-Pos1/0/0] commit
[~RouterC-Pos1/0/0] quit
```

Step 4 Verify the configuration.

# View the IS-IS LSDB of each router.

```
[~RouterA] display isis lsdb
                        Database information for ISIS(1)
                        --------------------------------
-------------------------------------------------------------------------------
                          Level-1 Link State Database
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0001.00-00* 0x00000006   0xbf7d        649           68      0/0/0
0000.0000.0002.00-00  0x00000003   0xef4d        545           68      0/0/0
0000.0000.0003.00-00  0x00000008   0x3340        582           111     1/0/0
```

```
*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
[~RouterB] display isis lsdb
                        Database information for ISIS(1)
                        --------------------------------
                          Level-1 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0001.00-00  0x00000006   0xbf7d        642           68      0/0/0
0000.0000.0002.00-00* 0x00000003   0xef4d        538           68      0/0/0
0000.0000.0003.00-00  0x00000008   0x3340        574           111     1/0/0
*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
[~RouterC] display isis lsdb
                        Database information for ISIS(1)
                        --------------------------------
                          Level-1 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0001.00-00  0x00000006   0xbf7d        638           68      0/0/0
0000.0000.0002.00-00  0x00000003   0xef4d        533           68      0/0/0
0000.0000.0003.00-00* 0x00000008   0x3340        569           111     1/0/0
*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
                          Level-2 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0003.00-00* 0x00000008   0x55bb        650           100     0/0/0
0000.0000.0004.00-00  0x00000005   0x651         629           84      0/0/0
*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
[~RouterD] display isis lsdb
                        Database information for ISIS(1)
                        --------------------------------
                          Level-2 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0003.00-00  0x00000008   0x55bb        644           100     0/0/0
0000.0000.0004.00-00* 0x00000005   0x651         624           84      0/0/0
*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
           ATT-Attached, P-Partition, OL-Overload
```

# View the IS-IS routing table of each router. In the routing table of a Level-1 device, there must be a default route with the next hop being a Level-1-2 device. The routing table of a Level-2 device must have all Level-1 and Level-2 routes.

```
[~RouterA] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------
 IPV4 Destination   IntCost   ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------
 10.1.1.0/24        10        NULL    P1/0/0          Direct        D/-/L/-/-
 10.1.2.0/24        20        NULL    P1/0/0          10.1.1.1      A/-/-/-/-
 192.168.0.0/24     20        NULL    P1/0/0          10.1.1.1      A/-/-/-/-
 0.0.0.0/0          10        NULL    P1/0/0          10.1.1.1      A/-/-/-/-
    Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
[~RouterC] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------
 IPV4 Destination   IntCost   ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------
```

```
 10.1.1.0/24        10        NULL    P1/0/0          Direct        D/-/L/-/-
 10.1.2.0/24        10        NULL    P2/0/0          Direct        D/-/L/-/-
 192.168.0.0/24     10        NULL    P3/0/0          Direct        D/-/L/-/-
    Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------
 IPV4 Destination   IntCost   ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------
 10.1.1.0/24        10        NULL    -               Direct        D/-/L/-/-
 10.1.2.0/24        10        NULL    -               Direct        D/-/L/-/-
 192.168.0.0/24     10        NULL    -               Direct        D/-/L/-/-
 172.16.0.0/16      20        NULL    P3/0/0          192.168.0.2   A/-/-/-/-
    Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
[~RouterD] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------
 IPV4 Destination   IntCost   ExtCost ExitInterface   NextHop          Flags
--------------------------------------------------------------------------
 192.168.0.0/24     10        NULL    P3/0/0          Direct         D/-/L/-/-
 10.1.1.0/24        20        NULL    P3/0/0          192.168.0.1    A/-/-/-/-
 10.1.2.0/24        20        NULL    P3/0/0          192.168.0.1    A/-/-/-/-
 172.16.0.0/16      10        NULL    GE2/0/0         Direct         D/-/L/-/-
    Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 is-level level-1
 network-entity 10.0000.0000.0001.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
 isis authentication-mode md5 N`C55QK<`=/Q=^Q`MAF4<1!!
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 is-level level-1
 network-entity 10.0000.0000.0002.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.2 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
```

```
 network-entity 10.0000.0000.0003.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
 isis enable 1
 isis authentication-mode md5 N`C55QK<`=/Q=^Q`MAF4<1!!
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.1 255.255.255.0
 isis enable 1
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.0.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
isis 1
 is-level level-2
 network-entity 20.0000.0000.0004.00
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.0.0
 isis enable 1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.0.2 255.255.255.0
 isis enable 1
#
return
```

7.17.2 Example for Configuring the IS-IS DIS Election

This section describes how to configure the IS-IS DIS election, including configuring basic IS-IS functions and configuring the DIS priority on each device.

Networking Requirements

CAUTION

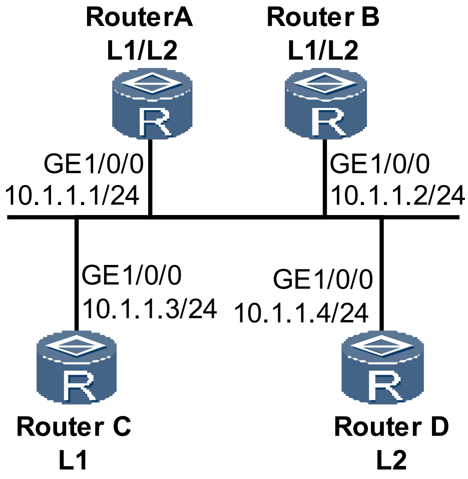
On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-9**:

- Router A, Router B, Router C, and Router D run IS-IS for IP interworking.

- Router A, Router B, Router C, and Router D belong to Area 10, and the network type is broadcast (Ethernet in this case).
- Router A and Router B are Level-1-2 devices; Router C is a Level-1 device; Router D is a Level-2 device.
- The DIS priority of the interface on Router A is 100.
- It is required to change the DIS priority of the interface to configure Router A as a Level-1-2 DIS (DR).

Figure 7-9 Networking diagram for configuring the IS-IS DIS election



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IS-IS and specify the NET on each router for interworking.
2. View information about the IS-IS interface on each router with the default DIS priority.
3. Configure the DIS priority of the interface on each router.

Data Preparation

To complete the configuration, you need the following data:

- Area addresses of the four routers
- Levels of the four routers
- DIS priority of the interface on Router A

Procedure

Step 1 Configure an IPv4 address for each interface. The configuration details are not mentioned here.

Step 2 View the MAC address of the GE interface on each router. When the DIS priority of each interface is the same, the router whose interface has the highest MAC address is elected as the DIS.

# View the MAC address of Gigabit Ethernet 1/0/0 on Router A.

```
[~RouterA] display arp interface gigabitethernet 1/0/0
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE        INTERFACE      VPN-INSTANCE
                                          VLAN/CEVLAN PVC
-------------------------------------------------------------------------
10.1.1.1        00e0-fc10-afec            I           GE1/0/0
-------------------------------------------------------------------------
Total:1         Dynamic:0       Static:0    Interface:1
```

# View the MAC address of Gigabit Ethernet 1/0/0 on Router B.

```
[~RouterB] display arp interface gigabitethernet 1/0/0
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE        INTERFACE      VPN-INSTANCE
                                          VLAN/CEVLAN PVC
-------------------------------------------------------------------------
10.1.1.2        00e0-fccd-acdf            I           GE1/0/0
-------------------------------------------------------------------------
Total:1         Dynamic:0       Static:0    Interface:1
```

# View the MAC address of Gigabit Ethernet 1/0/0 on Router C.

```
[~RouterC] display arp interface gigabitethernet 1/0/0
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE        INTERFACE      VPN-INSTANCE
                                          VLAN/CEVLAN PVC
-------------------------------------------------------------------------
10.1.1.3        00e0-fc50-25fe            I           GE1/0/0
-------------------------------------------------------------------------
Total:1         Dynamic:0       Static:0    Interface:1
```

# View the MAC address of Gigabit Ethernet 1/0/0 on Router D.

```
[~RouterD] display arp interface gigabitethernet 1/0/0
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE        INTERFACE      VPN-INSTANCE
                                          VLAN/CEVLAN PVC
-------------------------------------------------------------------------
10.1.1.4        00e0-fcfd-305c            I           GE1/0/0
-------------------------------------------------------------------------
Total:1         Dynamic:0       Static:0    Interface:1
```

Step 3 Enable IS-IS.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] quit
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis enable 1
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] isis enable 1
[~RouterB-GigabitEthernet1/0/0] commit
[~RouterB-GigabitEthernet1/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis 1
[~RouterC-isis-1] network-entity 10.0000.0000.0003.00
[~RouterC-isis-1] is-level level-1
[~RouterC-isis-1] quit
[~RouterC] interface gigabitethernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] isis enable 1
[~RouterC-GigabitEthernet1/0/0] commit
[~RouterC-GigabitEthernet1/0/0] quit
```

# Configure Router D.

```
[~RouterD] isis 1
[~RouterD-isis-1] network-entity 10.0000.0000.0004.00
[~RouterD-isis-1] is-level level-2
[~RouterD-isis-1] quit
[~RouterD] interface gigabitethernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] isis enable 1
[~RouterD-GigabitEthernet1/0/0] commit
[~RouterD-GigabitEthernet1/0/0] quit
```

# View information about IS-IS neighbors of Router A.

```
[~RouterA] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id     Interface         Circuit Id        State  HoldTime Type     PRI
0000.0000.0002 GE1/0/0          0000.0000.0002.01  Up    9s       L1(L1L2) 64
0000.0000.0003 GE1/0/0          0000.0000.0002.01  Up    27s      L1       64
0000.0000.0002 GE1/0/0          0000.0000.0004.01  Up    28s      L2(L1L2) 64
0000.0000.0004 GE1/0/0          0000.0000.0004.01  Up    8s       L2       64
Total Peer(s): 4
```

# View information about the IS-IS interface on Router A.

```
[~RouterA] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
 GE1/0/0          1          Up          Mtu:Up/Lnk:Dn/IP:Dn   1497 L1/L2 No/No
```

# View information about the IS-IS interface on Router B.

```
[~RouterB] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State          MTU  Type  DIS
 GE1/0/0          1         Up           Mtu:Up/Lnk:Dn/IP:Dn    1497 L1/L2 Yes/No
```

# View information about the IS-IS interface on Router D.

```
[~RouterD] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State          MTU  Type  DIS
 GE1/0/0          1         Up           Mtu:Up/Lnk:Dn/IP:Dn    1497 L1/L2 No/Yes
```

As shown in the preceding interface information, when the default DIS priority is used, the MAC address of the interface on Router B is the highest among those of the interfaces on Level-1 devices. Therefore, Router B is elected as the Level-1 DIS. The MAC address of the interface on Router D is the highest among those of the interfaces on Level-2 devices. Therefore, Router D is elected as the Level-2 DIS. Level-1 and Level-2 pseudo nodes are 0000.0000.0002.01 and 0000.0000.0004.01 respectively.

Step 4 Configure the DIS priority of the IS-IS interface on Router A.

```
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis dis-priority 100
[~RouterA-GigabitEthernet1/0/0] commit
```

# View information about IS-IS neighbors of Router A.

```
[~RouterA] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id     Interface         Circuit Id        State  HoldTime Type     PRI
0000.0000.0002 GE1/0/0          0000.0000.0001.01  Up    21s      L1(L1L2) 64
0000.0000.0003 GE1/0/0          0000.0000.0001.01  Up    27s      L1       64
0000.0000.0002 GE1/0/0          0000.0000.0001.01  Up    28s      L2(L1L2) 64
```

```
0000.0000.0004 GE1/0/0           0000.0000.0001.01  Up    30s      L2       64
Total Peer(s): 4
```

Step 5 Verify the configuration.

# View information about the IS-IS interface on Router A.

```
[~RouterA] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State         MTU  Type  DIS
 GE1/0/0          1          Up          Mtu:Up/Lnk:Dn/IP:Dn   1497 L1/L2 Yes/Yes
```

As shown in the preceding information, after the DIS priority of the IS-IS interface is changed, Router A becomes a Level-1-2 DIS (DR) immediately and its pseudo node is 0000.0000.0001.01.

# View information about IS-IS neighbors and the IS-IS interface on Router B.

```
[~RouterB] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id     Interface          Circuit Id       State  HoldTime Type     PRI
0000.0000.0001 GE1/0/0           0000.0000.0001.01 Up    7s       L1(L1L2) 100
0000.0000.0003 GE1/0/0           0000.0000.0001.01 Up    25s      L1       64
0000.0000.0001 GE1/0/0           0000.0000.0001.01 Up    7s       L2(L1L2) 100
0000.0000.0004 GE1/0/0           0000.0000.0001.01 Up    25s      L2       64
Total Peer(s): 4
 [~RouterB] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State          MTU  Type  DIS
 GE1/0/0          1          Up          Mtu:Up/Lnk:Dn/IP:Dn    1497 L1/L2 No/No
```

# View information about IS-IS neighbors and the IS-IS interface on Router D.

```
[~RouterD] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id     Interface          Circuit Id       State  HoldTime Type     PRI
0000.0000.0001 GE1/0/0           0000.0000.0001.01 Up    9s       L2       100
0000.0000.0002 GE1/0/0           0000.0000.0001.01 Up    28s      L2       64
Total peer(s): 2
[~RouterD] display isis interface 1
                       Interface information for ISIS(1)
                       ---------------------------------
 Interface       Id      IPV4.State          IPV6.State          MTU  Type  DIS
 GE1/0/0          1          Up          Mtu:Up/Lnk:Dn/IP:Dn    1497 L1/L2 No/No
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.1.1.1 255.255.255.0
 isis enable 1
 isis dis-priority 100 level-1
 isis dis-priority 100 level-2
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
 is-level level-1
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.1.1.3 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.1.1.4 255.255.255.0
 isis enable 1
#
return
```

7.17.3 Example for Configuring IS-IS Load Balancing

This section describes how to configure IS-IS load balancing.

Networking Requirements

CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

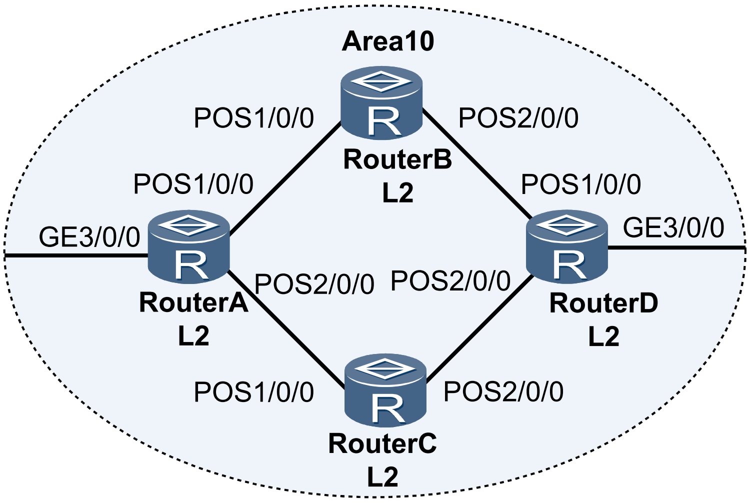
As shown in **Figure 7-10**:

- Router A, Router B, Router C, and Router D run IS-IS for IP interworking.

- Router A, Router B, Router C, and Router D are Level-2 devices in Area 10.

It is required that Router A send traffic to Router D through load balancing between Router B and Router C.

Figure 7-10 Networking diagram of configuring IS-IS load balancing



Device	Interface	IP Address
Router A	GE 3/0/0	172.16.1.1/24
	POS 1/0/0	10.1.1.1/24
	POS 2/0/0	10.1.2.1/24
Router B	POS 1/0/0	10.1.1.2/24
	POS 2/0/0	192.168.0.1/24
Router C	POS 1/0/0	10.1.2.2/24
	POS 2/0/0	192.168.1.1/24
Router D	GE 3/0/0	172.17.1.1/24
	POS 1/0/0	192.168.0.2/24
	POS 2/0/0	192.168.1.2/24

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic IS-IS functions on each router for IP interworking.
2. Cancel load balancing and view the routing table.
3. Configure load balancing on Router A and view the routing table.
4. Configure load balancing on Router A.
5. (Optional) Configure the preference for equal-cost routes on Router A.

Data Preparation

To complete the configuration, you need the following data:

- Area addresses and levels of the four routers
- Number (1 in this case) of equal-cost routes for load balancing on Router A
- Load balancing mode on Router A
- Preference (1 in this case) of equal-cost routes on Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure basic IS-IS functions. The configuration details are not mentioned here.

Step 3 Cancel load balancing on Router A, that is, set the maximum number of equal-cost routes for load balancing to 1.

```
[~RouterA] isis 1
[~RouterA-isis-1] maximum load-balancing 1
[~RouterA-isis-1] commit
[~RouterA-isis-1] quit
```

# View the routing table of Router A.

```
[~RouterA] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------
 IPV4 Destination    IntCost   ExtCost ExitInterface   NextHop          Flags
---------------------------------------------------------------------------
 192.168.1.0/24      20        NULL    P2/0/0          10.1.2.2       A/-/-/-/-
 10.1.1.0/24         10        NULL    P1/0/0          Direct         D/-/L/-/-
 172.16.1.0/24       10        NULL    GE3/0/0         Direct         D/-/L/-/-
 172.17.1.0/24       30        NULL    P1/0/0          10.1.1.2       A/-/-/-/-
 10.1.2.0/24         10        NULL    P2/0/0          Direct         D/-/L/-/-
 192.168.0.0/24      20        NULL    P1/0/0          10.1.1.2       A/-/-/-/-
     Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

As shown in the routing table, when the maximum number of equal-cost routes for load balancing is set to 1, the next hop of the route to network segment 172.17.1.0 is 10.1.1.2. This is because the Ifindex of the interface on Router B is greater, and IS-IS chooses the route with the next hop being 10.1.1.2 as the unique optimal route.

Step 4 Restore the number of equal-cost routes for load balancing on Router A to the default value 3.

```
[~RouterA] isis 1
[~RouterA-isis-1] undo maximum load-balancing
[~RouterA-isis-1] commit
[~RouterA-isis-1] quit
```

# View the routing table of Router A.

```
[~RouterA] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------
 IPV4 Destination    IntCost   ExtCost ExitInterface   NextHop          Flags
---------------------------------------------------------------------------
 192.168.1.0/24      20        NULL    P2/0/0          10.1.2.2       A/-/-/-/-
 10.1.1.0/24         10        NULL    P1/0/0          Direct         D/-/L/-/-
 172.16.1.0/24       10        NULL    GE3/0/0         Direct         D/-/L/-/-
 172.17.1.0/24       30        NULL    P1/0/0          10.1.1.2       A/-/-/-/-
```

```
                                       P2/0/0          10.1.2.2
 10.1.2.0/24         10        NULL    P2/0/0          Direct         D/-/L/-/-
 192.168.0.0/24      20        NULL    P1/0/0          10.1.1.2       A/-/-/-/-
     Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

As shown in the routing table, when the maximum number of equal-cost routes for load balancing is restored to the default value 3, the two routes with next hops being 10.1.1.2 (Router B) and 10.1.2.2 (Router C) become valid.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
 isis enable 1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
 isis enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0002.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.0.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
```

```
 is-level level-2
 network-entity 10.0000.0000.0003.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.2 255.255.255.0
 isis enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0004.00
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
 isis enable 1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.0.2 255.255.255.0
 isis enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.2 255.255.255.0
 isis enable 1
#
return
```

7.17.4 Example for Configuring IS-IS to Interact with BGP

This section describes how to configure IS-IS to interact with BGP, including configuring BGP and IS-IS to import routes from each other.

Networking Requirements

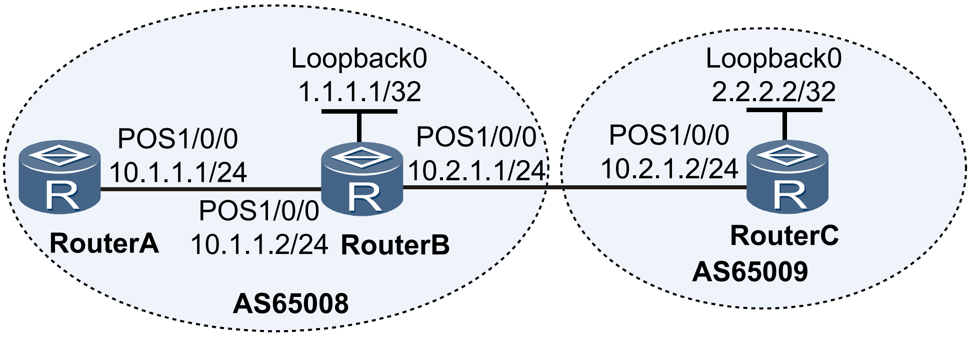
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-11**:

- Router A and Router B belong to the same AS, and the IS-IS neighbor relationship is established between the two devices. Router A is a non-BGP device in the AS.
- An EBGP connection is established between Router B and Router C. When IS-IS imports BGP routes, it is required to change the cost of a route by applying the routing policy.

Figure 7-11 Networking diagram for configuring IS-IS to interact with BGP



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable IS-IS and specify NETs on device A and device B.
2. Configure an EBGP connection on device B and device C.
3. Configure IS-IS and BGP to import routes from each other on Router B, and then check the routes.

Data Preparation

To complete the configuration, you need the following data:

- Area addresses of Router A and Router B
- RouterID and AS number of Router B
- RouterID and AS number of Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure basic IS-IS functions.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] quit
[~RouterA] interface pos 1/0/0
[~RouterA-Pos1/0/0] isis enable 1
[~RouterA-Pos1/0/0] commit
[~RouterA-Pos1/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
```

```
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface pos 1/0/0
[~RouterB-Pos1/0/0] isis enable 1
[~RouterB-Pos1/0/0] commit
[~RouterB-Pos1/0/0] quit
```

Step 3 Configure an EBGP connection.

# Configure Router B.

```
[~RouterB] bgp 65008
[~RouterB-bgp] router-id 1.1.1.1
[~RouterB-bgp] peer 10.2.1.2 as-number 65009
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] network 10.2.1.0 255.255.255.0
[~RouterB-bgp-af-ipv4] commit
```

# Configure Router C.

```
[~RouterC] bgp 65009
[~RouterC-bgp] router-id 2.2.2.2
[~RouterC-bgp] peer 10.2.1.1 as-number 65008
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] network 10.2.1.0 255.255.255.0
[~RouterB-bgp-af-ipv4] commit
```

Step 4 Configure IS-IS to import BGP routes.

# Configure a static route on Router C.

```
[~RouterC] ip route-static 200.1.1.1 32 NULL 0
[~RouterC] commit
```

# On Router C, configure BGP to import the static route.

```
[~RouterC] bgp 65009
[~RouterC-bgp] import-route static
[~RouterC-bgp] commit
```

# On Router B, configure IS-IS to import the BGP route.

```
[~RouterB] isis 1
[~RouterB-isis-1] import-route bgp
[~RouterB-isis-1] commit
[~RouterB-isis-1] quit
```

# View the routing table of Router A, and you can find that IS-IS successfully imports the BGP route 200.1.1.1/32.

```
[~RouterA] display ip routing-table
Route Flags: R - relied, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 6        Routes : 6

Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface

       10.1.1.0/24  Direct 0    0           D  10.1.1.1        Pos1/0/0
       10.1.1.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
       10.1.1.2/32  Direct 0    0           D  10.1.1.2        Pos1/0/0
       127.0.0.0/8  Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      200.1.1.1/32  ISIS  15   74          D  10.1.1.2        Pos1/0/0
```

# On Router B, configure the AS_Path filter, and apply the filter in the routing policy named **RTC**.

```
[~RouterB] ip as-path-filter 1 permit 65009
[~RouterB] route-policy RTC permit node 0
```

```
[~RouterB-route-policy] if-match as-path-filter 1
[~RouterB-route-policy] apply cost 20
[~RouterB-route-policy] commit
[~RouterB-route-policy] quit
```

# On Router B, configure IS-IS to import the BGP route.

```
[~RouterB] isis 1
[~RouterB-isis-1] import-route bgp route-policy RTC
[~RouterB-isis-1] commit
[~RouterB-isis-1] quit
```

# View the routing table of device A, and you can find that the AS_Path filter is successfully applied and the cost of the imported route 200.1.1.1/32 changes from 74 to 94.

```
[~RouterA] display ip routing-table
Route Flags: R - relied, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 6        Routes : 6

Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface

       10.1.1.0/24  Direct 0    0           D  10.1.1.1        Pos1/0/0
       10.1.1.1/32  Direct 0    0           D  127.0.0.1       Pos1/0/0
       10.1.1.2/32  Direct 0    0           D  10.1.1.2        Pos1/0/0
       127.0.0.0/8  Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      200.1.1.1/32  ISIS   15   94          D  10.1.1.2        Pos1/0/0
```

Step 5 Configure BGP to import IS-IS routes.

```
[~RouterB] bgp 65008
[~RouterB-bgp] import-route isis 1
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# View the routing table of Router C, and you can find that BGP successfully imports the IS-IS route 10.1.1.0/24.

```
[~RouterC] display ip routing-table
Route Flags: R - relied, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 7        Routes : 7

Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface

       10.1.1.0/24  BGP    255  0           D  10.2.1.1        Pos1/0/0
       10.2.1.0/24  Direct 0    0           D  10.2.1.2        Pos1/0/0
       10.2.1.1/32  Direct 0    0           D  10.2.1.1        Pos1/0/0
       10.2.1.2/32  Direct 0    0           D  127.0.0.1       InLoopBack0
       127.0.0.0/8  Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      200.1.1.1/32  Static 60   0           D  0.0.0.0         NULL0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 network-entity 10.0000.0000.0001.00
#
interface Pos1/0/0
```

```
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 import-route bgp route-policy RTC
 network-entity 10.0000.0000.0002.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
 isis enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.2.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 65008
 router-id 1.1.1.1
 peer 10.2.1.2 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
  network 10.2.1.0 255.255.255.0
  import-route static
  import-route isis 1
  peer 10.2.1.2 enable
#
route-policy RTC permit node 0
 if-match as-path-filter 1
 apply cost 20
#
ip as-path-filter 1 permit 65009
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.2.1.2 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 65009
 router-id 2.2.2.2
 peer 10.2.1.1 as-number 65008
 #
 ipv4-family unicast
  undo synchronization
  network 10.2.1.0 255.255.255.0
  import-route static
  peer 10.2.1.1 enable
#
```

```
ip route-static 200.1.1.1 255.255.255.255 NULL0
#
return
```

7.17.5 Example for Configuring Static BFD for IS-IS

This section describes how to configure static BFD for IS-IS, including configuring BFD parameters and enabling static BFD.

Networking Requirements

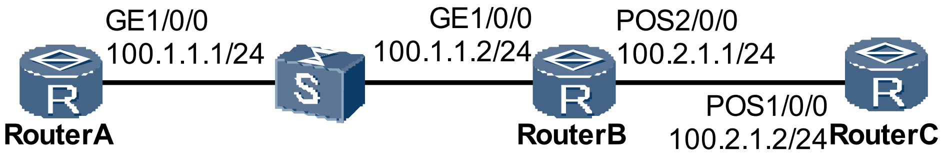
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-12**:

- Router A and Router B are connected through a Layer 2 switch.
- IS-IS runs on Router A, Router B, and Router C.
- BFD is configured to detect the IS-IS neighbor relationship between Router A and Router B. When the link between Router A and Router B becomes faulty, BFD can fast detect the fault and notify it to IS-IS.

Figure 7-12 Networking diagram for configuring static BFD for IS-IS



NOTE

BFD for IS-IS cannot be used to detect the multi-hop link between Router A and Router C. This is because the IS-IS neighbor relationship cannot be established between Router A and Router C.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic IS-IS functions on each router.
2. Enable BFD on Router A and Router B.

Data Preparation

To complete the configuration, you need the following data:

- IS-IS process ID

- Area addresses of Router A, Router B, and Router C
- Levels of Router A, Router B, and Router C
- Name of the BFD session established between Router A and Router B and peer IP address to be detected by BFD
- Local and remote discriminators of the BFD session established between Router A and Router B

Procedure

Step 1 Configure an IP address for the interface on each router. The configuration details are not mentioned here.

Step 2 Configure basic IS-IS functions.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] is-level level-2
[~RouterA-isis-1] network-entity aa.1111.1111.1111.00
[~RouterA-isis-1] quit
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis enable 1
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
[~RouterB-isis-1] is-level level-2
[~RouterB-isis-1] network-entity aa.2222.2222.2222.00
[~RouterB-isis-1] quit
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] isis enable 1
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] interface pos 2/0/0
[~RouterB-Pos2/0/0] isis enable 1
[~RouterB-Pos2/0/0] commit
[~RouterB-Pos2/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis 1
[~RouterC-isis-1] is-level level-2
[~RouterC-isis-1] network-entity aa.3333.3333.3333.00
[~RouterC-isis-1] quit
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] isis enable 1
[~RouterC-Pos1/0/0] commit
[~RouterC-Pos1/0/0] quit
```

# After the preceding configurations, you can view that the neighbor relationship is established between Router A and Router B.

```
[~RouterA] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id    Interface         Circuit Id        State  HoldTime  Type      PRI
2222.2222.2222 GE1/0/0           2222.2222.2222.00  Up    23s       L2        64
Total Peer(s): 1
```

The IS-IS routing table of Router A has the routes to Router B and Router C.

```
[~RouterA] display isis route
                         Route information for ISIS(1)
                         -----------------------------
```

```
                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------
 IPV4 Destination    IntCost   ExtCost ExitInterface   NextHop        Flags
-------------------------------------------------------------------------
 100.1.1.0/24        10        NULL    GE1/0/0         Direct       D/-/L/-
 100.2.1.0/24        20        NULL    GE1/0/0         100.1.1.2    A/-/-/-
      Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
```

Step 3 Configure BFD.

# Enable BFD and configure a BFD session on Router A.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] bfd atob bind peer-ip 100.1.1.2 interface gigabitethernet1/0/0
[~RouterA-bfd-session-atob] discriminator local 1
[~RouterA-bfd-session-atob] discriminator remote 2
[~RouterA-bfd-session-atob] commit
[~RouterA-bfd-session-atob] quit
```

# Enable BFD and configure a BFD session on Router B.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] bfd btoa bind peer-ip 100.1.1.1 interface gigabitethernet1/0/0
[~RouterB-bfd-session-btoa] discriminator local 2
[~RouterB-bfd-session-btoa] discriminator remote 1
[~RouterB-bfd-session-btoa] commit
[~RouterB-bfd-session-btoa] quit
```

# After the preceding configurations, run the **display bfd session** command on Router A or Router B, and you can view that the BFD session is Up.

Take the display on Router A as an example.

```
[~RouterA] display bfd session all
------------------------------------------------------------------------
Local   Remote   PeerIpAddr    State      Type       Interface Name
------------------------------------------------------------------------
1       2        100.1.1.2     Up         S_IP       GE1/0/0
------------------------------------------------------------------------
     Total UP/DOWN Session Number : 1/0
```

Step 4 Enable IS-IS fast sense.

# Configure Router A.

```
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis bfd static
[~RouterA-GigabitEthernet1/0/0] quit
```

# Configure Router B.

```
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] isis bfd static
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

Step 5 Verify the configuration.

# Enable the debugging on Router A.

```
<RouterA> debugging isis adjacency
<RouterA> debugging isis circuit-information
<RouterA> terminal debugging
```

# Run the **shutdown** command on Gigabit Ethernet 1/0/0 on Router D to simulate a link fault.

```
[~RouterB-GigabitEthernet1/0/0] shutdown
```

```
[~RouterB-GigabitEthernet1/0/0] commit
```

# On Router A, you can view the following log information, which indicates that IS-IS deletes the neighbor relationship between Router A and Router B after being notified by BFD of the fault.

```
#80/active/IsisAdjacencyChange/Major/occurredTime:2011-03-09 04:17:07/-/-/alarmI
D:0x08960007/VR=0:ISIS adjacency state change. (SysInstance=1, SysLevel=1, CircI
ndex=2, CircIfIndex=20, LspId=2222.2222.2222.00.00, AdjState=1, IfIndex=20, IfNa
me=GE1/0/0, Reason=The adjacency HoldTimer expired, SubReason=14)
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
isis 1
 is-level level-2
 network-entity aa.1111.1111.1111.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 100.1.1.1 255.255.255.0
 isis enable 1
 isis bfd static
#
bfd atob bind peer-ip 100.1.1.2 interface GigabitEthernet1/0/0
 discriminator local 1
 discriminator remote 2
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
bfd
#
isis 1
 is-level level-2
 network-entity aa.2222.2222.2222.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 100.1.1.2 255.255.255.0
 isis enable 1
 isis bfd static
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 100.2.1.1 255.255.255.0
 isis enable 1
#
bfd btoa bind peer-ip 100.1.1.1 interface GigabitEthernet1/0/0
 discriminator local 2
 discriminator remote 1
#
return
```

- Configuration file of Router C

```
#
```

```
sysname RouterC
#
isis 1
 is-level level-2
 network-entity aa.3333.3333.3333.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 100.2.1.2 255.255.255.0
 isis enable 1
#
return
```

7.17.6 Example for Configuring Dynamic BFD for IS-IS

This section describes how to configure dynamic BFD for IS-IS, including configuring basic IS-IS functions, setting the interface cost, configuring BFD for the IS-IS process, and configuring BFD for the IS-IS interface on each device.

Networking Requirements

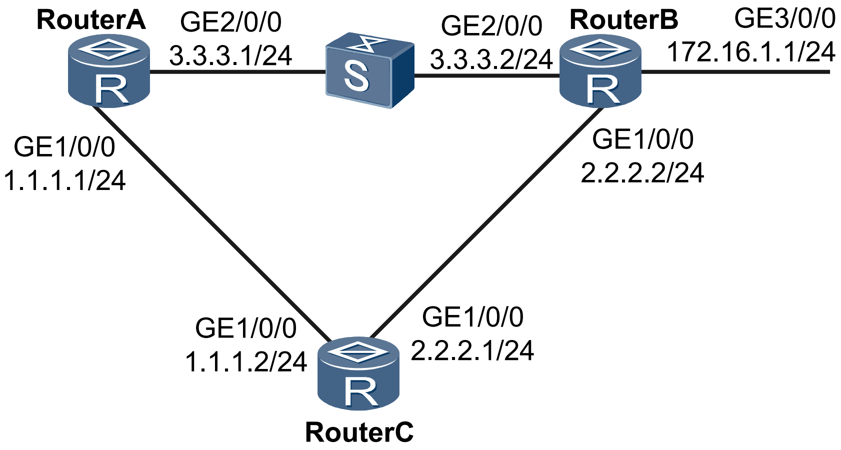
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-13**:

- IS-IS runs on Router A, Router B, and Router C.
- BFD is enabled for the IS-IS processes on Router A, Router B, and Router C.
- Service traffic is transmitted along the primary link Router A → Router B. The link Router A → Router C → Router B functions as the backup link.
- BFD is configured for the interface on the link between Router A and Router B. When the link fails, BFD can fast detect the fault and notify IS-IS of the fault so that service traffic can be transmitted through the backup link.

Figure 7-13 Networking diagram for configuring dynamic BFD for IS-IS



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic IS-IS functions on each router for IP interworking.
2. Set the IS-IS interface cost to control route selection.
3. Enable global BFD.
4. Enable BFD for the IS-IS processes on Router A, Router B, and Router C.
5. Enable BFD for the interfaces on Router A and Router B.

Data Preparation

To complete the configuration, you need the following data:

- IS-IS process ID
- Area addresses of Router A, Router B, and Router C
- Interface costs of Router A and Router B
- Numbers and types of the interfaces to be enabled with BFD on Router A, Router B, and Router C
- Minimum interval for sending BFD packets, minimum interval for receiving BFD packets, and local detection multiplier on Router A and Router B

Procedure

Step 1 Configure an IP address for the interface on each router. The configuration details are not mentioned here.

Step 2 Configure basic IS-IS functions.

# Configure Router A.

```
[~RouterA] isis
[~RouterA-isis-1] is-level level-2
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] quit
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis enable 1
[~RouterA-GigabitEthernet1/0/0] quit
[~RouterA] interface gigabitethernet 2/0/0
[~RouterA-GigabitEthernet2/0/0] isis enable 1
[~RouterA-GigabitEthernet2/0/0] commit
[~RouterA-GigabitEthernet2/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis
[~RouterB-isis-1] is-level level-2
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] isis enable 1
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] isis enable 1
[~RouterB-GigabitEthernet2/0/0] quit
[~RouterB] interface gigabitethernet 3/0/0
```

```
[~RouterB-GigabitEthernet3/0/0] isis enable 1
[~RouterB-GigabitEthernet3/0/0] commit
[~RouterB-GigabitEthernet3/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis
[~RouterC-isis-1] is-level level-2
[~RouterC-isis-1] network-entity 10.0000.0000.0003.00
[~RouterC-isis-1] quit
[~RouterC] interface gigabitethernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] isis enable 1
[~RouterC-GigabitEthernet1/0/0] quit
[~RouterC] interface gigabitethernet 2/0/0
[~RouterC-GigabitEthernet2/0/0] isis enable 1
[~RouterC-GigabitEthernet2/0/0] commit
[~RouterC-GigabitEthernet2/0/0] quit
```

# After the preceding configurations, run the **display isis peer** command. You can view that the neighbor relationships are established between Router A and Router B, and between Router A and Router C. Take the display on Router A as an example.

```
[~RouterA] display isis peer
                          Peer information for ISIS(1)
                          ----------------------------
  System Id     Interface          Circuit Id       State HoldTime Type     PRI
0000.0000.0002  GE2/0/0            0000.0000.0002.01 Up   9s       L2       64
0000.0000.0003  GE1/0/0            0000.0000.0001.02 Up   21s      L2       64
Total Peer(s): 2
```

# The routers have the routes learned from each other.

# Take the routing table of Router A as an example.

```
[~RouterA] display ip routing-table
Route Flags: R - relied, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 8        Routes : 9
Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface
        1.1.1.0/24  Direct 0    0           D  1.1.1.1        GigabitEthernet1/0/0
        1.1.1.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
        2.2.2.0/24  ISIS   15   20         D  1.1.1.2         GigabitEthernet1/0/0
        3.3.3.0/24  Direct 0    0          D  3.3.3.1         GigabitEthernet2/0/0
        3.3.3.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.0/8   Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      172.16.1.0/24 ISIS 15 20 D 3.3.3.2 GigabitEthernet2/0/0
```

As shown in the routing table, the next-hop address of the route to 172.16.1.0/24 is 3.3.3.2, and traffic is transmitted on the primary link Router A -> Router B.

Step 3 Set the interface cost.

# Configure Router A.

```
[~RouterA] interface gigabitethernet 2/0/0
[~RouterA-GigabitEthernet1/0/0] isis cost 5
[~RouterA-GigabitEthernet1/0/0] commit
[~RouterA-GigabitEthernet1/0/0] quit
```

# Configure Router B.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet1/0/0] isis cost 5
[~RouterB-GigabitEthernet1/0/0] commit
[~RouterB-GigabitEthernet1/0/0] quit
```

Step 4 Configure BFD for IS-IS processes.

# Enable BFD for the IS-IS process on Router A.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] isis
[~RouterA-isis-1] bfd all-interfaces enable
[~RouterA-isis-1] commit
[~RouterA-isis-1] quit
```

# Enable BFD for the IS-IS process on Router B.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] isis
[~RouterB-isis-1] bfd all-interfaces enable
[~RouterB-isis-1] commit
[~RouterB-isis-1] quit
```

# Enable BFD for the IS-IS process on Router C.

```
[~RouterC] bfd
[~RouterC-bfd] quit
[~RouterC] isis
[~RouterC-isis-1] bfd all-interfaces enable
[~RouterC-isis-1] commit
[~RouterC-isis-1] quit
```

# After the preceding configurations, run the **display isis bfd session all** command on Router A, Router B, or Router C, and you can view that the BFD session is Up.

Take the display on Router A as an example.

```
[~RouterA] display isis bfd session all
Peer System ID : 0000.0000.0002        Interface : GE2/0/0
BFD State : up        Type : L2
Peer IP Address : 3.3.3.2
Local IP Address: 3.3.3.1

Peer System ID : 0000.0000.0003        Interface : GE1/0/0
BFD State : up        Type : L2
Peer IP Address : 1.1.1.2
Local IP Address: 1.1.1.1

Total BFD session(s): 2
```

The preceding information shows that the BFD sessions between Router A and Router B and between Router A and Router C are Up.

Step 5 Configure BFD for IS-IS interfaces.

# On GE 2/0/0 of Router A, configure BFD, and set the minimum interval for sending BFD packets to 100 ms, the minimum interval for receiving BFD packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterA] interface gigabitEthernet 2/0/0
[~RouterA-GigabitEthernet2/0/0] isis bfd enable
[~RouterA-GigabitEthernet2/0/0] isis bfd min-tx-interval 100 min-rx-interval 100
detect-multiplier 4
[~RouterA-GigabitEthernet2/0/0] commit
[~RouterA-GigabitEthernet2/0/0] quit
```

# On GE 2/0/0 of Router B, configure BFD, and set the minimum interval for sending BFD packets to 100 ms, the minimum interval for receiving BFD packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterB] bfd
```

```
[~RouterB-bfd] quit
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] isis bfd enable
[~RouterB-GigabitEthernet2/0/0] isis bfd min-tx-interval 100 min-rx-interval 100
detect-multiplier 4
[~RouterB-GigabitEthernet2/0/0] commit
[~RouterB-GigabitEthernet2/0/0] quit
```

# After the preceding configurations, run the **display isis bfd session all** command on Router A or Router B, and you can view that BFD parameters already take effect. Take the display on Router B as an example.

```
[~RouterB] display isis bfd session all
                      BFD session information for ISIS(1)
                      -----------------------------------
Peer System ID : 0000.0000.0001        Interface : GE2/0/0
BFD State : up       Type : L2
Peer IP Address : 3.3.3.1
Local IP Address: 3.3.3.2

Peer System ID : 0000.0000.0003        Interface : GE1/0/0
BFD State : up       Type : L2
Peer IP Address : 2.2.2.1
Local IP Address: 2.2.2.2

Total BFD session(s): 2
```

Step 6 # Run the **shutdown** command on Gigabit Ethernet 2/0/0 on Router B to simulate the fault on the primary link.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] shutdown
[~RouterB-GigabitEthernet2/0/0] commit
```

Step 7 Verify the configuration.

# View the routing table of Router A.

```
[~RouterA] display ip routing-table
Route Flags: R - relied, D - download to fib
------------------------------------------------------------------------------
Routing Tables: _public_
         Destinations : 8        Routes : 8
Destination/Mask    Proto  Pre  Cost     Flags NextHop         Interface
        1.1.1.0/24  Direct 0    0           D  1.1.1.1         GigabitEthernet1/0/0
        1.1.1.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
        2.2.2.0/24  ISIS   15   20          D  1.1.1.2         GigabitEthernet1/0/0
        3.3.3.0/24  Direct 0    0           D  3.3.3.1         GigabitEthernet1/0/0
        3.3.3.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.0/8   Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
     172.16.1.0/24  ISIS   15   20          D  1.1.1.2      GigabitEthernet1/0/0
```

As shown in the routing table, the backup link Router A → Router C → Router B takes effect after the primary link fails, and the next-hop address of the route to 172.16.1.0/24 becomes 1.1.1.2.

# Run the **display isis bfd session all** command on Router A, and you can view that only the BFD session between Router A and Router C is Up.

```
[~RouterA] display isis bfd session all
                      BFD session information for ISIS(1)
                      -----------------------------------
Peer System ID : 0000.0000.0003        Interface : GE1/0/0
BFD State : up       Type : L2
Peer IP Address : 1.1.1.2
Local IP Address: 1.1.1.1
```

```
Total BFD session(s): 1
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
isis 1
 is-level level-2
 bfd all-interfaces enable
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.1 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 3.3.3.1 255.255.255.0
 isis enable 1
 isis cost 5
 isis bfd enable
 isis bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
bfd
#
isis 1
 is-level level-2
 bfd all-interfaces enable
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 2.2.2.2 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 3.3.3.2 255.255.255.0
 isis enable 1
 isis cost 5
 isis bfd enable
 isis bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
```

```
bfd
#
isis 1
 is-level level-2
 bfd all-interfaces enable
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 1.1.1.2 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 2.2.2.1 255.255.255.0
 isis enable 1
#
return
```

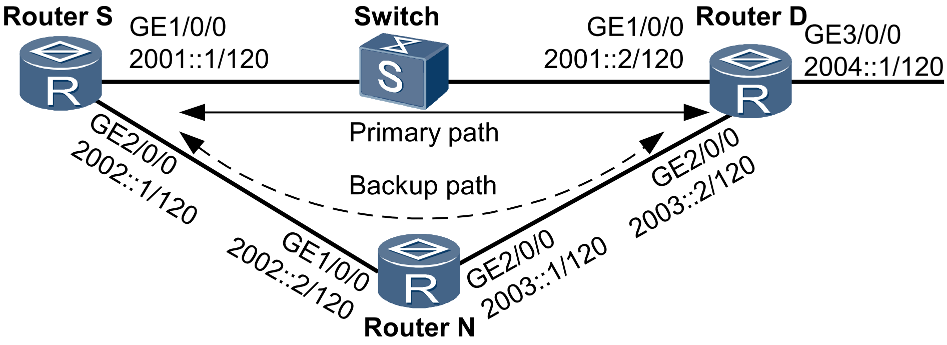
7.17.7 Example for Configuring Dynamic IPv6 BFD for IS-IS

This section provides an example for configuring dynamic IPv6 BFD to fast detect failures and trigger fast switchover of service traffic on IS-IS IPv6 networks.

Networking Requirements

Figure 7-14 shows an IS-IS IPv6 network. The primary path between Router S and Router D is Router S <---> Switch <---> Router D; the backup path between Router S and Router D is Router S <---> Router N <---> Router D.

Figure 7-14 Networking diagram for configuring dynamic IPv6 BFD for IS-IS



It is required to configure IPv6 BFD for IS-IS so that traffic between Router S and Router D can be rapidly switched to the backup path when the primary path or Switch fails.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic IS-IS IPv6 functions on each router to ensure that IPv6 routes are reachable.
2. Adjust the IS-IS cost on each router interface to allow the path of Router S <---> Switch <---> Router D to be the primary path and the path of Router S <---> Router N <---> Router D to be the backup path.

3. Enable BFD globally on each router.
4. Enable IPv6 BFD for IS-IS in the IS-IS view of each router.

Data Preparation

To complete the configuration, you need the following data:

- IS-IS process ID
- IS-IS NET
- Level of each router
- IS-IS cost of each interface
- Numbers of interfaces enabled with IPv6 BFD for IS-IS
- Minimum interval at which IPv6 BFD packets are sent and received, and BFD local detection time multiplier

Procedure

Step 1 Enable IPv6 forwarding capabilities and configure IPv6 addresses for all interfaces.

# Use the configuration of Router S as an example. The configuration of any other router is similar and is omitted.

```
<HUAWEI> system-view
[~HUAWEI] sysname RouterS
[~RouterS] ipv6
[~RouterS] interface gigabitethernet 1/0/0
[~RouterS-GigabitEthernet1/0/0] ipv6 enable
[~RouterS-GigabitEthernet1/0/0] ipv6 address 2001::1/120
[~RouterS-GigabitEthernet1/0/0] commit
[~RouterS-GigabitEthernet1/0/0] quit
```

Step 2 Configure basic IS-IS IPv6 functions.

# Configure Router S.

```
[~RouterS] isis 10
[~RouterS-isis-10] is-level level-2
[~RouterS-isis-10] network-entity 10.0000.0000.0001.00
[~RouterS-isis-10] ipv6 enable
[~RouterS-isis-10] quit
[~RouterS] interface gigabitethernet 1/0/0
[~RouterS-GigabitEthernet1/0/0] isis ipv6 enable 10
[~RouterS-GigabitEthernet1/0/0] quit
[~RouterS] interface gigabitethernet 2/0/0
[~RouterS-GigabitEthernet2/0/0] isis ipv6 enable 10
[~RouterS-GigabitEthernet2/0/0] commit
[~RouterS-GigabitEthernet2/0/0] quit
```

# Configure Router N.

```
[~RouterN] isis 10
[~RouterN-isis-10] is-level level-2
[~RouterN-isis-10] network-entity 10.0000.0000.0002.00
[~RouterN-isis-10] ipv6 enable
[~RouterN-isis-10] quit
[~RouterN] interface gigabitethernet 1/0/0
[~RouterN-GigabitEthernet1/0/0] isis ipv6 enable 10
[~RouterN-GigabitEthernet1/0/0] quit
[~RouterN] interface gigabitethernet 2/0/0
[~RouterN-GigabitEthernet2/0/0] isis ipv6 enable 10
[~RouterN-GigabitEthernet2/0/0] commit
RouterN-GigabitEthernet2/0/0] quit
```

# Configure Router D.

```
[~RouterD] isis 10
[~RouterD-isis-10] is-level level-2
[~RouterD-isis-10] network-entity 10.0000.0000.0003.00
[~RouterD-isis-10] ipv6 enable
[~RouterD-isis-10] quit
[~RouterD] interface gigabitethernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] isis ipv6 enable 10
[~RouterD-GigabitEthernet1/0/0] quit
[~RouterD] interface gigabitethernet 2/0/0
[~RouterD-GigabitEthernet2/0/0] isis ipv6 enable 10
[~RouterD-GigabitEthernet2/0/0] commit
[~RouterD-GigabitEthernet2/0/0] quit
```

# After the preceding configurations, run the **display ipv6 routing-table** command. You can view that routers have learned IPv6 routes from each other.

Step 3 Configure the IS-IS cost for each interface.

# Configure Router S.

```
[~RouterS] interface gigabitethernet 1/0/0
[~RouterS-GigabitEthernet1/0/0] isis cost 1 level-2
[~RouterS-GigabitEthernet1/0/0] quit
[~RouterS] interface gigabitethernet 2/0/0
[~RouterS-GigabitEthernet2/0/0] isis cost 10 level-2
[~RouterS-GigabitEthernet2/0/0] commit
[~RouterS-GigabitEthernet2/0/0] quit
```

# Configure Router N.

```
[~RouterN] interface gigabitethernet 1/0/0
[~RouterN-GigabitEthernet1/0/0] isis cost 10 level-2
[~RouterN-GigabitEthernet1/0/0] quit
[~RouterN] interface gigabitethernet 2/0/0
[~RouterN-GigabitEthernet2/0/0] isis cost 10 level-2
[~RouterN-GigabitEthernet2/0/0] commit
[~RouterN-GigabitEthernet2/0/0] quit
```

# Configure Router D.

```
[~RouterD] interface gigabitethernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] isis cost 1 level-2
[~RouterD-GigabitEthernet1/0/0] quit
[~RouterD] interface gigabitethernet 2/0/0
[~RouterD-GigabitEthernet2/0/0] isis cost 10 level-2
[~RouterD-GigabitEthernet2/0/0] commit
[~RouterD-GigabitEthernet2/0/0] quit
```

Step 4 Configure IPv6 BFD for IS-IS.

# Enable IPv6 BFD for IS-IS globally on Router S, Router N, and Router D, set the minimum interval at which IPv6 BFD packets are sent and received to 150 ms, and use the default local detection time multiplier 3.

# Configure Router S.

```
[~RouterS] bfd
[~RouterS-bfd] quit
[~RouterS] isis 10
[~RouterS-isis-10] ipv6 bfd all-interfaces enable
[~RouterS-isis-10] ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
[~RouterS-isis-10] commit
[~RouterS-isis-10] quit
```

# Configure Router N.

```
[~RouterN] bfd
[~RouterN-bfd] quit
[~RouterN] isis 10
[~RouterN-isis-10] ipv6 bfd all-interfaces enable
[~RouterN-isis-10] ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
[~RouterN-isis-10] commit
[~RouterN-isis-10] quit
```

# Configure Router D.

```
[~RouterD] bfd
[~RouterD-bfd] quit
[~RouterD] isis 10
[~RouterD-isis-10] ipv6 bfd all-interfaces enable
[~RouterD-isis-10] ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
[~RouterD-isis-10] commit
[~RouterD-isis-10] quit
```

# After the preceding configurations are complete, run the **display isis ipv6 bfd session all** command on Router S or Router D. You can view that the IPv6 BFD parameters take effect. Use the output of Router S as an example.

```
[~RouterS] display isis ipv6 bfd 10 session all
                    IPv6 BFD session information for ISIS(10)
                    ----------------------------------------
Peer System ID : 0000.0000.0003        Interface : GE1/0/0
IPv6 BFD State : up   Type : L2
Peer IPv6 Address : FE80::E0:2F47:B107:1
Local IPv6 Address: FE80::E0:2F47:B103:1

Peer System ID : 0000.0000.0002        Interface : GE2/0/0
IPv6 BFD State : up   Type : L2
Peer IPv6 Address : FE80::C964:0:B8B6:1
Local IPv6 Address: FE80::C964:0:B203:1

 Total BFD session(s): 2
```

Step 5 Verify the configuration

# Run the **display ipv6 routing-table 2004::1 120** command on Router S to view the IPv6 routing table. The next hop address is FE80::E0:2F47:B107:1; the outbound interface is GE 1/0/0.

```
[~RouterS] display ipv6 routing-table 2004::1 120
Routing Table : public
Summary Count : 1

Destination  : 2004::                                  PrefixLength : 120
NextHop      : FE80::E0:2F47:B107:1                    Preference   : 15
Cost         : 11                                      Protocol     : ISIS
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : GigabitEthernet1/0/0                    Flags        : D
```

# Run the **shutdown** command on GE 1/0/0 of Router D to simulate a fault in the primary link.

```
[~RouterD] interface gigabitethernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] shutdown
```

# Run the **display ipv6 routing-table2004::1 120** command on Router S to view the IPv6 routing table.

```
[~RouterS] display ipv6 routing-table 2004::1 120
Routing Table : public
Summary Count : 1

Destination  : 2004::                                  PrefixLength : 120
NextHop      : FE80::C964:0:B8B6:1                     Preference   : 15
Cost         : 20                                      Protocol     : ISIS
```

```
RelayNextHop : ::                                      TunnelID     : 0x0
Interface    : GigabitEthernet2/0/0                    Flags        : D
```

According to the routing table, after the primary link fails, the backup link takes effect. The next hop address of the route to 2004::/120 changes to FE80::C964:0:B8B6:1; the outbound interface changes to GE 2/0/0; the route cost may also be changed.

# Run the **display isis ipv6 bfd session all** command on Router S. You can see that only one BFD session is Up between Router S and Router N.

```
[~RouterS] display isis ipv6 bfd 10 session all
                    IPv6 BFD session information for ISIS(10)
                    ----------------------------------------
Peer System ID : 0000.0000.0002        Interface : GE2/0/0
IPv6 BFD State : up   Type : L2
Peer IPv6 Address : FE80::C964:0:B8B6:1
Local IPv6 Address: FE80::C964:0:B203:1

 Total BFD session(s): 1
```

----End

Configuration Files

- Configuration file of Router S

```
#
sysname RouterS
#
bfd
#
isis 10
 is-level level-2
 ipv6 enable topology standard
 ipv6 bfd all-interfaces enable
 ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001::1/120
 isis ipv6 enable 10
 isis cost 1 level-2
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2002::1/120
 isis ipv6 enable 10
 isis cost 10 level-2
#
return
```

- Configuration file of Router N

```
#
sysname RouterN
#
bfd
#
isis 10
 is-level level-2
 ipv6 enable topology standard
 ipv6 bfd all-interfaces enable
 ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
 network-entity 10.0000.0000.0002.00
#
interface GigabitEthernet1/0/0
```

```
 undo shutdown
 ipv6 enable
 ipv6 address 2002::2/120
 isis ipv6 enable 10
 isis cost 10 level-2
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2003::1/120
 isis ipv6 enable 10
 isis cost 10 level-2
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
bfd
#
isis 10
 is-level level-2
 ipv6 enable topology standard
 ipv6 bfd all-interfaces enable
 ipv6 bfd all-interfaces min-tx-interval 150 min-rx-interval 150
 network-entity 10.0000.0000.0003.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001::2/120
 isis ipv6 enable 10
 isis cost 1 level-2
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2003::2/120
 isis ipv6 enable 10
 isis cost 10 level-2
#
return
```

- Configuration file of the Switch

  The detailed configurations are not mentioned here.

7.17.8 Example for Configuring Basic IS-IS IPv6 Functions

This section describes how to configure basic IS-IS IPv6 functions, including enabling IPv6 globally, configuring an IPv6 address and enabling IPv6 for each interface, and configuring basic IS-IS functions and enabling IPv6.

Networking Requirements

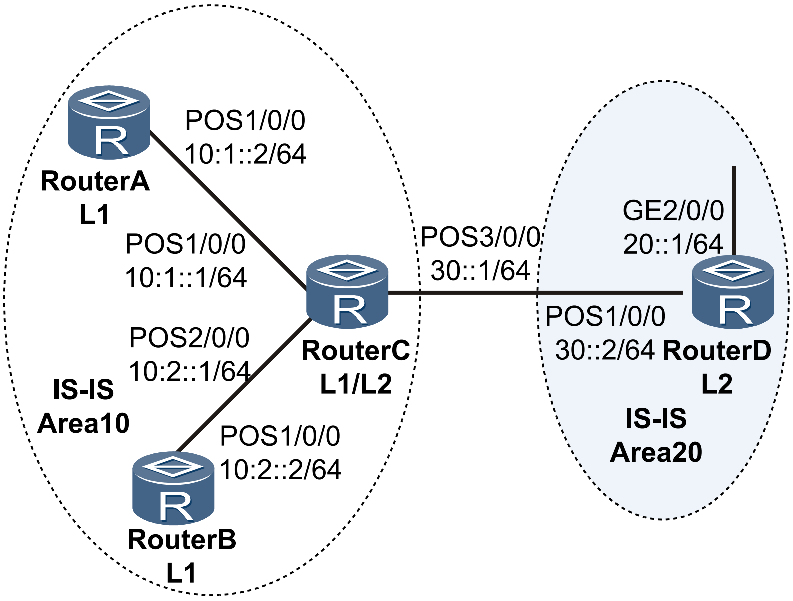
CAUTION

On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

As shown in **Figure 7-15**:

- Router A, Router B, Router C, and Router D belong to the same AS. It is required that IS-IS run on them to implement IPv6 interworking.
- Router A, Router B, and Router C belong to Area 10, and Router D belongs to Area 20.
- Router A and Router B are Level-1 devices; Router C is a Level-1-2 device; Router D is a Level-2 device.

Figure 7-15 Networking diagram for configuring basic IS-IS IPv6 functions



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable the IPv6 forwarding capability on each router, and configure an IPv6 address for each interface.
2. Enable IS-IS, configure the level, and specify the NET on each router.

Data Preparation

To complete the configuration, you need the following data:

- IPv6 addresses of the interfaces on Router A, Router B, Router C, and Router D
- AS numbers of Router A, Router B, Router C, and Router D
- Levels of Router A, Router B, Router C, and Router D

Procedure

Step 1 Enable the IPv6 forwarding capability, and configure an IPv6 address for each interface. Take the display on Router A as an example. The configurations of the other three routers are the same as that of Router A, and are not mentioned here.

```
<HUAWEI> system-view
[~HUAWEI] sysname RouterA
[~HUAWEI] commit
[~RouterA] interface pos 1/0/0
[~RouterA-Pos1/0/0] ipv6 enable
[~RouterA-Pos1/0/0] ipv6 address 10:1::2 64
[~RouterA-Pos1/0/0] commit
```

Step 2 Configure IS-IS.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] is-level level-1
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] ipv6 enable
[~RouterA-isis-1] quit
[~RouterA] interface pos 1/0/0
[~RouterA-Pos1/0/0] isis ipv6 enable 1
[~RouterA-Pos1/0/0] commit
[~RouterA-Pos1/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
[~RouterB-isis-1] is-level level-1
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] ipv6 enable
[~RouterB-isis-1] quit
[~RouterB] interface pos 1/0/0
[~RouterB-Pos1/0/0] isis ipv6 enable 1
[~RouterB-Pos1/0/0] commit
[~RouterB-Pos1/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis 1
[~RouterC-isis-1] network-entity 10.0000.0000.0003.00
[~RouterC-isis-1] ipv6 enable
[~RouterC-isis-1] quit
[~RouterC] interface pos 1/0/0
[~RouterC-Pos1/0/0] isis ipv6 enable 1
[~RouterC-Pos1/0/0] quit
[~RouterC] interface pos 2/0/0
[~RouterC-Pos2/0/0] isis ipv6 enable 1
[~RouterC-Pos2/0/0] quit
[~RouterC] interface pos 3/0/0
[~RouterC-Pos3/0/0] isis ipv6 enable 1
[~RouterC-Pos3/0/0] isis circuit-level level-2
[~RouterC-Pos3/0/0] commit
[~RouterC-Pos3/0/0] quit
```

# Configure Router D.

```
[~RouterD] isis 1
[~RouterD-isis-1] is-level level-2
[~RouterD-isis-1] network-entity 20.0000.0000.0004.00
[~RouterD-isis-1] ipv6 enable
[~RouterD-isis-1] quit
[~RouterD] interface pos 1/0/0
[~RouterD-Pos1/0/0] isis ipv6 enable 1
[~RouterD-Pos1/0/0] quit
[~RouterD] interface gigabitethernet 2/0/0
[~RouterD-GigabitEthernet2/0/0] isis ipv6 enable 1
[~RouterD-GigabitEthernet2/0/0] commit
[~RouterD-GigabitEthernet2/0/0] quit
```

Step 3 Verify the configuration.

# View the IS-IS routing table of Router A. You can view that Router A has the routes to each network segement of the Level-1 area.

```
[~RouterA] display isis route
                         Route information for ISIS(1)
                         -----------------------------
                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------
IPV6 Dest.     ExitInterface     NextHop                  Cost         Flags
----------------------------------------------------------------------------
 ::/0          Pos1/0/0          FE80::A83E:0:3ED2:1      10           A/-/-
 10:1::/64     Pos1/0/0          Direct                   10           D/L/-
 10:2::/64     Pos1/0/0          FE80::A83E:0:3ED2:1      20           A/-/-
     Flags: D-Direct, A-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

# View detailed information about IS-IS neighbors on Router C.

```
[~RouterC] display isis peer verbose
                          Peer information for ISIS(1)
                          ----------------------------
  System Id    Interface           Circuit Id       State HoldTime Type     PRI
0000.0000.0001 Pos1/0/0            0000000001        Up   24s      L1       --
  MT IDs supported     : 0(UP)
  Local MT IDs         : 0
  Area Address(es)     : 10
  Peer IPv6 Address(es): FE80::996B:0:9419:1
  Uptime               : 00:44:43
  Adj Protocol         : IPV6
0000.0000.0002 Pos2/0/0            0000000001        Up   28s      L1       --
  MT IDs supported     : 0(UP)
  Local MT IDs         : 0
  Area Address(es)     : 10
  Peer IPv6 Address(es): FE80::DC40:0:47A9:1
  Uptime               : 00:46:13
  Adj Protocol         : IPV6
0000.0000.0004 Pos3/0/0            0000000001        Up   24s      L2       --
  MT IDs supported     : 0(UP)
  Local MT IDs         : 0
  Area Address(es)     : 20
  Peer IPv6 Address(es): FE80::F81D:0:1E24:2
  Uptime               : 00:53:18
  Adj Protocol         :  IPV6
Total Peer(s): 3
```

# View detailed information about the IS-IS LSDB of Router C.

```
[~RouterC] display isis lsdb verbose
                        Database information for ISIS(1)
                        --------------------------------
                          Level-1 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
0000.0000.0001.00-00  0x0000000c   0x4e06        1117          113     0/0/0
 SOURCE       0000.0000.0001.00
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR V6 10:1::2
 IPV6         10:1::/64                     COST: 10
0000.0000.0002.00-00  0x00000009   0x738c        1022          83      0/0/0
 SOURCE       0000.0000.0002.00
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR V6 10:2::2
 IPV6         10:2::/64                     COST: 10
0000.0000.0003.00-00* 0x00000020   0x6b10        771           140     1/0/0
 SOURCE       0000.0000.0003.00
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR V6 30::1
 INTF ADDR V6 10:2::1
 INTF ADDR V6 10:1::1
 IPV6         10:2::/64                       COST: 10
 IPV6         10:1::/64                       COST: 10
                          Level-2 Link State Database
-------------------------------------------------------------------------------
LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
0000.0000.0003.00-00* 0x00000017   0x61b4        771           157     0/0/0
 SOURCE       0000.0000.0003.00
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR V6 30::1
 INTF ADDR V6 10:2::1
 INTF ADDR V6 10:1::1
 IPV6         30::/64                         COST: 10
 IPV6         10:2::/64                       COST: 10
 IPV6         10:1::/64                       COST: 10
0000.0000.0004.00-00  0x0000000b   0x6dfa        1024          124     0/0/0
 SOURCE       0000.0000.0004.00
 NLPID        IPV6
 AREA ADDR    20
 INTF ADDR V6 30::2
 INTF ADDR V6 20::1
 IPV6         30::/64                         COST: 10
 IPV6         20::/64                         COST: 10
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 is-level level-1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0001.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10:1::2/64
 isis ipv6 enable 1
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 is-level level-1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0002.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10:2::2/64
 isis ipv6 enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
```

```
#
isis 1
 ipv6 enable topology standard
 network-entity 10.0000.0000.0003.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10:1::1/64
 isis ipv6 enable 1
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10:2::1/64
 isis ipv6 enable 1
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 30::1/64
 isis ipv6 enable 1
 isis circuit-level level-2
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
isis 1
 is-level level-2
 ipv6 enable topology standard
 network-entity 20.0000.0000.0004.00
#
interface GigabitEthernet2/0/0
 ipv6 enable
 ipv6 address 20::1/64
 isis ipv6 enable 1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 30::2/64
 isis ipv6 enable 1
#
return
```

7.17.9 Example for Configuring IS-IS Auto FRR

Networking Requirements

⚠ **CAUTION**

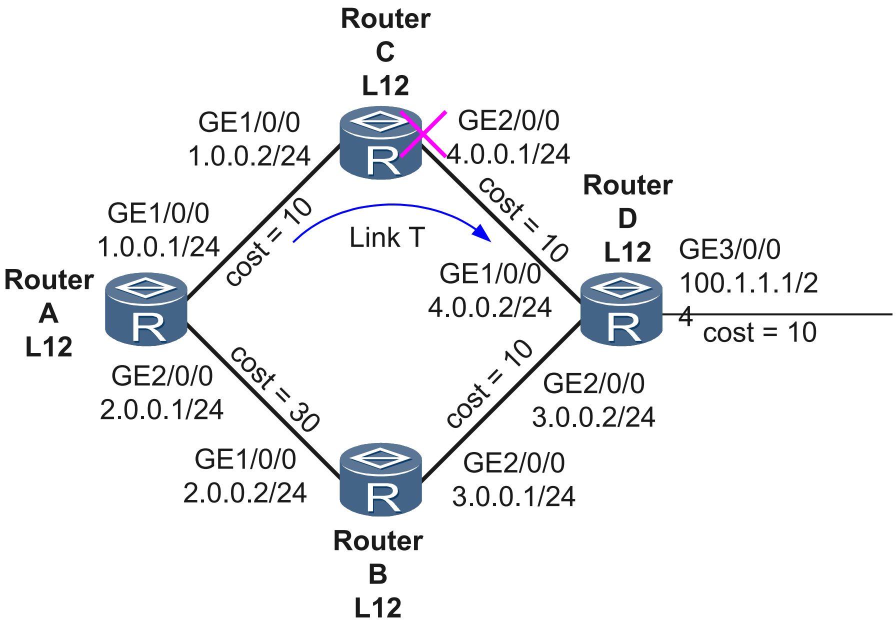
On a single NE5000E, an interface is numbered in the format of slot number/card number/interface number. On the NE5000E cluster, an interface is numbered in the format of chassis ID/slot number/card number/interface number. If the slot number is specified, the chassis ID of the slot must also be specified.

When a fault occurs on a network, IS-IS Auto FRR fast switches traffic to a backup link before the route convergence. This prevents traffic interruption.

In **Figure 7-16**:

- IS-IS runs between four routers.
- The four routers are all Level-1-2 routers.
- If Router C or Link T fails, it is required that the traffic forwarded by Router A is rapidly switched to the backup link.

Figure 7-16 Networking diagram of configuring IS-IS Auto FRR



Configuration Roadmap

The configuration roadmap is as follows:

1. Enable basic IS-IS functions on each router.
2. Set a larger link cost (in compliance with the traffic protection inequality of IS-IS Auto FRR) on GE 2/0/0 of Router A, and ensure that Link T is preferentially selected.
3. Enable IS-IS Auto FRR on Router A that forwards the protected traffic.

Data Preparation

To complete the configuration, you need the following data:

- IP addresses of interfaces on each router
- NET of each router
- Level of each router
- Costs of interfaces on each router

Procedure

Step 1 Configure IP addresses for interfaces. The details are omitted.

Step 2 Configure basic IS-IS functions.

# Configure Router A.

```
[~RouterA] isis 1
[~RouterA-isis-1] is-level level-1-2
[~RouterA-isis-1] network-entity 10.0000.0000.0001.00
[~RouterA-isis-1] quit
[~RouterA] interface gigabitethernet 1/0/0
[~RouterA-GigabitEthernet1/0/0] isis enable 1
[~RouterA-GigabitEthernet1/0/0] quit
[~RouterA] interface gigabitethernet 2/0/0
[~RouterA-GigabitEthernet2/0/0] isis enable 1
[~RouterA-GigabitEthernet2/0/0] commit
[~RouterA-GigabitEthernet2/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis 1
[~RouterB-isis-1] is-level level-1-2
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface gigabitethernet 1/0/0
[~RouterB-GigabitEthernet1/0/0] isis enable 1
[~RouterB-GigabitEthernet1/0/0] quit
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] isis enable 1
[~RouterB-GigabitEthernet2/0/0] commit
[~RouterB-GigabitEthernet2/0/0] quit
```

# Configure Router C.

```
[~RouterC] isis 1
[~RouterC-isis-1] is-level level-1-2
[~RouterC-isis-1] network-entity 10.0000.0000.0003.00
[~RouterC-isis-1] quit
[~RouterC] interface gigabitethernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] isis enable 1
[~RouterC-GigabitEthernet1/0/0] quit
[~RouterC] interface gigabitethernet 2/0/0
[~RouterC-GigabitEthernet2/0/0] isis enable 1
[~RouterC-GigabitEthernet2/0/0] commit
[~RouterC-GigabitEthernet2/0/0] quit
```

# Configure Router D.

```
[~RouterD] isis 1
[~RouterD-isis-1] is-level level-1-2
[~RouterD-isis-1] network-entity 10.0000.0000.0004.00
[~RouterD-isis-1] quit
[~RouterD] interface gigabitethernet 1/0/0
[~RouterD-GigabitEthernet1/0/0] isis enable 1
[~RouterD-GigabitEthernet1/0/0] quit
[~RouterD] interface gigabitethernet 2/0/0
[~RouterD-GigabitEthernet2/0/0] isis enable 1
[~RouterD-GigabitEthernet2/0/0] commit
[~RouterD-GigabitEthernet2/0/0] quit
```

Step 3 Set the cost of Gigabit Ethernet 2/0/0 on RouterA to 30, and then check routing information.

# Configure the cost of GE 2/0/0 on Router A to 30.

```
[~RouterA] interface gigabitethernet 2/0/0
[~RouterA-GigabitEthernet2/0/0] isis cost 30
[~RouterA-GigabitEthernet2/0/0] commit
```

```
[~RouterA-GigabitEthernet2/0/0] quit
```

# Check information about the link from Router A to Router D. Link T has a lower cost, and thereby IS-IS optimally selects Link T to send traffic that is forwarded by Router A.

```
<RouterA> display isis route verbose

                         Route information for ISIS(1)
                         -----------------------------

                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 1            Flags     : A/-/L/-/-
 Priority   : Medium
 NextHop    :                    Interface :              ExitIndex :
    1.0.0.2                            GE1/0/0                  0x00000003

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set


                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 3            Flags     : -/-/-/-/-

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set
```

Step 4 Enable IS-IS Auto FRR on Router A.

Step 5 Verify the configuration.

# Enable IS-IS Auto FRR on Router A.

```
<RouterA> isis
[~RouterA-isis-1] frr
[~RouterA-isis-1-frr] loop-free-alternate
[~RouterA-isis-1-frr] commit
```

# Check information about the link from Router A to Router D. You can find that IS-IS creates a backup link because IS-IS Auto FRR is enabled.

```
<RouterA> display isis route verbose

                         Route information for ISIS(1)
                         -----------------------------

                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 1            Flags     : A/-/L/-/-
 Priority   : Medium
 NextHop    :                    Interface :              ExitIndex :
    1.0.0.2                            GE1/0/0                  0x00000003
 (B)2.0.0.2                          GE2/0/0                  0x00000004

     Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set


                        ISIS(1) Level-2 Forwarding Table
                        --------------------------------

 IPV4 Dest  : 100.1.1.0/24       Int. Cost : 30           Ext. Cost : NULL
 Admin Tag  : -                  Src Count : 3            Flags     : -/-/-/-/-
```

```
Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, U-Up/Down Bit Set
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
isis 1
 network-entity 10.0000.0000.0001.00
 frr
  loop-free-alternate level-1
  loop-free-alternate level-2
#
interface gigabitethernet 1/0/0
 undo shutdown
 ip address 1.0.0.1 255.255.255.0
 isis enable 1
#
interface gigabitethernet 2/0/0
 undo shutdown
 ip address 2.0.0.1 255.255.255.0
 isis enable 1
 isis cost 30
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
isis 1
 network-entity 10.0000.0000.0002.00
#
interface gigabitethernet 1/0/0
 undo shutdown
 ip address 2.0.0.2 255.255.255.0
 isis enable 1
#
interface gigabitethernet 2/0/0
 undo shutdown
 ip address 3.0.0.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
 network-entity 10.0000.0000.0003.00
#
interface gigabitethernet 1/0/0
 undo shutdown
 ip address 1.0.0.2 255.255.255.0
 isis enable 1
#
interface gigabitethernet 2/0/0
 undo shutdown
 ip address 4.0.0.1 255.255.255.0
 isis enable 1
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
isis 1
 network-entity 10.0000.0000.0004.00
#
interface gigabitethernet 1/0/0
 undo shutdown
 ip address 4.0.0.2 255.255.255.0
 isis enable 1
#
interface gigabitethernet 2/0/0
 undo shutdown
 ip address 3.0.0.2 255.255.255.0
 isis enable 1
#
interface gigabitethernet 3/0/0
 undo shutdown
 ip address 100.1.1.1 255.255.255.0
 isis enable 1
#
return
```

8 BGP Configuration

About This Chapter

BGP, which is applicable to a large-scale network with a complicated structure, is used between ASs to transmit routing information.

8.10 Configuring Path MTU Auto Discovery
By configuring path MTU auto discovery, you can discover the minimum MTU on the network path from the source to the destination.

8.11 Configuring BGP Load Balancing
By configuring BGP load balancing, you can properly use network resources.

8.12 Configuring BGP Route Dampening
By configuring BGP route dampening, you can suppress unstable BGP routes.

8.13 Setting Parameters for a BGP Peer Connection
By setting parameters for a BGP peer connection, you can adjust and optimize the BGP network performance.

8.14 Configuring a BGP Route Reflector
By configuring a BGP route reflector, you can solve the problem of establishing full-mesh connections between multiple IBGP peers.

8.15 Configuring BFD for BGP
BFD for BGP provides a fast fault detection mechanism for BGP, and thus speeds up network convergence.

8.16 Configuring BGP Auto FRR
BGP Auto FRR is applicable to the network topology with both active and standby links as a protection measure against faults over the link. BGP Auto FRR is suitable for the services that are sensitive to packet loss and delay.

8.17 Configuring the BGP GR Helper
You can configure a device to function as a GR Helper to help a neighbor with the BGP GR process.

8.18 Improving the Security of a BGP Network
To improve the security of a BGP network, you can configure BGP authentication and GTSM on the BGP network.

8.19 Configuring BGP Extensions
Configuring BGP extensions enables BGP to provide routing information for multiple routing protocols.

8.20 Maintaining BGP
Maintaining BGP involves resetting BGP connections, clearing BGP statistics, and debugging BGP.

8.21 Configuration Examples
BGP configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

8.1 BGP Overview

BGP is a dynamic routing protocol used between ASs to control route advertisement and select optimal routes.

The Border Gateway Protocol (BGP) is a dynamic routing protocol used between Autonomous Systems (ASs). The three earlier versions of BGP are BGP-1 (defined in RFC 1105), BGP-2 (defined in RFC 1163), and BGP-3 (defined in RFC 1267). The current version of BGP is BGP-4 (defined in RFC 4271).

As an exterior routing protocol on the Internet, BGP is widely used among Internet Service Providers (ISPs).

NOTE

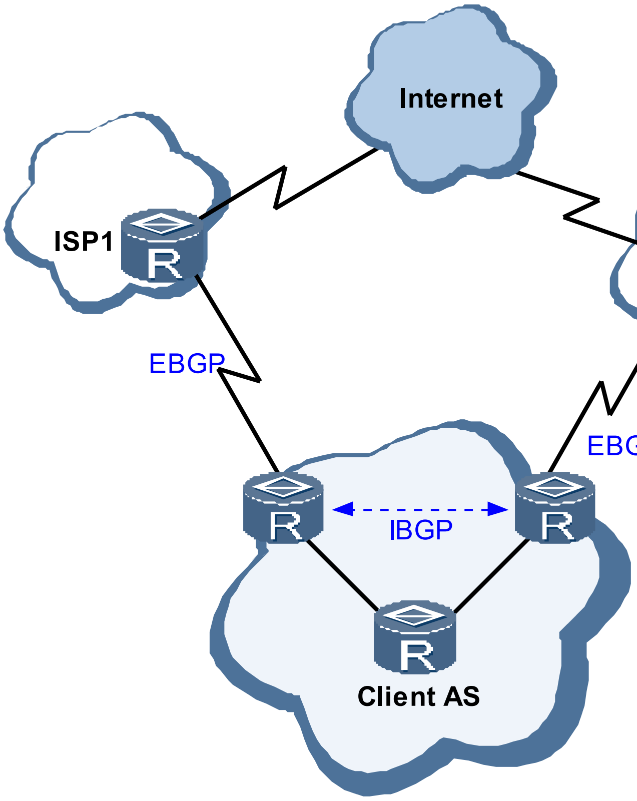
BGP stated in this manual refers to BGP-4, unless otherwise stated.

BGP has the following characteristics:

- Different from an Internal Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) and Routing Information Protocol (RIP), BGP is an Exterior Gateway Protocol (EGP), and is used to control route advertisement and select optimal routes between ASs rather than discover and calculate routes.
- BGP uses the Transport Control Protocol (TCP) with the port number being 179 as the transport layer protocol. The reliability of BGP is thus enhanced.
- BGP supports Classless Inter-Domain Routing (CIDR).
- When routes are being updated, BGP transmits only the updated routes. This reduces the bandwidths occupied by BGP for route distribution. Therefore, BGP is applicable to the Internet where a large number of routes are transmitted.
- BGP eliminates routing loops by adding the AS_Path to BGP routes.
- BGP provides rich routing policies to flexibly select and filter routes.
- BGP can be easily extended to adapt to the development of networks.

As shown in **Figure 8-1**, BGP is called Internal BGP (IBGP) when it runs within an AS, and is called External BGP (EBGP) when it runs among ASs.

Figure 8-1 Basic networking diagram of BGP



8.2 BGP Features Supported by the NE5000E

The BGP features supported by the NE5000E include load balancing, timers, suppression of route flapping, route aggregation, peer group, community, route reflector, four-byte AS number, MP-BGP, address family, BFD for BGP, BGP Auto FRR, BGP GR helper, ISSU, BGP NSR, MD5 authentication, keychain authentication, and GTSM.

Policies for BGP Route Selection

In the NE5000E, when multiple routes to the same destination are available, BGP selects routes based on the following rules:

1. Prefers the route with the highest PreVal.

   PrefVal is a Huawei-specific parameter. It is valid only on the device where it is configured.

2. Prefers the route with the highest Local_Pref.

   A route without Local_Pref is considered to have had the value set by using the **default local-preference** command or to have a value of 100 by default.

3. Prefers a locally originated route. A locally originated route takes precedence over a route learned from a peer.

   Locally originated routes include routes imported by using the **network** command or the **import-route** command, manually aggregated routes, and automatically summarized routes.

   (1) A summarized route is preferred. A summarized route takes precedence over a non-summarized route.

    (2) A route obtained by using the **aggregate** command is preferred over a route obtained by using the **summary automatic** command.

    (3) A route imported by using the **network** command is preferred over a route imported by using the **import-route** command.

4. Prefers the route with the shortest AS_Path.
   - The AS_CONFED_SEQUENCE and AS_CONFED_SET are not included in the AS_Path length.
   - An AS_SET counts as 1, no matter how many ASs are in the set.
   - After the **bestroute as-path-ignore** command is run, the AS_Path attributes of routes are not compared in the route selection process.
5. Prefers the route with the highest Origin type. IGP is higher than EGP, and EGP is higher than Incomplete.
6. Prefers the route with the lowest Multi Exit Discriminator (MED).
   - The MEDs of only routes from the same AS are compared. MEDs of two routes are compared only when the first AS number in the AS_SEQUENCE (excluding AS_CONFED_SEQUENCE) is the same for the two routes.
   - A route without any MED is assigned a MED of 0, unless the **bestroute med-none-as-maximum** command is run. If the **bestroute med-none-as-maximum** command is run, the route is assigned the highest MED of 4294967295.
   - After **compare-different-as-med** command is run, the MEDs in routes sent from peers in different ASs are compared. Do not use this command unless it is confirmed that different ASs use the same IGP and route selection mode. Otherwise, a loop may occur.
   - After the **deterministic-med** command is run, routes are not selected in the sequence in which routes are received.
7. Prefers EBGP routes over IBGP routes.

   EBGP is higher than IBGP, IBGP is higher than LocalCross, and LocalCross is higher than RemoteCross.

   If the ERT of a VPNv4 route in the routing table of a VPN instance on a PE matches the IRT of another VPN instance on the PE, the VPNv4 route will be added to the routing table of the second VPN instance. This is called LocalCross. If the ERT of a VPNv4 route from a remote PE is learned by the local PE and matches the IRT of a VPN instance on the local PE, the VPNv4 route will be added to the routing table of that VPN instance. This is called RemoteCross.
8. Prefers the route with the lowest IGP metric to the BGP next hop.

   After the **bestroute igp-metric-ignore** command is run, the IGP metrics are not compared for routes during route selection.

   NOTE

   Assume that load balancing is configured. If the preceding rules are the same and there are multiple external routes with the same AS_Path, load balancing will be performed based on the number of configured routes.
9. Prefers the route with the shortest Cluster_List.
10. Prefers the route advertised by the router with the smallest router ID.

    After the **bestroute router-id-ignore** command is run, router IDs of the devices are not compared in the route selection process.

NOTE

If routes carry the Originator_ID, the originator ID is substituted for the router ID during route selection. The route with the smallest Originator_ID is preferred.

11. Prefers the route learned from the peer with the smallest address if the IP addresses of peers are compared in the route selection process.

Policies for BGP Route Advertisement

In the implementation of the NE5000E, BGP advertises routes according to the following principles:

- The BGP speaker advertises only the optimal route to its peer when there are multiple valid routes.
- The BGP speaker sends only the routes in use to its peer.
- The BGP speaker advertises the routes learned from EBGP peers to all BGP peers.
- The BGP speaker does not advertise the routes learned from an IBGP peer to other IBGP peers.
- The BGP speaker advertises the routes learned from IBGP peers to its EBGP peers when synchronization between BGP and an IGP is not enabled.
- The BGP speaker advertises all BGP routes to new peers when the connections with the new peers are established.

Route Selection Policies During Applications of BGP Load Balancing

In BGP, the next-hop address of a route may not be the address of the peer directly connected to the router. This is because the next hop does not change when IBGP peers advertise routes. In this case, to ensure that a packet is correctly forwarded, the router must find a reachable address, and then forwards the packet to the next hop according to the routing table. In this process, the route to the reachable address is called a dependent route. BGP forwards packets according to dependent routes. The process of finding a dependent route according to the next hop address is called route iteration.

The NE5000E supports BGP load balancing based on route iteration. That is, if load balancing is performed among dependent routes (assume that there are three next hop addresses), BGP generates the same number of next hop addresses to guide the forwarding of packets. The route iteration-based BGP load balancing does not need to be configured through commands. This feature is always enabled on the NE5000E.

BGP load balancing is different from that of an IGP in terms of implementation methods as follows:

- For different routes to the same destination address, an IGP calculates the metrics of routes according to its routing algorithm. Load balancing is performed among the routes with the same metric.
- BGP does not have a routing algorithm. Therefore, BGP cannot determine whether to perform load balancing among routes according to metrics. BGP, however, has many route attributes with different priorities in route selection policies. BGP performs load balancing according to route selection policies. That is, BGP load balancing is performed according to the maximum number of equal-cost routes only when all attributes of routes with the higher preference are the same.

NOTE

BGP load balancing is performed only among the routes with the same AS_Path attribute.

Synchronization Between IBGP and an IGP

The synchronization between IBGP and an IGP is performed to avoid misleading external AS routers.

If the synchronization function is configured, the router checks the IGP routing table before adding an IBGP route to the routing table and advertising it to EBGP peers. The IBGP route is added to the routing table and advertised to the EBGP peers only when the IGP also learns this IBGP route.

The synchronization function can be disabled in the following cases:

- The local AS is not a transit AS.
- All routers in the local AS establish fully meshed connections.

Flexibility

BGP has many route attributes. By applying the routing policies defined according to these attributes, you can flexibly control the selection, receiving, and advertisement of routes. The common BGP route attributes are as follows:

- Origin
- AS_Path
- Next_Hop
- Multi-Exit-Discriminator (MED)
- Local_Pref
- Community

Stability

The stability of a routing protocol plays a critical role on a large- scale network. On a small-scale network such as the Internet, a large number of flapping routes seriously affect the normal running of the network.

- BGP timers

  By using various timers, BGP can minimize the impact of interfaces or routes frequently alternating between Up and Down.
- BGP route dampening

  BGP route dampening solves the problem of route instability. Generally, BGP is applied to complex networks where routes change frequently. Frequent route flapping consumes lots of bandwidths and CPU resources and even affects the normal running of networks. To avoid the impact of frequent route flapping, BGP adopts route dampening to suppress unstable routes.

Expansibility

The expansibility of BGP is measured according to the number of peers and the number of routes. The expansibility of BGP can reduce the number of routes to be maintained and the number of Update packets.

- Route Aggregation

  On a large-scale network, the BGP routing table is rather large. You can configure route aggregation to reduce the size of the routing table.

Route aggregation refers to the process of aggregating multiple routes into one route. After route aggregation is configured, BGP advertises only the aggregated route rather than all the specific routes to BGP peers.

- Peer group

  A peer group is a group of peers with the same routing policy. On a large-scale BGP network, there are a large number of peers and most of them have the same policy. You need to repeatedly use certain commands when configuring the policy for the peers. In most situations, you can simplify the configurations by using a peer group.

  In addition, adding peers to a group speeds up route advertisement.

- Community

  Peers in the same peer group share the same policy, whereas BGP routers configured with the same community attribute of multiple ASs share the same policy. The community is a route attribute, and is transmitted among BGP peers, regardless of whether the BGP peers belong to the same AS.

  Before advertising a route with the community attribute to peers, a BGP router can change the original community attribute of this route.

  Besides the well-known communities, you can use a community filter to filter self-defined extended community attributes to control routing policies flexibly.

- Route reflector

  To ensure the connectivity between IBGP peers, you need to establish fully meshed connections between IBGP peers. If there are n routers in an AS, n (n-1)/2 IBGP connections need to be established. When there are a large number of IBGP peers, network resources and CPU resources are greatly consumed.

  Route reflection can solve the problem. In an AS, one router functions as a Route Reflector (RR) and the other routers function as clients. The clients establish IBGP connections with the RR. The RR reflects routes among clients, and BGP connections do not need to be established between the clients.

  A BGP router that functions as neither the RR nor the client is called a non-client. A non-client must establish a fully meshed connection with the RR and all the other non-clients.

- 4-byte AS number

  Currently, 2-byte AS numbers used on the network range from 0 to 65535. Available AS numbers, however, become almost exhausted. Therefore, 2-byte AS numbers need to be extended to 4-byte AS numbers, which should also be compatible with the old speaker that supports only 2-byte AS numbers.

  The 4-byte AS number feature extends a 2-byte AS number to a 4- byte AS number, and negotiates the 4-byte AS number capability and transmits 4-byte AS numbers by defining a new capability code and new optional transitive attributes. This implements communication between new speakers that support 4-byte AS numbers, and between old speakers that support only 2-byte AS numbers and new speakers.

- Distributed BGP

  Distributed BGP enables BGP to run on multiple boards. Thus, you can perform capacity expansion by increasing the number of boards, which is a sustainable low-cost solution.

  Distributed BGP based on peers is implemented through the following instances:

  - CBGP

    Centralized BGP (CBGP) can implement all functions of BGP. In addition, CBGP can implement the centralized management function such as route management and Indirect ID (IID) management.

  - PD-BGP

    Peer Distributed-BGP (PD-BGP) divides the peers according to instances, and then maintains partial peers in a PD-BGP instance according to the manual configurations or automatic configurations.

- MP-BGP

  Conventional BGP-4 manages only the IPv4 routing information. The inter-AS transmission of packets of other network layer protocols (such as IPv6), however, is limited.

  To support multiple types of network layer protocols, the Internet Engineering Task Force (IETF) extends BGP-4 to Multiprotocol Extensions for BGP-4 (MP-BGP). The current MP-BGP standard is RFC 2858.

  MP-BGP is forward compatible. That is, the routers supporting BGP extension can communicate with the routers that do not support BGP extension.

- Address family

  BGP uses address families to distinguish different network layer protocols. For the values of address families, refer to RFC 3232 (Assigned Numbers). The NE5000E supports multiple MP-BGP extension applications, such as VPN extension and IPv6 extension, which are configured in the respective address family views.

  The BGP-related address families include BGP address families, BGP-IPv4 unicast address families, BGP-VPNv4 address families, and BGP-VPN instance address families.

  NOTE

  Most commands in the BGP extended address family view are the same as those in the BGP view. The commands used in the BGP extended address family view, however, are valid only in corresponding applications.

Reliability

BGP uses TCP as the transport layer protocol. The reliability of BGP is thus enhanced.

- BFD for BGP

  In the NE5000E, Bidirectional Forwarding Detection (BFD) can be used in IPv4 networks to fast detect link faults for BGP.

  BFD can fast detect faults on links between BGP peers and report the faults to BGP. The fast convergence of BGP routes is thus implemented.

- BGP Auto FRR

  As a protection measure against link faults, BGP Auto Fast Reroute (FRR) is applicable to networks with primary and backup links and services that are sensitive to packet loss and delay. With BGP Auto FRR, switching between two BGP peers or two next hops can be implemented at the sub-second level.

- BGP GR

  When BGP restarts, the peer relationship is re-established and traffic forwarding is interrupted. After Graceful Restart (GR) is enabled, traffic interruption can be avoided.

  NOTE

  By default, the system supports only the GR helper.

- BGP NSR

  Non-Stop Routing (NSR) is a technique that prevents a peer from sensing the fault on the control plane of a router that provides a slave control plane. With NSR, when the control plane of the router becomes faulty, the peer relationships set up through specific routing protocols, MPLS, and other protocols that carry services are not interrupted.

During the master/slave switchover, BGP NSR ensures the continuous forwarding at the forwarding plane and continuous advertisement of BGP routes. In this process, the peer relationships are not affected, with peers not knowing the switchover on the local router. This ensures uninterrupted transmission of BGP services.

- ISSU

  BGP supports In-Service Software Upgrade (ISSU). ISSU can shorten the service interruption period during the software upgrade, thus greatly improving the reliability of devices; it also minimizes the impact of upgrade failure on the system by means of the rollback mechanism.

Security

To enhance BGP security, you can configure BGP authentication and GTSM on the BGP network.

- BGP authentication includes:
  - MD5 authentication

    BGP uses TCP as the transport layer protocol. To enhance BGP security, you can perform Message Digest 5 (MD5) authentication when a TCP connection is established. MD5 authentication, however, does not authenticate BGP packets. Instead, MD5 authentication sets the MD5 authentication password for the TCP connection, and the authentication is performed by TCP. If the authentication fails, the TCP connection cannot be established.
- BGP GTSM

  The GTSM mechanism protects a router by checking whether the TTL value in an IP packet header is within a pre-defined range to enhance the system security.

QoS

BGP accounting propagates traffic indexes through BGP attributes to identify routes, and then accounts services.

- The transmitter of BGP routes can set the community, MED, and Local_Pref attributes for the BGP routes through routing policies.
- The receiver of BGP routes can set the BGP traffic index based on BGP route attributes.
- After BGP accounting is enabled on an interface, the interface generates a traffic index table to implement the statistics function. The statistics can be based on the destination addresses or source addresses of packets.

BGP accounting can collect the traffic of the local AS and inter-ASs, and also collect the incoming and outgoing traffic of an AS. BGP accounting is valid only when a router needs to search the forwarding table. For example, if BGP accounting is configured for outgoing traffic on an originating interface, BGP accounting is invalid.

NOTE

For the detailed configuration of BGP accounting, see the *HUAWEI NetEngine5000E Core Router Configuration Guide - QoS*.

8.3 Configuring Basic BGP Functions

Before building a BGP network, you must configure basic BGP functions.

Applicable Environment

Before using BGP to implement communication between ASs, you must configure basic BGP functions.

NOTE

The commands in the BGP-IPv4 unicast address family view can be run in the BGP view. These commands are described in the BGP-IPv4 unicast address family view in configuration files.

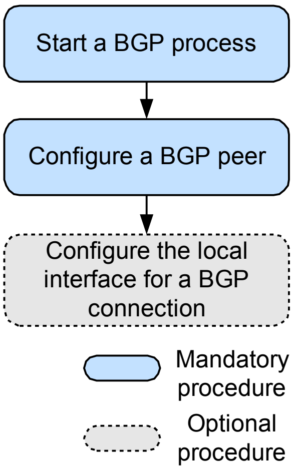
Pre-configuration Tasks

Before configuring basic BGP functions, complete the following task:

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

Configuration Procedures

Figure 8-2 Networking diagram of configuring basic BGP functions



8.3.1 Starting a BGP Process

Starting a BGP process is a prerequisite for configuring basic BGP functions. When starting a BGP process on a device, you need to specify the number of the AS that the device belongs to.

Context

CAUTION

Changing the router ID of a BGP peer causes the re-establishment of the BGP connection between routers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

A BGP process is enabled (the local AS number is specified) and the BGP view is displayed.

Step 3 (Optional) Run:

```
router-id ipv4-address
```

The ID of the BGP router is configured.

By default, BGP automatically selects the router ID in the system view. For the rules of selecting the router ID, see the *HUAWEI NetEngine5000E Core Router Command Reference*.

NOTE

If the ID of a router on a network is the IP address of a physical interface, route flapping occurs when the IP address of the physical interface changes. To enhance the stability of a network, it is recommended that you should configure the address of a loopback interface as the router ID manually.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

8.3.2 Configuring a BGP Peer

Devices can exchange BGP routing information only after BGP peers are configured and the BGP peer relationship is established.

Context

Because BGP uses TCP connections, you need to configure the IP addresses of peers when configuring BGP. A BGP peer may not be a neighboring node, but the BGP peer relationship can be created through logical links. To enhance the stability of BGP connections, you are recommended to establish connections by using loopback interface addresses.

The devices in the same AS establish IBGP peer relationships and the devices of different ASs establish EBGP peer relationships.

Procedure

- Configure an IBGP peer.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

The BGP view is displayed.

3. Run:

   ```
   peer ipv4-address as-number as-number
   ```

   The IP address of the peer and the number of the AS where the peer resides are specified.

   The number of the AS where the specified peer resides should be the same as that of the local AS.

   The IP address of the specified peer can be one of the following types:

   - IP address of an interface on a directly connected peer
   - Address of a loopback interface on a reachable peer
   - IP address of a sub-interface on a directly connected peer

   When the IP address of the specified peer is the address of a loopback interface, you need to complete the task of **Configure the Local Interface for a BGP Connection** to ensure the correct establishment of the peer.

4. (Optional) Run:

   ```
   peer ipv4-address description description-text
   ```

   The description of the peer is configured.

   You can simplify network management by configuring the descriptions of peers.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure an EBGP peer.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   peer ipv4-address as-number as-number
   ```

   The IP address of the peer and the number of the AS where the peer resides are specified.

   The number of the AS where the specified peer resides should be different from that of the local AS.

   The IP address of the specified peer can be one of the following types:

   - IP address of an interface on a directly connected peer
   - Address of a loopback interface on a reachable peer
   - IP address of a sub-interface on a directly connected peer

When the IP address of the specified peer is the address of a loopback interface, you need to complete the task of **Configure the Local Interface for a BGP Connection** to ensure the correct establishment of the peer.

4. (Optional)Run:

   ```
   peer ipv4-address ebgp-max-hop [ number ]
   ```

   The maximum number of hops for an EBGP connection is set.

   Generally, a directly connected physical link must be available between EBGP peers. If the requirement is not met, you must use the **peer ebgp-max-hop** command to configure EBGP peers to establish a TCP connection through multiple hops.

   NOTE

   > When an EBGP peer is established through a loopback interface, you must run the **peer ebgp-max-hop** (of which the value of *hop-count* is not smaller than 2) command. Otherwise, the peer fails to be established.

5. (Optional) Run:

   ```
   peer ipv4-address description description-text
   ```

   The description of the peer is configured.

   You can simplify network management by configuring the descriptions of peers.

6. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.3.3 (Optional) Configuring the Local Interface for a BGP Connection

To establish a BGP connection through loopback interfaces, you need to specify the interfaces for the BGP connection.

Context

To improve the reliability and stability of BGP connections, you can configure the local interface used for the BGP connection as the loopback interface. In this manner, if redundant links exist on the network, the BGP connection is not torn down when an interface or a link fails.To establish a BGP connection through loopback interfaces, you need to specify the interfaces for the BGP connection.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

peer *ipv4-address* **connect-interface** { *interface-type interface-number* | *ipv4-source-address* }

The source interface and source address are specified for the TCP connection for BGP.

By default, BGP uses the interface that is directly connected to the peer as the local interface of a TCP connection.

NOTE

When loopback interfaces are used to establish a BGP connection between two peers, it is recommended that you should run the **peer connect-interface** command on the two peers to ensure that the two peers are correctly connected. If you run the command only on one peer, the BGP connection may fail to be established.

If two devices need to set up multiple peer relationships through multiple direct links, run the **peer connect-interface** command on the two ends of each peer relationship to be set up to specify an interface name or an interface address used for setting up the peer relationship. This ensures correctly connections between the two devices.

Step 4 Run:

commit

The configuration is submitted.

----End

8.3.4 Checking the Configuration

After basic BGP functions are configured, you can check BGP peer information.

Prerequisite

All configurations of basic BGP functions are complete.

Procedure

- Run the **display bgp peer** [**verbose**] command to check the information about all BGP peers.
- Run the **display bgp peer** *ipv4-address* { **log-info** | **verbose** } command to check the information about a specified BGP peer.

----End

Example

Run the **display bgp peer** command, and you can view the status of the connection between BGP peers.

```
<HUAWEI> display bgp peer
 BGP local router ID : 2.2.2.2
 Local AS number : 65009
 Total number of peers : 3                 Peers in established state : 3
  Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  9.1.1.2         4 65009       49       62     0 00:44:58 Established       0
  9.1.3.2         4 65009       56       56     0 00:40:54 Established       0
  200.1.1.2       4 65008       49       65     0 00:44:03 Established       1
```

8.4 Configuring a BGP Peer Group

By configuring a BGP peer group, you can simplify the management of routing policies, and thus improve the efficiency of route advertisement.

Applicable Environment

A large number of peers exist on a large-scale BGP network, which is inconvenient for configuration and maintenance. In this case, you can configure peer groups to simplify the management and improve the efficiency of route advertisement.

Based on the ASs where peers reside, you can classify peer groups into IBGP peer groups and EBGP peer groups. You can classify EBGP peer groups into pure EBGP peer groups and mixed EBGP peer groups according to whether the peers reside in the same external AS.

Pre-configuration Tasks

Before configuring a BGP peer group, complete the following task:

- **Configuring Basic BGP Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
group group-name [ internal | external ]
```

A BGP peer group is created.

Step 4 Run:

```
peer ipv4-address group group-name
```

A peer is added to the peer group.

NOTE

You can add multiple peers to the peer group by repeating Step 4. The system creates a peer in the BGP view automatically, and sets its AS number to the AS number of the peer group.

If peers already exist in the peer group, you cannot change the AS number of the peer group, but can delete the specified AS number by using the **undo** command.

Step 5 Run:

```
peer group-name as-number as-number
```

The number of the AS where this peer group resides is set.

If a peer in the peer group has been configured with an AS number, the AS number of the peer is valid. Otherwise, the peer inherits the AS number configured for the peer group.

Step 6 (Optional) Run:

```
peer group-name connect-interface { interface-type interface-number | ipv4-source-address }
```

The source interface and source address are specified for the TCP connection for BGP.

By default, BGP uses the interface that is directly connected to the peer as the local interface of a TCP connection.

NOTE

When loopback interfaces are used to establish a BGP connection between two peers, it is recommended that you should run the **peer connect-interface** command on the two peers to ensure that the two peers are correctly connected. If you run the command only on one peer, the BGP connection may fail to be established.

Step 7 (Optional) Run:

```
peer group-name ebgp-max-hop [ number ]
```

The maximum number of hops for an EBGP connection is set.

If all peers in a peer group are EBGP peers, you need to run the **peer ebgp-max-hop** command to set the maximum number of hops for an EBGP connection.

NOTE

When a loopback interface is used to establish an EBGP peer, you must run the **peer ebgp-max-hop** (of which, the value of *hop-count* is not smaller than 2) command. Otherwise, the EBGP peer fails to be established.

Step 8 (Optional) Run:

```
peer group-name description description-text
```

The description of the peer group is configured.

You can simplify network management by configuring the descriptions of peers.

Step 9 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the configuration, you can run the following commands to check the configuration.

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to check the detailed information about a peer.
- Run the **display bgp group** [*group-name*] command to check the information about a BGP peer group.

Run the **display bgp group** [*group-name*], and you can view the information about a BGP peer group. For example:

```
<HUAWEI> display bgp group my-peer
 BGP peer-group: my-peer
 Remote AS 200
 Group's BFD has been enabled
 Type : internal
 Maximum allowed route limit: 150000
 Threshold: 75%
 Configured hold timer value: 180
 Keepalive timer value: 60
 Minimum route advertisement interval is 15 seconds
 listen-only has been configured
PeerSession Members:
   2.2.2.2
 Peer Preferred Value: 0
 No routing policy is configured
  Peer Members:
  Peer            V    AS   MsgRcvd  MsgSent  OutQ  Up/Down       State  PrefRcv
  2.2.2.2         4    200        0        0     0 00:00:47     Active       0
```

8.5 Controlling the Import of Routing Information

Importing routes from other routing protocols can enrich the BGP routing table. When importing IGP routes, BGP filters the routes according to routing protocols.

Applicable Environment

BGP itself cannot discover routes. Therefore, it needs to import routes from other protocols such as static route or OSPF to the BGP routing table. In this manner, these imported routes can be transmitted within an AS or between ASs. When importing routes, BGP filters the routes according to different routing protocols.

Pre-configuration Tasks

Before controlling the import of routing information, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except "Checking the Configuration") according to the applicable environment.

8.5.1 Configuring BGP to Import Routes

BGP can import the routes from other routing protocols. When BGP needs to import routes from a dynamic routing protocol, you need to specify the process ID of the protocol.

Context

BGP itself cannot discover routes. Therefore, it needs to import routes from other protocols such as an IGP or static route to the BGP routing table. In this manner, these imported routes can be transmitted within an AS or between ASs.

BGP can import routes in either Import or Network mode:

- In Import mode, BGP imports routes according to protocol types, for example, RIP routes, OSPF routes, Intermediate System to Intermediate System (IS-IS) routes, static routes, or direct routes, to the BGP routing table.

- The Network mode is more precise than the Import mode. In Network mode, routes with the specified prefix and mask are imported to the BGP routing table.

Procedure

- Import mode

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. (Optional) Run:

     ```
     ipv4-family unicast
     ```

     The BGP-IPv4 unicast address family view is displayed.

     By default, the system is configured in the IPv4 unicast address family view.

  4. Run:

     ```
     import-route protocol [ process-id ] [ med med | route-policy route-policy-name ] *
     ```

     BGP is configured to import routes from other routing protocols.

     NOTE

     When BGP needs to import routes from IS-IS, OSPF, or RIP, you need to specify the process ID of the protocol.

     If the **default-route imported** command is not used, BGP cannot import default routes when you run the **import-route** command to configure BGP to import routes from other protocols.

  5. Run:

     ```
     commit
     ```

     The configuration is committed.

- Network mode

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. (Optional) Run:

     ```
     ipv4-family unicast
     ```

     The BGP-IPv4 unicast address family view is displayed.

     By default, the system is configured in the IPv4 unicast address family view.

  4. Run:

     ```
     network ipv4-address [ mask | mask-length ] [ route-policy route-policy-name ]
     ```

BGP is configured to advertise local routes.

If the mask or mask length of an IPv4 address is not specified, the IPv4 address is processed as a classful address.

The local routes to be advertised must be in the local IP routing table. You can use routing policies to flexibly control the routes to be advertised.

NOTE

- The destination address and mask specified in the **network** command must be consistent with the corresponding entries in the local IP routing table. Otherwise, the specified route cannot be advertised.
- When running the **undo network** command to clear the existing configuration, you need to specify the correct mask.

5. Run:

```
commit
```

The configuration is committed.

----End

8.5.2 Configuring BGP to Import Default Routes

Only the default routes that exist in the local routing table can be imported.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
default-route imported
```

BGP is configured to import default routes.

To import a default route, you need to use the **default-route imported** command and the **import-route** command. This is because default routes cannot be imported only through the **import-route** command and the **default-route imported** command is used to import only default routes that exist in the local routing table.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.5.3 Checking the Configuration

After the import of routes is controlled, you can check the imported routes that match a specified filter.

Prerequisite

All configurations of controlling the import of routing information are complete.

Procedure

- Run the **display bgp routing-table** [*network*] [*mask* | *mask-length*] command to check BGP routes.
- Run the **display bgp routing-table as-path-filter** *as-path-filter-number* command to check the routes matching a specified AS_Path filter.
- Run the **display bgp routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check the routes matching a specified BGP community filter.
- Run the **display bgp routing-table cidr** command to check CIDR routes.

----End

Example

Run the **display big routing-table 60.0.0.35** command, and you can view specified BGP routes.

```
<HUAWEI> display bgp routing-table 60.0.0.35

 BGP local router ID : 30.0.0.35
 Local AS number : 400
 Paths:   1 available, 1 best, 1 select
 BGP routing table entry information of 60.0.0.35/32:
 Network route.
 From: 0.0.0.0 (0.0.0.0)
 Route Duration: 3d04h00m12s
 Direct Out-interface: InLoopBack0
 Original nexthop: 127.0.0.1
 Qos information : 0x0
 AS-path Nil, origin igp, MED 0, pref-val 0, valid, local, best, select, pre 0
 Not advertised to any peer yet
```

8.6 Controlling the Advertisement of Routing Information

BGP can filter the routes to be advertised by a peer or apply routing policies to the routes to be advertised by a peer.

Applicable Environment

To ensure that devices can exchange routing information, BGP needs to advertise its routing information to peers according to network planning. BGP can filter or perform routing policies for the routes advertised by a certain peer or a peer group.

Pre-configuration Tasks

Before controlling the advertisement of routing information, complete the following task:

- Configuring Basic BGP Functions

Configuration Procedures

You can choose to perform the following configuration tasks (except "Checking the Configuration") according to the applicable environment.

8.6.1 Configuring BGP Route Aggregation

By configuring route aggregation, you can reduce the size of the routing table of a peer. BGP supports automatic aggregation and manual aggregation.

Context

On a large-scale BGP network, when a router advertises routing information to its peers, you can configure route aggregation on the router. This can reduce the number of prefixes to be advertised and enhance the stability of BGP.

BGP supports automatic aggregation and manual aggregation.

Procedure

- Configure automatic aggregation.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     summary automatic
     ```
     Subnet routes are aggregated according to natural network segments.

     The command is used to aggregate the routes imported into BGP. These routes can be direct routes, static routes, RIP routes, OSPF routes, or IS-IS routes. The command, however, is invalid for the routes imported through the **network** command.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.
- Configure manual aggregation.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```

The BGP view is displayed.

3. Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

4. Configure automatic aggregation according to the actual networking.

   - To advertise both the aggregated route and specific routes, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } command.
   - To advertise only the aggregated route, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **detail-suppressed** command.
   - To advertise a specific route, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **suppress-policy** *route-policy-name* command.

     To advertise a specific route, you can also run the **peer route-policy** command.
   - To generate an aggregated route used to detect a loop, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **as-set** command.
   - To set the attributes of an aggregated route, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **attribute-policy** *route-policy-name* command.

     To set the attributes of an aggregated route, you can also run the **peer route-policy** command.

     If **as-set** is set in the **aggregate** command and the **apply as-path** command is run to set the AS_Path attribute, the AS_Path attribute does not take effect.
   - To generate an aggregated route according to certain specific routes, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **origin-policy** *route-policy-name* command.

   Manual aggregation is valid for the existent routing entries in the local BGP routing table. For example, if the route with the mask length longer than 16 such as 10.1.1.1/24, does not exist in the BGP routing table, BGP does not advertise the aggregated route after the aggregate 10.1.1.1 16 command is used to aggregate routes.

5. Run:

```
commit
```

The configuration is committed.

----End

8.6.2 Configuring a BGP Router to Advertise Default Routes to Its Peer

A BGP router advertises a default route with the local address being the next-hop address to a specified peer for load balancing, regardless of whether there is a default route in the local routing table. In this manner, the number of routes on the network is greatly reduced.

Context

Default routes are usually used on the following networks:

- There are multiple EBGP neighbors, and full Internet routes will be received from each neighbor.

- There are multiple route reflectors (RRs), and full Internet routes will be received from each RR.

When load balancing is not performed on the network, a BGP peer receives at most one copy of active full Internet routes. When load balancing is performed on the network, the number of active routes received by a BGP peer is doubled, which causes the number of routes on the network to sharply increase. In this case, you can configure the local router to advertise only default routes to its BGP peer and use default routes for load balancing, which can greatly reduce the number of routes on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
peer { group-name | ipv4-address } default-route-advertise [ route-policy route-policy-name ] [ conditional-route-match-all { ipv4-address1 { mask1 | mask-length1 } } &<1-4> | conditional-route-match-any { ipv4-address2 { mask2 | mask-length2 } } &<1-4> ]
```

The BGP router sends a default route to a peer or a peer group.

NOTE

After the **peer default-route-advertise** command is used, a BGP router sends a default route with the local address being the next hop address to a specified peer, regardless of whether there is a default route in the routing table.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.6.3 Configuring a BGP Router to Advertise the Community Attribute

The community attribute is used to simplify the application, maintenance, and management of routing policies.

Context

The community attribute is transmitted between BGP peers and is not restricted by ASs. The community allows a group of routes to share the same policy. Before advertising a route with the community attribute to peers, a BGP router can change the original community attribute of this route.

Procedure

- Configure a BGP router to advertise the community attribute to its peer.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.
  3. Run:

     ```
     ipv4-family unicast
     ```

     The IPv4 unicast address family view is displayed.
  4. Configure a BGP router to advertise the community attribute to a peer or a peer group according to the actual networking.
     - To advertise the standard community attribute to a peer or a peer group, run the **peer** { *ipv4-address* | *group-name* } **advertise-community** command.

       By default, the community attribute is not advertised to any peer or peer group.
     - To advertise the extended community attribute to a peer or a peer group, run the **peer** { *ipv4-address* | *group-name* } **advertise-ext-community** command.

       By default, the extended community is not advertised to any peer or peer group.
  5. Run:

     ```
     commit
     ```

     The configuration is committed.
- Apply a routing policy to the routes to be advertised.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.
  3. Run:

     ```
     ipv4-family unicast
     ```

     The IPv4 unicast address family view is displayed.
  4. Run:

     ```
     peer { ipv4-address | group-name } route-policy route-policy-name export
     ```

     An outbound routing policy is configured.

NOTE

When configuring a BGP community attribute, you must use a routing policy to define the specific community attribute, and then apply the routing policy when advertising routing information.

For the configuration of routing policies, see chapter "Routing Policy Configuration".

5. Run:

```
commit
```

The configuration is committed.

----End

8.6.4 Setting the Interval for Sending Update Packets

When a route changes, a router sends an Update packet to notify its peer. If a route changes frequently, to prevent the router from sending Update packets for every change, you can set the interval for sending Update packets for changes of this route.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
peer { ipv4-address | group-name } route-update-interval interval
```

The interval for sending Update packets is set.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.6.5 Checking the Configuration

After the advertisement of routing information is controlled, you can check the advertised routes that match a specified filter.

Prerequisite

The configurations of controlling the advertisement of routing information are complete.

Procedure

- Run the **display bgp network** command to check the routes advertised by a BGP router.
- Run the **display bgp routing-table cidr** command to check CIDR routes.
- Run the **display bgp routing-table peer** *ipv4-address* { **advertised-routes** | **received-routes** } [**statistics**] command to check the routes advertised by a BGP router to its peer or the routes received by a BGP router from its peer.

----End

Example

Run the **display bgp routing-table peer** *ipv4-address* **advertised-routes** command, and you can view the routes advertised by a BGP router.

```
<HUAWEI> display bgp routing-table peer 200.1.3.2 advertised-routes

 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 6
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          0.0.0.0         0                     0      ?
 *>   200.1.2.1/32       0.0.0.0         0                     0      ?
 *>   200.1.3.0          0.0.0.0         0                     0      ?
 *>   200.1.3.1/32       200.1.3.2       0                     0      30?
 *>   200.1.3.2/32       0.0.0.0         0                     0      ?
 *>   200.1.4.1/32       200.1.3.2       0                     0      30?
```

8.7 Controlling BGP Route Selection

You can change the policies for BGP route selection by configuring BGP route attributes.

Applicable Environment

BGP has many route attributes that are used to define routing policies and describe the features of BGP route prefixes. By configuring these attributes, you can change BGP routing policies.

Pre-configuration Tasks

Before controlling BGP route selection, complete the following tasks:

- **Configuring Basic BGP Functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

8.7.1 Setting the BGP Preference

Setting the BGP preference can affect route selection between BGP and another routing protocol.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
preference { external internal local | route-policy route-policy-name }
```

By default, the preference value is as follows:

- EBGP external routes: 255
- IBGP internal routes: 255
- BGP local routes: 255

The BGP preference is set.

BGP has the following types of routes:

- Routes learned from external peers (EBGP)
- Routes learned from internal peers (IBGP)
- Locally originated routes, a local route refers to the route aggregated through an aggregation command, that is, (**summary automatic**) or (**aggregate**)

You can set different preferences for the three types of routes.

You can also apply routing policies to set preferences for the specified routes that meet the requirements. You can set default preferences for the routes that do not meet the requirements.

NOTE

Currently, you cannot run the **peer route-policy** command to apply routing policies to set the preference for BGP.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.7.2 Setting the Preferred Values for BGP Routes

After the preferred values are set for BGP routes, the route with the largest preferred value is preferred when multiple routes to the same destination exist in the BGP routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
peer { group-name | ipv4-address } preferred-value value
```

The preferred values of all the routes learned from a specified peer are set.

By default, the original preferred value of a route learned from a peer is 0.

After the **peer preferred-value** command is run, all the routes learned from a peer have the same preferred value.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.7.3 Setting the Default Local_Pref Attribute for the Local Device

The function of the Local_Pref attribute is similar to that of the preferred value. The priority of the Local_Pref attribute, however, is lower than that of the preferred value.

Context

The Local_Pref attribute is used to determine the optimal route for the traffic that leaves an AS. When a BGP device obtains multiple routes to the same destination address but with different next hops from different IBGP peers, the BGP device prefers route with the largest Local_Pref.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
default local-preference local-preference
```

The default Local_Pref attribute is set for the local device.

By default, the local preference of BGP is 100.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.7.4 Setting the MED Attribute

The MED attribute equals the metric used by an IGP. After the MED attributes of routes are set, an EBGP peer selects the route with the smallest MED value for the traffic that enters an AS.

Context

The MED serves as the metric used by an IGP. It is used to determine the optimal route when traffic enters an AS. When a BGP router obtains multiple routes to the same destination address but with different next hops through EBGP peers, the route with the smallest MED value is selected as the optimal route.

Procedure

- Set the default MED value for the local device.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.
  3. Run:

     ```
     ipv4-family unicast
     ```

     The IPv4 unicast address family view is displayed.
  4. Run:

     ```
     default med med
     ```

     The default MED is set.
  5. Run:

     ```
     commit
     ```

     The configuration is committed.
- Compare the MED values of routes from different ASs.
  1. Run:

     ```
     system-view
     ```

The system view is displayed.

2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

4. Run: **compare-different-as-med** The MED values of routes from different ASs are compared.

   By default,a BGP router compares only the MED values of routes from the same AS (different peers). You can run this command to enable a BGP router to compare the MED values of routes from different ASs.

5. Run:

```
commit
```

The configuration is committed.

- Configure the processing method when the MED value of a route is not set.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

4. Run:

```
bestroute med-none-as-maximum
```

The MED value of a route is taken as the maximum value when the MED value of the route is not set.

After the **bestroute med-none-as-maximum** command is run, a BGP takes the MED value of a route that has no MED value as the maximum value. Otherwise, the MED value of the route is 0.

5. Run:

```
commit
```

The configuration is committed.

----End

8.7.5 Setting the Next_Hop Attribute

By setting the Next_Hop attribute, you can flexibly control BGP route selection.

Context

The Next_Hop attribute of BGP is different from that of an IGP. It is not necessarily the IP address of a neighboring router.

Procedure

- Configure a router to change the next hop when the router advertises a route to an IBGP peer.

  Do as follows on an IBGP router.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     ipv4-family unicast
     ```

     The IPv4 unicast address family view is displayed.

  4. Run:

     ```
     peer { ipv4-address | group-name } next-hop-local
     ```

     The address of the router is set as the next hop when the router advertises a route.

     On certain networks, to ensure that an IBGP peer can find the correct next hop, you can configure the local router to change the next hop of a route to the address of the local router when the local router advertises the route to its IBGP peer.

     By default, a router does not change the next hop address when advertising a route to its IBGP peer.

     NOTE

     If BGP load balancing is configured, the local router changes the next hop address as its address when advertising a route to an IBGP peer group, regardless of whether the **peer next-hop-local** command is run.

  5. Run:

     ```
     commit
     ```

     The configuration is committed.

- Configure the ASBR not to change the next hop when the ASBR advertises a route to an EBGP peer.

  Do as follows on the PE router.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

The BGP view is displayed.

3. Run: **ipv4-family vpnv4** [**unicast**] The BGP-VPNv4 sub-address family view is displayed.
4. Run:

   ```
   peer { group-name | ipv4-address } next-hop-invariable
   ```

   The router is configured not to change the next hop address when the router advertises a route to an EBGP peer.

   By default, PEs of different ASs set up EBGP peer relationships, and change the next hop address when advertising routes.
5. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.7.6 Setting the AS_Path Attribute

The AS_Path attribute is used to prevent routing loops and control route selection.

Procedure

- Allow repeated local AS numbers.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.
  3. Run:

     ```
     ipv4-family unicast
     ```

     The IPv4 unicast address family view is displayed.
  4. Run:

     ```
     peer { ipv4-address | group-name } allow-as-loop [ number ]
     ```

     Repeated local AS numbers are allowed.

     Generally, a BGP router checks the AS_Path attribute of a route sent from a peer. If the local AS number is carried by the route, the BGP router ignores this route to avoid routing loops.

     In some special applications, you can use the **peer** { *ipv4-address* | *group-name* } **allow-as-loop** [*number*] command to allow the AS_Path attribute of a route sent from a peer to contain the local AS number. You can also set the number of repeated local AS numbers.
  5. Run:

     ```
     commit
     ```

     The configuration is committed.

- Prevent the AS_Path attribute from becoming one of the route selection rules.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     bestroute as-path-ignore
     ```
     The AS_Path attributed is not configured as one of the route selection rules.
- Configure a fake AS number.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     peer { ipv4-address | group-name } fake-as fake-as-number
     ```
     A fake AS number is configured.

     This command is used to hide an actual AS number. EBGP peers in other ASs can learn only this fake AS number. That is, when the peers in other ASs need to specify the number of the AS where the local peer resides, you need to specify the AS number as the fake AS number.

     NOTE

     The **peer fake-as** command is applicable to only EBGP peers.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.
- Substitute the AS number in the AS-Path attribute.

  If the AS_Path attribute of a route to be advertised to a peer contains the number of the AS where the peer resides, the local router substitutes the AS number of the peer with its AS number before advertising the route. On a VPN, if CEs of different sites use the same AS number, the **peer substitute-as** command needs to be run.

CAUTION

Use the **peer substitute-as** command with caution. If the command is not used properly, a routing loop is caused.

1. Run:
   ```
   system-view
   ```
   The system view is displayed.
2. Run:
   ```
   bgp as-number
   ```
   The BGP view is displayed.
3. Run:
   ```
   ipv4-family vpn-instance
   ```
   The VPN instance view is displayed.
4. Run:
   ```
   peer { ipv4-address | group-name } substitute-as
   ```
   The AS number in the AS_Path attribute of a route is substituted with the local AS number.
5. Run:
   ```
   commit
   ```
   The configuration is committed.

- Configure the AS_Path attribute to carry only the public AS number.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     peer { ipv4-address | group-name } public-as-only
     ```
     The AS_Path attribute is configured to carry only the public AS number.

     Normally, the AS number ranges from 1 to 4294967295. The public AS number ranges from 1 to 64511, and 65536 (1.0 in the format of x.y) to 4294967295 (65535.65535 in the format of x.y), and the private AS number ranges from 64512 to 65534. 65535 is used as the reserved AS number in certain applications.

     The public AS number can be used on the Internet, because Internet addresses are managed and assigned by the Internet Assigned Number Authority (IANA). The private AS number cannot be advertised to the Internet and is used only in an internal routing domain.

Generally, the route advertised by a BGP router to its peer carries an AS number (either public or private AS number). In certain cases, the private AS number does not need to be transmitted. You can then use the command to configure the AS_Path attribute to carry only the public AS number.

NOTE

The **peer public-as-only** command is applicable to only EBGP peers.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

- Set the maximum number of AS numbers in the AS-Path attribute.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     as-path-limit as-path-limit-num
     ```

     The maximum number of AS numbers in the AS-Path attribute is set.

     By default, the maximum number of AS numbers in the AS_Path attribute is 255.

     After the **as-path-limit** command is run, a router checks whether the number of AS numbers in the AS-Path attribute of a received route exceeds the maximum value. If the number of AS numbers exceeds the maximum value, the router discards the route. Therefore, if the maximum number of AS numbers in the AS-Path attribute of a route is set too small, the route is lost.

  4. Run:

     ```
     commit
     ```

     The configuration is committed.

- Prevent a BGP device from checking the first AS number contained in the AS_Path attribute of an Update message received from an EBGP peer.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     undo check-first-as
     ```

     The BGP device is prevented from checking the first AS number contained in the AS_Path attribute of an Update message received from an EBGP peer.

     By default, a BGP device checks whether the first AS number contained in the AS_Path attribute of an Update message received from an EBGP peer is the same as the number of the AS where the EBGP peer resides. If the numbers are not the same, the BGP device discards the Update message and closes the EBGP connection with the EBGP peer.

CAUTION

Exercise caution when running the **undo check-first-as** command, because use of this command increases the possibility of routing loops.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

   After the configuration is complete, run the **refresh bgp** command if you want to check the received routes again.

----End

8.7.7 Checking the Configuration

After configuring BGP route selection, you can check information about route attributes.

Prerequisite

All configurations of BGP route attributes are complete.

Procedure

- Run the **display bgp paths** [*as-regular-expression*] command to check information about BGP paths.
- Run the **display bgp routing-table different-origin-as** command to check routes with different source ASs but the same destination address.
- Run the **display bgp routing-table regular-expression** *as-regular-expression* command to check routes matching the AS regular expression.
- Run the **display bgp routing-table** [*network*] [*mask* | *mask-length*] [**longer-prefixes**] command to check information about the BGP routing table.
- Run the **display bgp routing-table community** [*community-number* | *aa:nn*] &<1-13> [**internet** | **no-advertise** | **no-export** | **no-export-subconfed**] * [**whole-match**] command to check routes matching a specified BGP community attribute.
- Run the **display bgp routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check the routes matching a specified BGP community filter.

----End

Example

Run the **display bgp paths** command, and you can view information about BGP paths.

```
<HUAWEI> display bgp paths

Total Number of Routes: 4
```

```
Total Number of Paths: 4

    Address        Refcount  MED         Path/Origin
    0x50EEF20      1         0            ?
    0x50EEEB8      1         0            ?
    0x50EEF88      1                     i
    0x50EF0C0      1         0           65410?
```

8.8 Configuring BGP Routing Policies

By using BGP routing policies, you can flexibly control the sending and receiving of routes.

Applicable Environment

Routing policies can set and re-set BGP route attributes through the predefined conditions, which is a flexible and effective method of controlling BGP route selection. By using BGP routing policies, you can flexibly control the sending and receiving of routes.

Based on the inbound (or outbound) routing policy specified by the peer, you can configure the associated inbound (or outbound) conditions (**if-match** clauses) to filter routes, and set or modify route attributes through **apply** clauses. In this manner, the routes that match the routing policy can be received or advertised.

Pre-configuration Tasks

Before configuring BGP routing policies, complete the following tasks:

- Configuring IP addresses for interfaces to make neighboring nodes reachable at the network layer
- **Configuring Basic BGP Functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

8.8.1 Configuring BGP Access Lists

To view the BGP running status and routing policies, you can use BGP-related ACLs.

Context

BGP has two private access lists, that is, AS_Path filter and community filter. The lists can be used to display the BPG running status and routing policies.

- AS_Path filter

  The AS_Path filter is used to match the AS_Path attributes contained in BGP routes and filter routes that do not meet matching rules. You can define multiple rules (permit or deny) for the same filter.
- Community filter

  The community filter maintains a series of community attribute lists. The community attribute lists are classified into two types, that is, standard community access lists and extended community access lists.

Procedure

- Configure the AS_Path filter.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ip as-path-filter as-path-filter-number { permit | deny } regular-expression
     ```
     The AS_Path filter is configured.

     When applying a routing policy to BGP routes, you can run the **peer as-path-filter** command to filter out the routes that do not meet the matching rules.

     The AS_Path filter defines matching rules with a regular expression. The regular expression is composed of the following parts:

     – Meta character: defines matching rules.

     – General character: defines matching objects.

Table 8-1 Description of metacharacters

Metacharacter	Description
\	Erase character.
.	Matches any single character including the space except for \n.
*	Characters to its left are displayed 0 times or many times continuously in the target object.
+	Characters to its left are displayed once or many times in the target object.
|	The relationship between characters on its left and right is "OR."
^	Characters on its right must be displayed at the beginning of the target object.
$	Characters on its left must be displayed at the end of the target object.
[xyz]	Matches the character listed in the square bracket.
[^xyz]	Matches any character that is not listed in the square bracket (^ is on the left of the character).
[a-z]	Matches any character within the specified range.
[^a-z]	Matches any character that is not within the specified range.
{n}	The matching is displayed n times (n is a non-negative integer).

Metacharacter	Description
{n,}	The matching is displayed at least n times (n is a non-negative integer).
{n,m}	The matching is displayed n-m times (m and n are non-negative integers and n is smaller than or equal to m). There is no space between n and m.

For example, ^10 indicates that only the AS_Path attribute with the value 10 as the first character is matched. ^ indicates that the beginning of a string character is matched.

You can define multiple rules (permit or deny) for the same filter. During the matching, the relationship between these rules is OR. That is, when a route meets one of the matching rules, it indicates that the route matches this AS_Path filter.

NOTE

For the usage of the regular expression, see the *HUAWEI NetEngine5000E Core Router Configuration Guide - Basic Configurations*.

3. Run:
   ```
   commit
   ```
   The configuration is committed.

- Configure the community filter.

  There are two types of community filters, that is, the standard community filter and advanced community filter. The advanced community filter supports the regular expression, and is more flexible than the standard community filter.

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ip community-filter
     ```
     The community filter is configured.
     - To configure the standard community filter, run the **ip community-filter** { **basic** *comm-filter-name* { **permit** | **deny** } [*community-number* | *aa:nn*] * &<1-9> | *basic-comm-filter-num* { **permit** | **deny** } [*community-number* | *aa:nn*] * &<1-16> } [**internet** | **no-export-subconfed** | **no-advertise** | **no-export**] * command.
     - To configure the advanced community filter, run the **ip community-filter** { **advanced** *comm-filter-name* | *adv-comm-filter-num* } { **permit** | **deny** } *regular-expression* command.
  3. Run:
     ```
     commit
     ```
     The configuration is committed.

- Configure the extended community filter.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   ip extcommunity-filter extcomm-filter-number { permit | deny } rt { as-number : nn | ipv4-address : nn } &<1-16>
   ```

   The extended community filter is configured.

   You can define multiple entries for the same extended community filter. During the matching, the relationship between the entries is OR. That is, when a route meets one of the entries, it indicates that the route matches the extended community filter.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.8.2 Configuring a Route-Policy

A route-policy is used to match routes or attributes of routes, and to change the attributes when the matching rules are met.

Context

A route-policy is used to match routes or attributes of routes, and to change the attributes when the matching conditions are met. The matching conditions can be other access lists.

A route-policy can consist of multiple nodes, and each node can comprise the following clauses:

- **if-match** clause

  The **if-match** clause defines matching rules of routes. That is, routing information meets the requirements of the current route-policy and the matched object is some attributes of the routing information.

- **apply** clause

  The **apply** clause specifies actions, that is, configuration commands are run after routes meet the matching rules specified by the **if-match** clause. The **apply** clause can change some attributes of routes.

NOTE

This section describes only BGP-related routing policies. For detailed configurations of the route-policy, see the chapter "Routing Policy Configuration."

Procedure

- Create a route-policy.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     route-policy route-policy-name { permit | deny } node node
     ```

A node of a route-policy is created and the route-policy view is displayed.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure the **if-match** clause.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   route-policy route-policy-name { permit | deny } node node
   ```

   The route-policy view is displayed.

3. Run the following commands to configure the **if-match** clause of the route-policy on the current node.

   - To match BGP routes with the AS_Path attribute, run the **if-match as-path-filter** *as-path-filter-number* &<1-16> command.
   - To match BGP routes with the community attribute, run either of the following commands:
     - **if-match community-filter** { *basic-comm-filter-num* [**whole-match**] | *adv-comm-filter-num* }* &<1-16>
     - **if-match community-filter** *comm-filter-name* [**whole-match**]
   - To match BGP routes with the extended community attribute, run the **if-match extcommunity-filter** *extcomm-filter-number* &<1-16> command.

   The commands in Step 3 can be run regardless of the sequence. There may be no or multiple **if-match** clauses on a node.

   NOTE

   - For the same node of a route-policy, the relationship between the **if-match** clauses is AND. That is, the actions defined by the **apply** clause can be performed only when a route must meet all the matching rules.
   - If no **if-match** clause is specified, all the routes can pass the filtering on the node.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure the **apply** clause.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   route-policy route-policy-name { permit | deny } node node
   ```

   The route-policy view is displayed.

3. Run the following commands to configure the **apply** clause of the route-policy on the current node.

- To substitute the AS number in the AS_Path attribute of a BGP route or add an AS number to the AS_Path attribute of a BGP route, run the **apply as-path** *as-number* &<1-10> [**additive**] command.
- To delete a specified BGP community attribute, run the **apply comm-filter** *comm-filter-number* **delete** command.

  TIP

  The **apply comm-filter delete** command is used to delete the community attribute according to the specified value in the community filter. Each community filter defined by the **ip community-filter** command contains only one community attribute. To delete several community attributes, run the **apply comm-filter delete** command for several times. If multiple community attributes are configured for the same community filter, these attributes cannot be deleted. For examples, see the *HUAWEI NetEngine5000E Core Router Command Reference*.
- To delete the community attribute of a BGP route, run the **apply community none** command.
- To set the community attribute of a BGP route, run the **apply community** { { *community-number* | *aa:nn* } &<1-32> | **internet** | **no-advertise** | **no-export** | **no-export-subconfed** }* [**additive**] command.
- To set the MED, run the **apply cost** { [*apply-type*] *cost* | **inherit** } command.
- To set the MED value of BGP routes as the IGP cost of the next hop, run the **apply cost-type internal** command.
- To set the BGP extended community attribute (Route-Target), run the **apply extcommunity** { **rt** { *as-number:nn* | *ipv4-address:nn* } } &<1-16> [**additive**] command.
- To set the Local-Pref attribute of a BGP route, run the **apply local-preference** *preference* command.
- To set the Origin attribute of a BGP route, run the **apply origin** { **igp** | **egp** *as-number* | **incomplete** } command.
- To set the preferred value of a BGP route, run the **apply preferred-value** *preferred-value* command.
- To set the dampening parameters of a BGP route, run the **apply dampening** *half-life-reach reuse suppress ceiling* command.

The commands in Step 3 can be run regardless of the sequence.

4. Run:

```
commit
```

The configuration is committed.

----End

8.8.3 Configuring the Policy for Advertising BGP Routes

After BGP filters imported routes, only the eligible routes are added to the local BGP routing table and advertised to BGP peers.

Context

BGP can use a routing policy to filter the globally advertised routes or only the routes advertised to a certain peer or peer group.

Procedure

- Configure BGP to filter the routes to be globally advertised.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-
     name } export [ protocol [ process-id ] ]
     ```
     BGP is configured to filter the routes to be globally advertised.

     After BGP filters the routes to be imported through the **import-route** command, only the eligible routes are added to the local BGP routing table and advertised to BGP peers.

     If *protocol* is set, only the routes of a specific routing protocol are filtered. If *protocol* is not specified, all the BGP routes to be advertised are filtered, including the imported routes.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.
- Configure BGP to filter the routes to be advertised to a specified peer or peer group.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run the following commands to configure BGP to filter routes according to different filters for specified peers.
     - To filter routes based on an ACL, run the **peer** { *ipv4-address* | *group-name* } **filter-policy** { *acl-number* | **acl-name** *acl-name* } **export** command.
     - To filter routes based on the AS_Path filter, run the **peer** { *ipv4-address* | *group-name* } **as-path-filter** *as-path-filter-number* **export** command.
     - To filter routes based on the IP prefix list, run the **peer** { *ipv4-address* | *group-name* } **ip-prefix** *ip-prefix-name* **export** command.

- To filter routes based on the route-policy filter, run the **peer** { *ipv4-address* | *group-name* } **route-policy** *route-policy-name* **export** command.

  NOTE

  The routing policy set in the **peer route-policy export** command does not support a certain interface as one of the matching rules. That is, the routing policy does not support the **if-match interface** command.

The members of a peer group and the peer group can use different export routing policies. That is, each member in the peer group can select its policy when advertising routes.

5. Run:

```
commit
```

The configuration is committed.

----End

8.8.4 Configuring the Policy for Receiving BGP Routes

The routes received by BGP are filtered. Only those routes that meet the matching rules are received by BGP and are added to the routing table.

Context

BGP can use a routing policy to filter globally received routes or only the routes received from a certain peer or peer group.

When the router running BGP is attacked or network configuration errors occur, the router receives a large number of routes from its neighbor. As a result, a large number of resources of the router are consumed. Therefore, the administrator must limit the resources used by the router based on network planning and the capacity of the router. BGP provides peer-based route control to limit the number of routes to be sent by a neighbor. Thus, the preceding problem is addressed.

Procedure

- Configure BGP to filter the globally received routes.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

4. Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-
name } import
```

BGP is configured to filter globally received routes.

Only the routes that match the policy for receiving routes can be received by BGP and added to the routing table.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure BGP to filter routes received from a specified peer or peer group.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   ipv4-family unicast
   ```

   The IPv4 unicast address family view is displayed.

4. Run the following command to configure BGP to filter routes received from peers according to different filters.

   - To filter routes based on an ACL, run the **peer** { *ipv4-address* | *group-name* } **filter-policy** { *acl-number* | **acl-name** *acl-name* } **import** command.
   - To filter routes based on the AS_Path filter, run the **peer** { *ipv4-address* | *group-name* } **as-path-filter** *as-path-filter-number* **import** command.
   - To filter routes based on the IP prefix list, run the **peer** { *ipv4-address* | *group-name* } **ip-prefix** *ip-prefix-name* **import** command.
   - To filter routes based on the route-policy filter, run the **peer** { *ipv4-address* | *group-name* } **route-policy** *route-policy-name* **import** command.

   NOTE

   The routing policy set in the **peer route-policy import** command does not support a certain interface as one of the matching rules. That is, the routing policy does not support the **if-match interface** command.

   The members of a peer group and the peer group can use different import routing policies to filter routes. That is, each peer can select its policy when receiving routes.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

- Limit the number of the routes received from a peer.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   ipv4-family unicast
   ```

The IPv4 unicast address family view is displayed.

4. Run:

```
peer { group-name | ipv4-address } route-limit limit [ percentage ]
[ alert-only | idle-forever | idle-timeout times ]
```

The number of routes that can be received from a peer or peer group is set.

The command provides the limit on the number of received routes based on peers. You can configure specific parameters as required to control BGP after the number of the routes received from a peer exceeds the threshold.

- **alert-only**: The peer relationship is kept. No route is received after the number of received routes exceeds the threshold, and an alarm is generated and recorded in the log.
- **idle-forever**: The peer relationship is interrupted. The router does not retry setting up a connection. An alarm is generated and recorded in the log. In this case, run the **display bgp peer** [**verbose**] command, and you can find that the status of the peer is Idle. To restore the BGP connection, run the **reset bgp** command.
- **idle-timeout**: The peer relationship is interrupted. The router retries setting up a connection after the timer expires. An alarm is generated and recorded in the log. In this case, run the **display bgp peer** [**verbose**] command, and you can find that the status of the peer is Idle. To restore the BGP connection before the timer expires, run the **reset bgp** command.
- If none of the preceding parameters is set, the peer relationship is disconnected. The router retries setting up a connection after 30 seconds. An alarm is generated and recorded in the log.

NOTE

If the number of routes received by the local router exceeds the upper limit and the **peer route-limit** command is used for the first time, the local router and its peer reestablish the peer relationship, regardless of whether **alert-only** is set.

5. Run:

```
commit
```

The configuration is committed.

----End

8.8.5 Configuring BGP Soft Resetting

When routing policies are changed, the system can refresh the BGP routing table dynamically without interrupting BGP connections.

Context

BGP peers perform capability negotiation through Open messages during the initialization of a BGP connection. To enable the new capability or disable the existing capability (for example, to change the capability in an address family), you need to reset the connection. As a result, other services on the connection are interrupted. The dynamic capability negotiation allows dynamically updating the capability on the established BGP connection, and notifies the capability to be updated through a Capability message. In this manner, the capability is updated between BGP peers without interrupting the BGP connection.

In the implementation of the NE5000E, BGP supports the route-refresh capability. When routing policies are changed, the system can refresh the BGP routing table dynamically without interrupting BGP connections.

- If the peer router supports the route-refresh capability, you can run the **refresh bgp** command to manually perform soft resetting for the BGP connection. The routing table is thus refreshed.
- If the peer router does not support the route-refresh capability, you can run the **peer keep-all-routes** command. In this manner, the BGP routing table can be refreshed.

Procedure

- Enable the route-refresh capability.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     peer { ipv4-address | group-name } capability-advertise { route-refresh |
     4-byte-as | conventional }
     ```
     The route-refresh capability is enabled.

     When all BGP routers are enabled with the route-refresh capability, the local router sends a route-refresh packet to its peer if the BGP routing policy changes. After receiving the packet, the peers send the packet to the local router. In this case, the BGP routing table is dynamically refreshed and the new routing policy is applied with the BGP connection being kept.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.
- Keep all the routing updates of a peer.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     peer { ipv4-address | group-name } keep-all-routes
     ```
     All the routing updates of a peer are kept.

After this command is used, all routing updates sent by a specified peer are kept regardless of whether the filtering policy is used. When the local routing policy changes, such information can be used to regenerate BGP routes.

5. Run:

```
commit
```

The configuration is committed.

- Softly reset a BGP connection.

1. Run:

```
refresh bgp [ vpn-instance vpn-instance-name | vpnv4 ] { all | ipv4-
address | group group-name | external | internal } { export | import }
```

A BGP connection is softly reset.

Run the **refresh bgp** command in the user view.

----End

8.8.6 Checking the Configuration

After the configurations of BGP routing policies are complete, you can check information about routes advertised and received by BGP.

Prerequisite

All configurations of BGP route policies are complete.

Procedure

- Run the **display bgp network** command to check routes imported by BGP through the **network** command.
- Run the **display bgp routing-table as-path-filter** *as-path-filter-number* command to check the routes matching a specified AS_Path filter.
- Run the **display bgp routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check the routes matching a specified BGP community filter.
- Run the **display bgp routing-table peer** *ipv4-address* { **advertised-routes** | **received-routes** } [**statistics**] command to check the routes advertised by a BGP router to its peer or received by a BGP router from its peer.

----End

Example

Run the **display bgp network** command, and you can view routes imported by BGP through the **network** command.

```
<HUAWEI> display bgp network
  BGP Local Router ID is 1.1.1.9
  Local AS Number is 10(Public)
  Network          Mask             Route-policy
  1.2.0.0          255.255.0.0
  3.0.0.0          255.0.0.0        Policy1
  4.4.4.0          255.255.255.0
```

8.9 Configuring Prefix-based BGP ORF

Prefix-based BGP ORF enables a device to send its peer the prefix-based inbound policy that can be used by the peer to filter routes to be sent.

Applicable Environment

When a device wants to receive only required routes from its peer but the peer cannot maintain different outbound policies for each connected device, you can configure prefix-based ORF to meet the requirements of the two devices.

Pre-configuration Tasks

Before configuring prefix-based BGP ORF, complete the following tasks:

- **Configuring Basic BGP Functions**
- **Configuring an IPv4 Prefix List**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
peer { group-name | ipv4-address } ip-prefix ip-prefix-name import
```

The prefix-based inbound policy is configured on a peer or a peer group.

Step 5 Run:

```
peer { group-name | ipv4-address } capability-advertise orf [ cisco-compatible ] ip-prefix { both | receive | send }
```

The prefix-based ORF is configured on a BGP peer or a BGP peer group.

By default, prefix-based ORF is not enabled on a BGP peer or a BGP peer group.

Step 6 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

Run the following command to check the previous configuration.

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to check detailed information about BGP peers.
- Run the **display bgp peer** *ipv4-address* **orf-info ip-prefix** command to check information about the prefix-based ORF received by a device from a specified peer.

8.10 Configuring Path MTU Auto Discovery

By configuring path MTU auto discovery, you can discover the minimum MTU on the network path from the source to the destination.

Applicable Environment

When hosts need to communicate across multiple networks, the smallest MTU on the communication path is most important to both ends. This is because different networks along the communication path have different MTUs of the link layer. The minimum MTU on the communication path is called the path MTU.

During communication, path MTUs of host depend on the selected path and thus may change. In addition, the path MTUs of the inbound direction and outbound direction may be inconsistent. Path MTU auto discovery is the process of discovering the minimum MTU on the network path from the source to the destination. The discovered path MTU is used to ensure proper fragmentation during packet transmission.

Pre-configuration Tasks

Before configuring path MTU auto discovery, complete the following task:

- **Configuring Basic BGP Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
peer { group-name | ipv4-address } path-mtu auto-discovery
```

Path MTU auto discovery is enabled.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After configuring path MTU auto discovery, you can check whether it is configured.

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to view detailed information about the BGP peer to check whether path MTU auto discovery is successfully configured.

8.11 Configuring BGP Load Balancing

By configuring BGP load balancing, you can properly use network resources.

Applicable Environment

Equal-cost BGP routes can be generated for load balancing only when the first 8 attributes, described in "Policies for BGP Route Selection" in **8.2 BGP Features Supported by the NE5000E**, of the routes are the same and the BGP routes have the same AS-Path attribute.

You can change load balancing rules by performing configurations, for example, ignoring the comparison of AS_Path attributes, ignoring the comparison of IGP metrics. When performing these configurations, ensure that no routing loops occur.

Pre-configuration Tasks

Before configuring BGP load balancing, complete the following task:

- **Configuring Basic BGP Functions**

Procedure

- Set the number of BGP routes to be used for load balancing.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv4-family unicast
     ```
     The IPv4 unicast address family view is displayed.
  4. Run:
     ```
     maximum load-balancing number
     ```
     The number of BGP routes to be used for load balancing is set.

     By default, the number of BGP routes to be used for load balancing is 1, meaning that load balancing is not implemented.
  5. (Optional) Run:
     ```
     load-balancing as-path-ignore
     ```
     The router is configured not to compare the AS-Path attributes of the routes to be used for load balancing.

By default, the router compares the AS-Path attributes of the routes to be used for load balancing.

NOTE

- If there are multiple routes to the same destination but these routes pass through different ASs, load balancing cannot be implemented among these routes by default. To implement load balancing among these routes, run the **load-balancing as-path-ignore** command. After the **load-balancing as-path-ignore** command is run, the device no longer compares the AS-Path attributes of the routes to be used for load balancing. Therefore, exercise caution when using this command.
- The **load-balancing as-path-ignore** and **bestroute as-path-ignore** commands are mutually exclusive.

6. Run:

```
commit
```

The configuration is committed.

- Set the maximum number of EBGP and IBGP routes to be used for load balancing.

This configuration is used in a VPN where a CE is dual-homed to two PEs. When the CE and one PE belong to an AS and the CE and the other PE belong to a different AS, you can set the number of EBGP and IBGP routes to be used for load balancing. This allows VPN traffic to be balanced among EBGP and IBGP routes.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv4-family vpn-instance vpn-instance-name
```

The BGP-VPN instance view is displayed.

4. Run:

```
maximum load-balancing eibgp number
```

The maximum number of EBGP and IBGP routes is set for load balancing.

By default, the maximum number of EBGP and IBGP routes to be used for load balancing is not set.

5. (Optional) Run:

```
load-balancing as-path-ignore
```

The router is configured not to compare the AS-Path attributes of the routes to be used for load balancing.

6. Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

All configurations are complete.

- Run the **display bgp routing-table** [*network*] [*mask* | *mask-length*] [**longer-prefixes**] command to check information about the BGP routing table.
- Run the **display ip routing-table** [**verbose**] command to check information about the IP routing table.

Run the **display bgp routing-table** command, and you can view information about BGP load balancing. For example:

```
<HUAWEI>display bgp routing-table 8.1.1.0 24

 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Paths : 2 available, 1 best, 2 select
 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.1.2 (2.2.2.2)
 Route Duration: 0d00h03m55s
 Direct Out-interface: Pos1/0/0
 Original nexthop: 200.1.1.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, best, select, pre 255
 Advertised to such 2 peers:
    200.1.1.2
    200.1.2.2

 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.2.2 (3.3.3.3)
 Route Duration: 0d00h03m56s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, select, pre 255, not
selected for router ID
 Not advertised to any peers yet
```

8.12 Configuring BGP Route Dampening

By configuring BGP route dampening, you can suppress unstable BGP routes.

Applicable Environment

By configuring BGP route dampening, you can suppress unstable BGP routes. After BGP route dampening is configured, unstable routes neither are added to the BGP routing table nor are advertised to other BGP peers.

Route instability is reflected by route flapping. That is, a route in a routing table disappears and reappears frequently. In most situations, BGP is applied to complex networks where routes change frequently. Frequent route flapping consumes lots of bandwidths and CPU resources and even affects the normal operation of a network. To avoid the impact of frequent route flapping, BGP adopts route dampening to suppress unstable routes.

With specified-requirement route dampening, you can configure routes to be differentiated based on route-policies so that BGP can use different route dampening parameters to suppress different routes. You can also configure different route dampening parameters for different nodes in the same route-policy. When route flapping occurs, BGP can use different route dampening parameters to suppress the routes that match the route-policy. For example, on a network, the routes with a longer mask are set with a longer dampening time, and the routes with a shorter mask (such as the 8-bit mask) are set with a shorter dampening time.

Pre-configuration Tasks

Before configuring BGP route dampening, complete the following task:

- **Configuring Basic BGP Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
dampening [ half-life-reach reuse suppress ceiling | route-policy route-policy-name ] *
```

The parameters of BGP route dampening are set.

NOTE

The **dampening** command is valid for only EBGP routes.

When BGP route dampening is configured, the values of *reuse*, *suppress*, and *ceiling* should be in the relationship of *reuse*<*suppress*<*ceiling*.

If routes are differentiated based on route-policies, when the **dampening** command is run to reference a route-policy, BGP can use different route dampening parameters to suppress different routes.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

All configurations are complete.

- Run the **display bgp routing-table dampened** command to check BGP dampened routes.
- Run the **display bgp routing-table dampening parameter** command to check the parameters of BGP route dampening.

- Run the **display bgp routing-table flap-info** [**regular-expression** *as-regular-expression* | **as-path-filter** *as-path-filter-number* | *network-address* [{ *mask* | *mask-length* } [**longer-match**]]] command to check the statistics about flapping routes.

Run the **display bgp routing-table dampened** command, and you can view information about dampened BGP routes. For example:

```
<HUAWEI> display bgp routing-table dampened

 BGP Local router ID is 223.1.41.102
 Status codes: * - valid, > - best, d - damped
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 8
     Network            From              Reuse     Path/Origin
  d  8.6.244.0/23       223.1.41.247      01:06:25  65534 4837 174 11096 6356i
  d  9.17.79.0/24       223.1.41.247      01:06:25  65534 837 3356 23504 29777i
  d  9.17.110.0/24      223.1.41.247      01:06:25  65534 837 3356 23504 29777i
  d  61.57.144.0/20     223.1.41.247      01:06:25  65534 4837 10026 9924
18429,18429i
  d  63.76.216.0/24     223.1.41.247      01:06:25  65534 4837 701 26959i
  d  63.78.142.0/24     223.1.41.247      01:06:25  65534 4837 701 26959i
  d  63.115.136.0/23    223.1.41.247      01:06:25  65534 4837 701 26956i
  d  65.243.170.0/24    223.1.41.247      01:06:25  65534 4837 701 26959i
```

8.13 Setting Parameters for a BGP Peer Connection

By setting parameters for a BGP peer connection, you can adjust and optimize the BGP network performance.

Applicable Environment

By using various timers, BGP can minimize the impact of interfaces or routes frequently alternating between Up and Down.

After a BGP connection is set up between peers, the peers periodically send Keepalive messages to each other. In this case, the BGP connection is not regarded as closed by the peers. If a router does not receive any Keepalive message or any other types of messages from the peer within the specified hold time, the BGP connection is regarded as closed.

When a router creates a BGP connection with its peer, they need to negotiate the hold time. The smaller one of the hold time of the BGP router and that of its peer is considered as the negotiated hold time. If the negotiated hold time is 0, no Keepalive message is transmitted and the hold time is not detected.

NOTE

When the hold timer value changes, the BGP connection may be closed for a short time. This is because the router and its peer need to negotiate the value of the timer again.

Pre-configuration Tasks

Before setting parameters for a BGP peer connection, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

8.13.1 Configuring BGP Timers

Configuring timers properly can improve network performance. Changing the values of BGP timers will interrupt the peer relationship.

Context

CAUTION

If the **timer** command or the **peer timer** command is run to change the value of the associated timer, the BGP peer relationship set up between routers is interrupted. So, confirm the action before you use the command.

Procedure

- Configure timers globally.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     timer keepalive keepalive-time hold hold-time
     ```
     The timers are configured globally.

     The proper maximum interval for sending Keepalive messages is one third as long as the hold time and is not smaller than one second. Thus, if the hold time is not set to 0, it must be 3 seconds at least. By default, the Keepalive time is 60 seconds and the hold time is 180 seconds.

     When setting the values of the Keepalive timer and the Hold timer, note the following:

     - The values of *keepalive-time* and *hold-time* cannot be 0 at the same time. Otherwise, the BGP timers become invalid. That is, BGP does not detect link faults according to the timers.
     - The value of *hold-time* is greater than that of *keepalive-time*, such as, **timer keepalive 1 hold 65535**. If the hold time is too long, however, faults on a link cannot be detected in time.

     After two peers set up a connection, the values of *keepalive-time* and *hold-timehold-time* are negotiated by the two peers. The smaller value of *hold-time* contained in Open messages of both peers is taken as the value of *hold-time*. The smaller value of one third *hold-time* and *keepalive-time* is used as the value of *keepalive-time*.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure timers for a peer.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   peer { ipv4-address | group-name } timer keepalive keepalive-time hold
   hold-time
   ```

   The Keepalive time and hold time are set for a peer or peer group.

   For the relationship between the Keepalive time and the hold time, see **Configuring Timers Globally**.

   The timers configured for a peer takes precedence over the timers configured globally.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.13.2 Enabling Quick Resetting of EBGP Connections

After quick resetting of EBGP connections is enabled, BGP rapidly detects the failure on an EBGP link and then resets the BGP connection on the interface immediately.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ebgp-interface-sensitive
```

Quick resetting of EBGP connections is enabled.

After this function is enabled, BGP detects the failure on an EBGP link rapidly and then resets BGP connections on the interface immediately.

When an interface used by a BGP connection frequently alternates between Up and Down, you need to disable the function to prevent repeated establishment and deletion of the BGP session. This saves network bandwidths.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

8.13.3 Checking the Configuration

After parameters of a BGP peer connection are configured, you can check information about BGP peers and peer groups.

Prerequisite

All configurations of parameters of a BGP peer connection are complete.

Procedure

- Run the **display bgp peer** [**verbose**] command to check information about BGP peers.
- Run the **display bgp group** [*group-name*] command to check information about BGP peer groups.

----End

Example

Run the **display bgp peer verbose** command, and you can view information about BGP peers. For example:

```
<HUAWEI> display bgp peer verbose

         BGP Peer is 10.1.1.2,  remote AS 10
         Type: IBGP link
         BGP version 4, Remote router ID 10.1.1.2

  Group ID : 1
         BGP current state: Established, Up for 00h00m04s
         BGP current event: RecvUpdate
         BGP last state: Established
         BGP Peer Up count: 1
         Port: Local - 179        Remote - 30903
         Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Received  : Active Hold Time: 180 sec
         Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Peer optional capabilities:
         Peer supports bgp multi-protocol extension
         Peer supports bgp route refresh capability
         Peer supports bgp 4-byte-as capability
         Address family IPv4 Unicast: advertised and received
 Received: Total 3 messages
                  Update messages                1
                  Open messages                  1
                  KeepAlive messages             1
                  Notification messages          0
                  Refresh messages               0
 Sent: Total 3 messages
                  Update messages                1
                  Open messages                  1
```

```
                 KeepAlive messages             1
                 Notification messages          0
                 Refresh messages               0
 Authentication type configured: None
 Last keepalive received:2011-02-22 03:39:16
 Minimum route advertisement interval is 15 seconds
 Optional capabilities:
 Route refresh capability has been enabled
 4-byte-as capability has been enabled
 Peer Preferred Value: 0
 Routing policy configured:
 No routing policy is configured
```

8.14 Configuring a BGP Route Reflector

By configuring a BGP route reflector, you can solve the problem of establishing full-mesh connections between multiple IBGP peers.

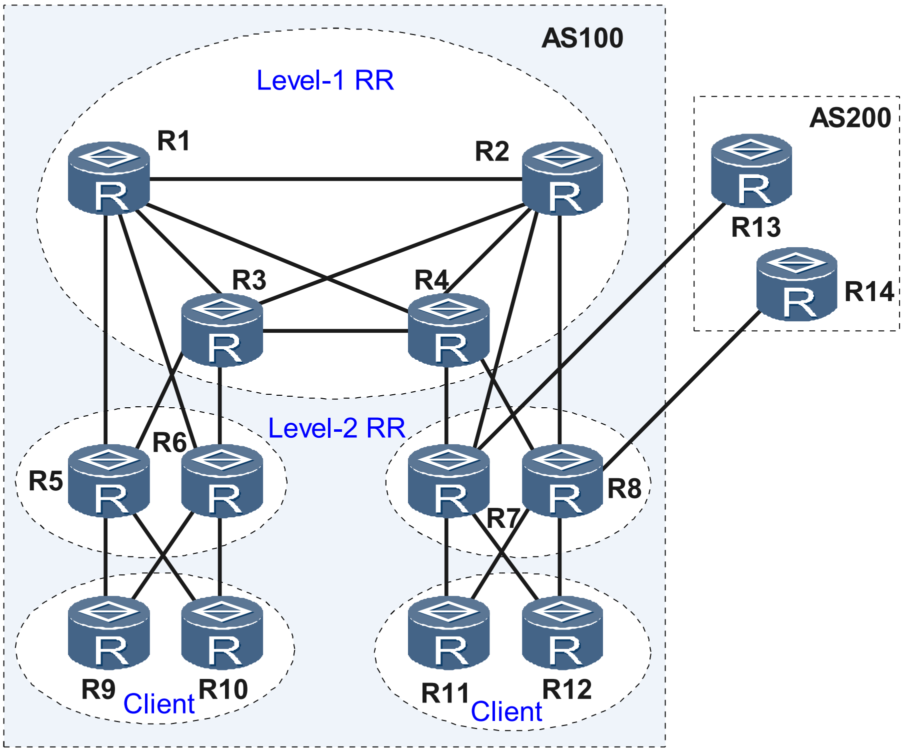
Applicable Environment

To ensure the connectivity between IBGP peers in an AS, you need to establish full-mesh connections between IBGP peers. When there are many IBGP peers, establishing a fully-meshed network is costly. In this case, you can configure a route reflector (RR) to solve the problem.

RRs can reduce the total number of IBGP connections. On a large network, to reduce the number of clients of each route reflector, you need to configure multiple RRs. Because full-mesh connections need to be established between RRs, there are still a large number of IBGP connections on the network. Therefore, the hierarchical RR is introduced to further reduce the number of IBGP connections.

Figure 8-3 shows the typical networking of a hierarchical RR. In this networking, R1, R2, R3, and R4 function as Level-1 RRs; R5, R6, R7, and R8 function as level-2 RRs and the clients of level-1 RRs. Level-1 RRs are not the clients of any RR and thus must be fully meshed. Level-2 RRs function as the clients of Level-1 RRs and thus do not need to be fully meshed.

Figure 8-3 Typical networking diagram for a hierarchical RR



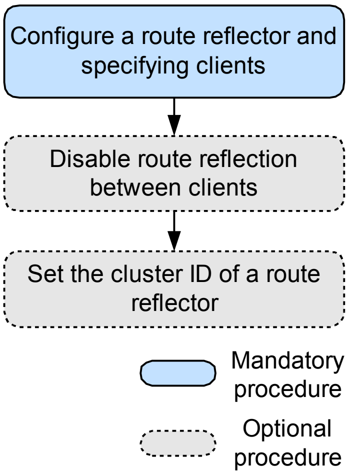
Pre-configuration Tasks

Before configuring a BGP route reflector, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedures

Figure 8-4 Flowchart for configuring a BGP route reflector


Configure a route reflector and specifying clients
Disable route reflection between clients
Set the cluster ID of a route reflector
Mandatory procedure
Optional procedure

8.14.1 Configuring a Route Reflector and Specifying Clients

RRs can reflect routes between clients, and clients do no need to establish IBGP connections.

Context

In an AS, one router functions as a route reflector (RR), the other router function as clients. IBGP connections are established between the RR and clients. The RR and its clients form a cluster. The RR transmits (or reflects) routes between clients, and clients do not need to establish IBGP connections.

An RR is easy to configure because it needs to be configured only on the router that functions as a reflector, and clients do not need to know that they are clients.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
peer { ipv4-address | group-name } reflect-client
```

An RR and its clients are configured.

The router where the **peer reflect-client** command is run serves as the RR and certain peers are specified as clients.

NOTE

The configurations of RRs and clients in an address family are valid only in this address family and cannot be inherited by other address families. Therefore, it is recommended to configure RRs and clients in the required address family.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.14.2 (Optional) Disabling Route Reflection Between Clients

If the clients of a route reflector are fully connected, you need to disable route reflection between clients to reduce the cost.

Context

On some networks, if fully meshed IBGP connections have been established between clients, they can directly exchange routing information. Therefore, route reflection between clients is unnecessary, which also occupies bandwidth. In this case, you can disable route reflection between clients to reduce the network cost.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
undo reflect between-clients
```

Route reflection is disabled between clients.

By default, route reflection between clients is enabled.

If the clients of the route reflector are fully connected, you can use the **undo reflect between-clients** command to disable route reflection between the clients. The cost is thus reduced. This command is applicable to only the route reflector.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.14.3 (Optional) Setting the Cluster ID of a Route Reflector

When there are multiple route reflectors in a cluster, you need to configure the same cluster ID for all the route reflectors in this cluster to avoid routing loops.

Context

To enhance the network reliability and avoid a single node fault, sometimes you need to configure more than one route reflector in a cluster. In this case, you need to set the same cluster ID for all the route reflectors in the same cluster. This can reduce the number of routes to be received by each route reflector and save the memory.

NOTE

To ensure that a client can learn the routes reflected by a route reflector, the cluster ID of the route reflector must be different from the router ID of the client. If they are the same, the client discards the received routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
reflector cluster-id cluster-id
```

The cluster ID of a route reflector is set.

When there are multiple route reflectors in a cluster, you need to run the command to configure the same *cluster-id* for all the route reflectors in this cluster.

The **reflector cluster-id** command can be configured on only route reflectors.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.14.4 (Optional) Preventing BGP Routes from Being Added into the IP Routing Table

Disabling BGP route delivery to the IP routing table on an RR can prevent traffic from being forwarded by the RR, improving route advertisement efficiency.

Context

Usually, BGP routes are delivered to the IP routing table on the router to guide traffic forwarding. If the router does not need to forward traffic, disable BGP route delivery to the IP routing table on the router.

BGP route delivery to the IP routing table is generally disabled on RRs. An RR transmits routes and forwards traffic within an AS. If the RR is connected to many clients and non-clients, the route transmission task will consume a lot of CPU resources of the RR and cause the RR unable to implement traffic forwarding. To improve the efficiency of route transmission, disable BGP route delivery to the IP routing table on the RR to make the RR dedicated to route transmission.

Perform the following steps on the router that is running BGP:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv4-family unicast
```

The IPv4 unicast address family view is displayed.

Step 4 Run:

```
bgp-rib-only [ route-policy route-policy-name ]
```

BGP route delivery to the IP routing table is disabled.

The routes preferred by BGP are delivered to the IP routing table by default.

If **route-policy** *route-policy-name* is configured in the **bgp-rib-only** command, routes matching the policy are not delivered to the IP routing table, and routes not matching the policy are delivered to the IP routing table, with the route attributes unchanged.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.14.5 Checking the Configuration

After a BGP route reflector is configured, you can check information about BGP routes and BGP peer groups.

Prerequisite

All configurations of a BGP route reflector are complete.

Procedure

- Run the **display bgp group** [*group-name*] command to check information about BGP peer groups.
- Run the **display bgp routing-table** [*network*] [*mask* | *mask-length*] [**longer-prefixes**] command to check information about the BGP routing table.

----End

Example

Run the **display bgp routing-table** [*network*] command, and you can view information about the BGP routing table. For example:

```
<HUAWEI> display bgp routing-table 9.1.1.0

BGP local router ID : 4.4.4.4
 Local AS number : 65010
 Paths:   1 available, 0 best, 0 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 10.1.4.1 (2.2.2.2)
 Route Duration: 00h00m14s
 Relay IP Nexthop: 0.0.0.0
 Relay IP Out-Interface:
 Original nexthop: 10.1.1.2
 Qos information : 0x0
 AS-path Nil, origin igp, MED 0, localpref 100, pref-val 0, internal, pre 255
 Originator:  1.1.1.1
```

```
Cluster list: 0.0.0.1
Not advertised to any peer yet
```

8.15 Configuring BFD for BGP

BFD for BGP provides a fast fault detection mechanism for BGP, and thus speeds up network convergence.

Applicable Environment

BFD can rapidly detect forwarding failures. By adopting BFD, a network can transmit voice services, video services, and VoD services with high QoS. This enables service providers to provide their customers with highly available and reliable VoIP and other real-time services.

BGP periodically sends Keepalive messages to the peer to detect the status of the peer. The detection, however, lasts more than one second. When the data transmission rate reaches the level of G bit/s, such a slow detection will cause a large amount of data to be lost. As a result, the requirement for high reliability of carrier-class networks cannot be met.

Therefore, BFD for BGP is introduced to fast detect faults on the links between BGP peers so that the network convergence speed is accelerated.

Pre-configuration Tasks

Before configuring BFD for BGP, complete the following tasks:

- Configuring parameters of the link layer protocol and IP addresses for interfaces to ensure that the link layer protocol on the interfaces is Up
- **Configuring Basic BGP Functions**

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

bfd

BFD is enabled on the local node.

Step 3 Run:

quit

Back to the system view.

Step 4 Run:

bgp *as-number*

The BGP view is displayed.

Step 5 Run:

peer { *group-name* | *ipv4-address* } **bfd enable**

BFD is configured on a peer or a peer group. The default values of BFD parameters are used to set up a BFD session.

Step 6 Run:

```
peer { group-name | ipv4-address } bfd { min-tx-interval min-tx-interval | min-rx-interval min-rx-interval | detect-multiplier multiplier } *
```

The values of the parameters used to set up a BFD session are set.

If BFD is configured on a peer group, the peers, on which the **peer bfd block** command is not run, of the peer group set up BFD sessions.

NOTE

- A BFD session is set up only when the BGP session is in the Established state.
- If BFD parameters of a peer are set, the BFD session is set up by using the BFD parameters of the peer.

Step 7 (Optional) Run:

```
peer ipv4-address bfd block
```

A peer is prevented from inheriting BFD of its peer group.

If a peer joins a group enabled with BFD, the peer inherits the BFD configuration of the group and creates a BFD session. If you do not want the peer to inherit BFD of its peer group, you can prevent the peer from inheriting BFD of its peer group.

NOTE

peer *ipv4-address* **bfd block** and **peer** *ipv4-address* **bfd enable** are mutually exclusive. After the **peer bfd block** command is run, the BFD session is automatically deleted.

Step 8 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

All configurations are complete.

- Run the **display bgp bfd session** { [**vpnv4 vpn-instance** *vpn-instance-name*] **peer** *ipv4-address* | **all** } command to check the information about BFD sessions set up between BGP routers.
- Run the **display bgp** [**vpnv4 vpn-instance** *vpn-instance-name*] **peer** [*ipv4-address*] [**verbose**] command to check the information about BGP peers.
- Run the **display bgp group** [*group-name*] command to check information about BGP peer groups.
- Run the **display bgp vpnv4** { **all** | **vpn-instance** *vpn-instance-name* } **group** [*group-name*] command to check the information about BGP peer groups.

Run the **display bgp bfd session all** command, you can view the BFD sessions set up between BGP routers. For example:

```
<HUAWEI> display bgp bfd session all
--------------------------------------------------------------------------------
  Local_Address      Peer_Address       LD/RD       Interface
  1.1.1.2             1.1.1.1            8192/8193    Unknown
  Tx-interval(ms)    Rx-interval(ms)    Multiplier  Session-State
```

```
      1000             1000             3            Up
```

8.16 Configuring BGP Auto FRR

BGP Auto FRR is applicable to the network topology with both active and standby links as a protection measure against faults over the link. BGP Auto FRR is suitable for the services that are sensitive to packet loss and delay.

Applicable Environment

BGP Auto FRR is applicable to IP services that are sensitive to packet loss and delay. After BGP Auto FRR is enabled, among the routes that have the same prefix and are learned from multiple peers, the optimal route is used as the primary link to forward packets, and the less optimal route is used as a backup link. When the primary link becomes faulty, the system rapidly responds to the notification that the BGP route becomes unreachable, and then switches traffic from the primary link to the backup link.

Pre-configuration Tasks

Before configuring BGP Auto FRR, complete the following tasks:

- Configuring static routes or enabling an IGP to ensure that IP routes between routers are reachable
- **Configuring Basic BGP Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
auto-frr
```

BGP Auto FRR is enabled for unicast routes.

By default, BGP Auto FRR is not enabled for unicast routes.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

All configurations are complete.

- Run the **display ip routing-table** [**vpn-instance** *vpn-instance-name*] [*ipv4-address*] [*mask* | *mask-length*] [**longer-match**] **verbose** command to check the backup forwarding information in the routing table.

Run the **display ip routing-table** command, and you can view information about the BGP routing table. For example:

```
<HUAWEI> display ip routing-table 4.4.4.4 32 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 4.4.4.4/32
     Protocol: BGP             Process ID: 0
   Preference: 255                   Cost: 80
      NextHop: 10.1.1.2         Neighbour: 10.1.1.2
        State: Active Adv             Age: 00h52m45s
          Tag: 0                 Priority: low
        Label: NULL               QoSInfo: 0x0
   IndirectID: 0x4
 RelayNextHop: 0.0.0.0          Interface: Pos1/0/0
     TunnelID: 0x0                  Flags:  D
    BkNextHop: 10.2.1.2       BkInterface: Pos2/0/0
      BkLabel: NULL           SecTunnelID: 0x0
 BkPETunnelID: 0x0       BkPESecTunnelID: 0x0
```

8.17 Configuring the BGP GR Helper

You can configure a device to function as a GR Helper to help a neighbor with the BGP GR process.

Applicable Environment

When BGP restarts, the peer relationship is re-established and traffic forwarding is interrupted. After Graceful Restart (GR) is enabled, traffic interruption can be avoided.

To avoid the forwarding interruption due to the BGP restart, you need to enable BGP GR and set up a BGP session that has the GR capability between the GR restarter and its peers.

NOTE

By default, the system supports only the GR helper.

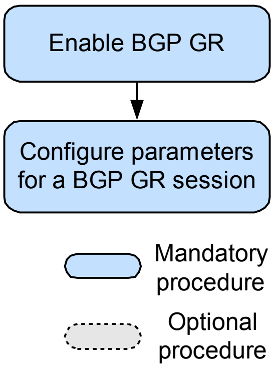
Pre-configuration Tasks

Before configuring the BGP GR helper, complete the following task:

- Configuring Basic BGP Functions

Configuration Procedures

Figure 8-5 Flowchart of configuring the BGP GR helper



8.17.1 Enabling BGP GR

Enabling or disabling GR may delete and reestablish all sessions and instances.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
graceful-restart
```

BGP GR is enabled.

By default, BGP GR is disabled.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

8.17.2 Configuring Parameters for a BGP GR Session

Changing the restart period leads to the re-establishment of the BGP peer relationship.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

bgp *as-number*

The BGP view is displayed.

Step 3 Run:

graceful-restart timer wait-for-rib *timer*

The period during which the restarting speaker and receiving speaker wait for End-Of-RIB messages is set.

By default, the period for waiting for End-Of-RIB messages is 600 seconds.

NOTE

> You can adjust parameter values of a BGP GR session as required. Generally, the default values of parameters are recommended.

Step 4 Run:

commit

The configuration is committed.

----End

8.17.3 Checking the Configuration

After the BGP GR helper is configured, you can check the BGP GR status.

Prerequisite

All BGP GR helper configurations are complete.

Procedure

- Run the **display bgp peer verbose** command to check the BGP GR status.

----End

Example

Run the **display bgp peer verbose** command, and you can view the BGP GR status. For example:

```
<HUAWEI> display bgp peer 2.2.2.9 verbose

         BGP Peer is 2.2.2.9,  remote AS 200,
         Type: EBGP link
         BGP version 4, Remote router ID 0.0.0.0

Group ID : 0
Peer Local Interface Name: Pos1/0/0
Local Ifnet Tunnel: 0xb0010000
         BGP current state: Established, Up for 20h21m17s
         BGP current event: KATimerExpired
         BGP last state: OpenConfirm
         BGP Peer Up count: 3
         Received total routes: 0
         Received active routes total: 0
         Advertised total routes: 0
         Port:  Local - 179      Remote - 54446
         Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Received  : Active Hold Time: 180 sec
         Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
```

```
        Peer optional capabilities:
        Peer supports bgp multi-protocol extension
        Peer supports bgp route refresh capability
        Peer supports bgp 4-byte-as capability
        Graceful Restart Capability: advertised
        Address family IPv4 Unicast: advertised and received

Received: Total 76 messages
                 Update messages                0
                 Open messages                  5
                 KeepAlive messages             71
                 Notification messages          0
                 Refresh messages               0

Sent: Total 91 messages
                 Update messages                0
                 Open messages                  10
                 KeepAlive messages             77
                 Notification messages          4
                 Refresh messages               0
Last keepalive received: 2009-03-30 09:14:14
Minimum route advertisement interval is 30 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Listen-only has been configured
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

8.18 Improving the Security of a BGP Network

To improve the security of a BGP network, you can configure BGP authentication and GTSM on the BGP network.

Applicable Environment

By performing authentication for BGP peer connections, you can improve the security of a BGP network.

- BGP authentication

  BGP uses TCP as the transport layer protocol. To enhance BGP security, you can perform authentication when a TCP connection is set up. During the authentication, BGP packets are not authenticated. Instead, the authentication is dedicated to the TCP connection and is performed by TCP. If the authentication fails, the TCP connection cannot be established.

- BGP GTSM

  The GTSM mechanism protects a router by checking whether the TTL value in an IP packet header is within a pre-defined range to enhance the system security.

NOTE

GTSM supports only unicast addresses. Therefore, GTSM must be configured on all the routers configured with routing protocols.

Pre-configuration Tasks

Before improving the security of a BGP network, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

8.18.1 Configuring BGP Authentication

BGP authentication is dedicated to TCP connections and is performed by TCP. If BGP authentication fails, the TCP connection cannot be established.

Context

BGP authentication includes:

- MD5 authentication

  BGP uses TCP as the transmission protocol, and BGP considers a packet valid as long as the source address, destination address, source port, destination port, and TCP sequence number of the packet are correct. Most parameters in a packet can be easily obtained by attackers. Therefore, to protect BGP from attacks, you can use MD5 authentication of TCP between BGP peers to reduce the possibility of attack.

  To prevent the MD5 password set on the BGP peers from being decrypted, you need to update the MD5 password periodically.

Procedure

- Configure MD5 authentication.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     peer { ipv4-address | group-name } password { cipher | simple } password
     ```
     The MD5 authentication password is set.

     In MD5 authentication of BGP, you only need to set MD5 authentication passwords for TCP connections, and the authentication is performed by TCP. If the authentication fails, the TCP connections cannot be established.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.

----End

8.18.2 Checking the Configuration

After configuring BGP security, you can view the authentication information about BGP peers.

Prerequisite

The configurations of BGP security are complete.

Procedure

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to view the authentication information about BGP peers.

----End

Example

# Run the **display bgp peer** [*ipv4-address*] **verbose** command to view the authentication information about BGP peers.

```
<HUAWEI> display bgp peer verbose
BGP Peer is 10.1.1.2,  remote AS 100
         Type: IBGP link
         BGP version 4, Remote router ID 10.1.1.2

  Group ID : 1
         BGP current state: Established, Up for 00h00m39s
         BGP current event: RecvUpdate
         BGP last state: Established
         BGP Peer Up count: 3
         Port: Local - 179        Remote - 30404
         Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Received  : Active Hold Time: 180 sec
         Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Peer optional capabilities:
         Peer supports bgp multi-protocol extension
         Peer supports bgp route refresh capability
         Peer supports bgp 4-byte-as capability
         Address family IPv4 Unicast: advertised and received
 Received: Total 229 messages
                  Update messages                5
                  Open messages                  3
                  KeepAlive messages             221
                  Notification messages          0
                  Refresh messages               0
 Sent: Total 236 messages
                  Update messages                5
                  Open messages                  4
                  KeepAlive messages             225
                  Notification messages          2
                  Refresh messages               0
 Authentication type configured: MD5
 Last keepalive received: 2010-09-20 14:41:10
 Minimum route advertisement interval is 15 seconds
 Optional capabilities:
 Route refresh capability has been enabled
 4-byte-as capability has been enabled
 Peer Preferred Value: 0
 Routing policy configured:
 No routing policy is configured
```

8.19 Configuring BGP Extensions

Configuring BGP extensions enables BGP to provide routing information for multiple routing protocols.

Applicable Environment

The section does not describe the commands that are associated with specific applications and used in the MP-BGP address family view in details.

For the BGP configurations in the IPv6 address family view, see chapter "BGP4+ Configuration."

For the BGP configurations in the BGP-IPv4 address family view and the BGP-VPN instance address family view, see the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN*.

For the applications of MP-BGP on multicast networks, see chapter "MBGP Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Multicast*.

Pre-configuration Tasks

Before configuring BGP extensions, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedures

None.

8.20 Maintaining BGP

Maintaining BGP involves resetting BGP connections, clearing BGP statistics, and debugging BGP.

8.20.1 Resetting BGP Connections

Resetting a BGP connection will interrupt the peer relationship.

Context

CAUTION

The BGP peer relationship between routers is interrupted after you reset BGP connections with the **reset bgp** command. So, confirm the action before you use the command.

When the BGP routing policy on the device that does not support the router-fresh capability changes, you need to reset BGP connections to make the change take effect. To reset BGP connections, run the following reset commands in the user view.

Procedure

- After confirming that all BGP connections need to be reset, run the **reset bgp all** command.
- After confirming that BGP connections between specified ASs need to be reset, run the **reset bgp** *as-number* command.
- After confirming that BGP connections with specified peers need to be reset, run the **reset bgp** *ipv4-address* command.
- After confirming that all EBGP connections need to be reset, run the **reset bgp external** command.

- After confirming that BGP connections with specified peer groups need to be reset, run the **reset bgp group** *group-name* command.
- After confirming that all IBGP connections need to be reset, run the **reset bgp internal** command.

----End

8.20.2 Clearing BGP Statistics

This section describes how to clear the statistics about flapping routes and dampened routes.

Context

CAUTION

BGP statistics cannot be restored after being cleared. Therefore, confirm the action before you run the command.

Procedure

- After confirming that the statistics about flapping routes need to be cleared, run the **reset bgp flap-info** [**regexp** *as-path-regexp* | **as-path-filter** *as-path-filter-number* | *ipv4-address* [*mask* | *mask-length*]] command in the user view.
- After confirming that the statistics about flapping routes of specified peers need to be cleared, run the **reset bgp** *ipv4-address* **flap-info** command in the user view.
- After confirming that the statistics about dampened routes need to be cleared and dampened routes are released, run the **reset bgp dampening** [*ipv4-address* [*mask* | *mask-length*]] command in the user view.

----End

8.21 Configuration Examples

BGP configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

8.21.1 Example for Configuring Basic BGP Functions

Before building BGP networks, you need to configure basic BGP functions.

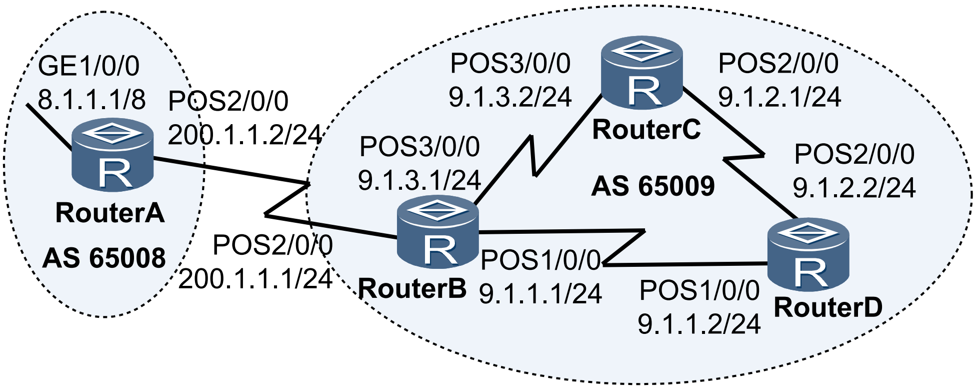
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-6**, all routers are BGP routers; an EBGP connection is established between Router A and Router B; IBGP fully meshed connections are established between Router B, Router C, and Router D.

Figure 8-6 Networking diagram of configuring basic BGP functions



Configuration Notes

When configuring basic BGP functions, pay attention to the following:

- When establishing the peer relationship, if the specified IP address of the peer is a loopback interface address or a sub-interface address, you need to run the **peer connect-interface** command on the two ends of the peer relationship to ensure that the two ends are correctly connected.
- If there is no directly connected physical link between EBGP peers, you must run the **peer ebgp-max-hop** command to allow EBGP peers to establish TCP connections through multiple hops.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish IBGP connections between Router B, Router C, and Router D.
2. Establish an EBGP connection between Router A and Router B.
3. Advertise routes through the **network** command on Router A, and then check the routing tables of Router A, Router B, and Router C.
4. Configure BGP on Router B to import direct routes, and then check the routing tables of Router A and Router C.

Data Preparation

To complete the configuration, you need the following data:

- Router ID and AS number of Router A
- Router IDs and AS numbers of Router B, Router C, and Router D

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure OSPF.

# Configure Router B.

```
[~RouterB] ospf 1
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 9.1.3.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] quit
```

# Configure Router C.

```
[~RouterC] ospf 1
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.2.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.3.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 3.3.3.3 0.0.0.0
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
[~RouterC-ospf-1] quit
```

# Configure Router D.

```
[~RouterD] ospf 1
[~RouterD-ospf-1] area 0
[~RouterD-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.0] network 9.1.2.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.0] network 4.4.4.4 0.0.0.0
[~RouterD-ospf-1-area-0.0.0.0] commit
[~RouterD-ospf-1-area-0.0.0.0] quit
[~RouterD-ospf-1] quit
```

Step 3 Configure IBGP connections.

# Configure Router B.

```
[~RouterB] bgp 65009
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 3.3.3.3 as-number 65009
[~RouterB-bgp] peer 4.4.4.4 as-number 65009
[~RouterB-bgp] peer 3.3.3.3 connect-interface LoopBack0
[~RouterB-bgp] peer 4.4.4.4 connect-interface LoopBack0
[~RouterB-bgp] commit
```

# Configure Router C.

```
[~RouterC] bgp 65009
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 2.2.2.2 as-number 65009
[~RouterC-bgp] peer 4.4.4.4 as-number 65009
[~RouterC-bgp] peer 2.2.2.2 connect-interface LoopBack0
[~RouterC-bgp] peer 4.4.4.4 connect-interface LoopBack0
[~RouterC-bgp] commit
```

# Configure Router D.

```
[~RouterD] bgp 65009
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 2.2.2.2 as-number 65009
[~RouterD-bgp] peer 3.3.3.3 as-number 65009
[~RouterD-bgp] peer 2.2.2.2 connect-interface LoopBack0
[~RouterD-bgp] peer 3.3.3.3 connect-interface LoopBack0
```

```
[~RouterD-bgp] commit
```

Step 4 Configure an EBGP connection.

# Configure Router A.

```
[~RouterA] bgp 65008
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.1 as-number 65009
[~RouterA-bgp] commit
```

# Configure Router B.

```
[~RouterB-bgp] peer 200.1.1.2 as-number 65008
[~RouterB-bgp] commit
```

Check the status of BGP connections.

```
[~RouterB] display bgp peer
 BGP local router ID : 2.2.2.2
 Local AS number : 65009
 Total number of peers : 3                 Peers in established state : 3
  Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  3.3.3.3         4 65009        5        5     0 00:44:58 Established       0
  4.4.4.4         4 65009        4        4     0 00:40:54 Established       0
  200.1.1.2       4 65008        3        3     0 00:44:03 Established       0
```

You can view that Router B has established BGP connections with other routers, and the connection status is Established.

Step 5 Configure Router A to advertise the route to 8.0.0.0/8.

# Configure Router A to advertise the route.

```
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] network 8.0.0.0 255.0.0.0
[~RouterA-bgp-af-ipv4] commit
```

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 1
      Network           NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   8.0.0.0           0.0.0.0         0                     0      i
```

# Check the routing table of Router B.

```
[~RouterB] display bgp routing-table
 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 1
      Network           NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   8.0.0.0           200.1.1.2       0                     0      65008i
```

# Check the routing table of Router C.

```
[~RouterC] display bgp routing-table
 BGP Local router ID is 3.3.3.3
 Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 1
      Network           NextHop        MED        LocPrf    PrefVal Path/Ogn
```

```
   i  8.0.0.0            200.1.1.2       0          100        0      65008i
```

NOTE

In the routing table, you can view that Router C has learned the route to 8.0.0.0 in AS 65008, but the next hop 200.1.1.2 is unreachable. Therefore, this route is invalid.

Step 6 Configure BGP to import direct routes.

# Configure Router B.

```
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] import-route direct
[~RouterB-bgp-af-ipv4] commit
```

# Check the BGP routing table of Router A.

```
[~RouterA] display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 8
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   2.2.2.2/32         200.1.1.1       0                     0      65009?
 *>   8.0.0.0            0.0.0.0         0                     0      i
 *>   9.1.1.0/24         200.1.1.1       0                     0      65009?
 *>   9.1.1.2/32         200.1.1.1       0                     0      65009?
 *>   9.1.3.0/24         200.1.1.1       0                     0      65009?
 *>   9.1.3.2/32         200.1.1.1       0                     0      65009?
 *    200.1.1.0          200.1.1.1       0                     0      65009?
 *    200.1.1.2/32       200.1.1.1       0                     0      65009?
```

# Check the BGP routing table of Router C.

```
[~RouterC] display bgp routing-table
 BGP Local router ID is 3.3.3.3
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 8
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

   i  2.2.2.2/32         2.2.2.2         0          100        0      ?
 *>i  8.0.0.0            200.1.1.2       0          100        0      65008i
 *>i  9.1.1.0/24         2.2.2.2         0          100        0      ?
 *>i  9.1.1.2/32         2.2.2.2         0          100        0      ?
 * i  9.1.3.0/24         2.2.2.2         0          100        0      ?
 * i  9.1.3.2/32         2.2.2.2         0          100        0      ?
 *>i  200.1.1.0          2.2.2.2         0          100        0      ?
 *>i  200.1.1.2/32       2.2.2.2         0          100        0      ?
```

You can view that the route to 8.0.0.0 becomes valid, and the next hop is the address of Router A.

# Verify the configuration by using the **ping** command.

```
[~RouterC] ping 8.1.1.1
  PING 8.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 8.1.1.1: bytes=56 Sequence=1 ttl=254 time=31 ms
    Reply from 8.1.1.1: bytes=56 Sequence=2 ttl=254 time=47 ms
    Reply from 8.1.1.1: bytes=56 Sequence=3 ttl=254 time=31 ms
    Reply from 8.1.1.1: bytes=56 Sequence=4 ttl=254 time=16 ms
    Reply from 8.1.1.1: bytes=56 Sequence=5 ttl=254 time=31 ms
  --- 8.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
```

```
0.00% packet loss
round-trip min/avg/max = 16/31/47 ms
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 8.1.1.1 255.0.0.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
bgp 65008
 router-id 1.1.1.1
 peer 200.1.1.1 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
  network 8.0.0.0 255.0.0.0
  peer 200.1.1.1 enable
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.3.1 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 65009
 router-id 2.2.2.2
 peer 3.3.3.3 as-number 65009
 peer 3.3.3.3 connect-interface LoopBack0
 peer 4.4.4.4 as-number 65009
 peer 4.4.4.4 connect-interface LoopBack0
 peer 200.1.1.2 as-number 65008
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 3.3.3.3 enable
  peer 4.4.4.4 enable
  peer 200.1.1.2 enable
#
```

```
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.2.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.3.2 255.255.255.0
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 65009
 router-id 3.3.3.3
 peer 2.2.2.2 as-number 65009
 peer 2.2.2.2 connect-interface LoopBack0
 peer 4.4.4.4 as-number 65009
 peer 4.4.4.4 connect-interface LoopBack0
#
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.2 enable
  peer 4.4.4.4 enable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 9.1.2.0 0.0.0.255
  network 9.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.2.2 255.255.255.0
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 65009
 router-id 4.4.4.4
 peer 2.2.2.2 as-number 65009
 peer 2.2.2.2 connect-interface LoopBack0
 peer 3.3.3.3 as-number 65009
 peer 3.3.3.3 connect-interface LoopBack0
 #
 ipv4-family unicast
```

```
  undo synchronization
  peer 2.2.2.2 enable
  peer 3.3.3.3 enable
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.2.0 0.0.0.255
#
return
```

8.21.2 Example for Configuring BGP to Advertise Default Routes

By controlling the advertising of default routes, you can specify traffic from a specific path to enter ASs.

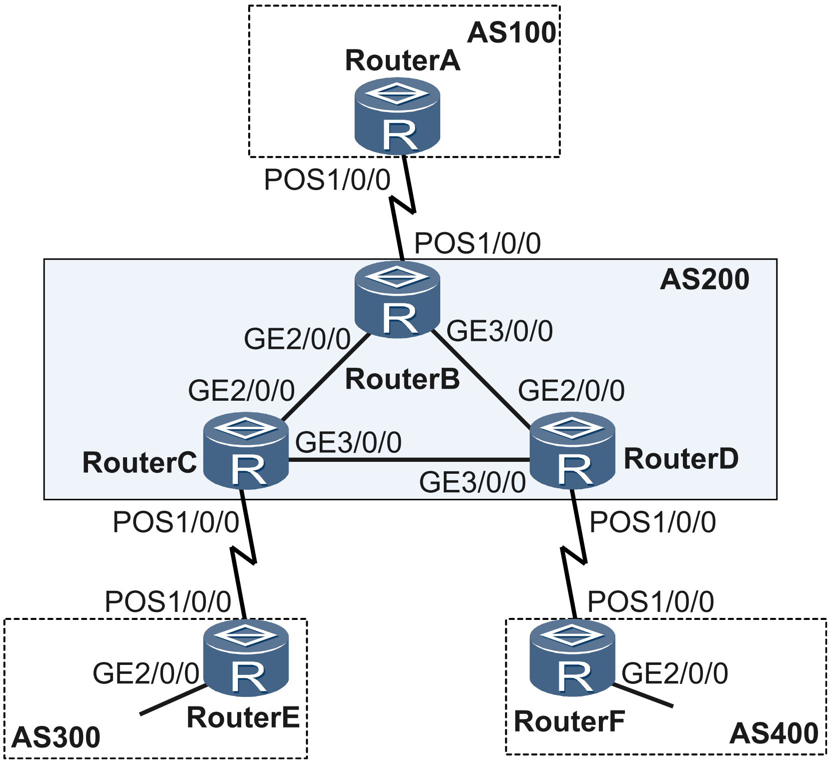
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-7**, all routers are BGP routers. To ensure that the outgoing traffic of AS 200 is forwarded through Router E and Router F, EBGP connections are established between Router A and Router B, between Router C and Router E, and between Router D and Router F; IBGP connections are established between Router B and Router C, and between Router B and Router D.

Figure 8-7 Networking diagram of configuring BGP to advertise default routes



Device	Interface	IP Address	Device	Interface	IP Address
Router A	POS 1/0/0	200.1.1.1/24	Router D	POS 1/0/0	200.1.3.2/24
	Loopback0	1.1.1.1/32		GE 2/0/0	9.1.3.1/24
Router B	POS 1/0/0	200.1.1.2/24		GE 3/0/0	9.1.2.2/24
	GE 2/0/0	9.1.1.1/24		Loopback0	4.4.4.4/32
	GE 3/0/0	9.1.3.2/24	Router E	POS 1/0/0	200.1.2.1/24
	Loopback0	2.2.2.2/32		GE 2/0/0	10.1.1.1/24
Router C	POS 1/0/0	200.1.2.2/24		Loopback0	5.5.5.5/32
	GE 2/0/0	9.1.1.2/24	Router F	POS 1/0/0	200.1.3.1/24
	GE 3/0/0	9.1.2.1/24		GE 2/0/0	11.1.1.1/24
	Loopback0	3.3.3.3/32		Loopback0	6.6.6.6/32

Configuration Notes

When configuring BGP to advertise default routes, pay attention to the following:

- Default routes have two functions. They can represent all network routes. For example, in a stub AS, instead of advertising all network routes, you can use only a default route to forward traffic destined outside the stub AS. In addition, they can represent all routes except specific routes; for example, they can be used in the multi-home load balancing scenario.
- When establishing the peer relationship, if the specified IP address of the peer is a loopback interface address or a sub-interface address, you need to run the **peer connect-interface**

command on the two ends of the peer relationship to ensure that the two ends are correctly connected.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on Router B, Router C, and Router D.
2. Establish EBGP connections between Router A and Router B, between Router C and Router E, and between Router D and Router F.
3. Establish IBGP connections between Router B and Router C, and between Router B and Router D.
4. Configure an inbound routing policy on Router C to allow only default routes to be accepted.
5. Configure an inbound routing policy on Router D to allow default routes and all specific routes to be accepted, and then set Local_Pref values for the accepted default routes.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, Router C, Router D, Router E, and Router F
- Names of the inbound routing policies to be configured on Router C and Router D
- Local_Pref values to be set for the accepted default routes on Router D

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure OSPF.

# Configure Router B.

```
[~RouterB] ospf 1
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 9.1.3.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] quit
```

# Configure Router C.

```
[~RouterC] ospf 1
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.2.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 3.3.3.3 0.0.0.0
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
[~RouterC-ospf-1] quit
```

# Configure Router D.

```
[~RouterD] ospf 1
[~RouterD-ospf-1] area 0
[~RouterD-ospf-1-area-0.0.0.0] network 9.1.2.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.0] network 9.1.3.0 0.0.0.255
```

```
[~RouterD-ospf-1-area-0.0.0.0] network 4.4.4.4 0.0.0.0
[~RouterD-ospf-1-area-0.0.0.0] commit
[~RouterD-ospf-1-area-0.0.0.0] quit
[~RouterD-ospf-1] quit
```

Step 3 Configure BGP connections.

# Configure Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.2 as-number 200
[~RouterA-bgp] commit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 200
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.1.1 as-number 100
[~RouterB-bgp] network 200.1.1.0 24
[~RouterB-bgp] peer 3.3.3.3 as-number 200
[~RouterB-bgp] peer 3.3.3.3 connect-interface LoopBack0
[~RouterB-bgp] peer 4.4.4.4 as-number 200
[~RouterB-bgp] peer 4.4.4.4 connect-interface LoopBack0
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 200
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.2.1 as-number 300
[~RouterC-bgp] network 200.1.2.0 24
[~RouterC-bgp] peer 2.2.2.2 as-number 200
[~RouterC-bgp] peer 2.2.2.2 connect-interface LoopBack0
[~RouterC-bgp] commit
[~RouterC-bgp] quit
```

# Configure Router D.

```
[~RouterD] bgp 200
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 200.1.3.1 as-number 400
[~RouterD-bgp] network 200.1.3.0 24
[~RouterD-bgp] peer 2.2.2.2 as-number 200
[~RouterD-bgp] peer 2.2.2.2 connect-interface LoopBack0
[~RouterD-bgp] commit
[~RouterD-bgp] quit
```

# Configure Router E.

```
[~RouterE] bgp 300
[~RouterE-bgp] router-id 5.5.5.5
[~RouterE-bgp] peer 200.1.2.2 as-number 200
[~RouterE-bgp] network 10.1.1.0 24
[~RouterE-bgp] commit
[~RouterE-bgp] quit
```

# Configure Router F.

```
[~RouterF] bgp 400
[~RouterF-bgp] router-id 6.6.6.6
[~RouterF-bgp] peer 200.1.3.2 as-number 200
[~RouterF-bgp] network 11.1.1.0 24
[~RouterF-bgp] commit
[~RouterF-bgp] quit
```

Step 4 # Configure Router E and Router F to advertise default routes.

# Configure Router E to advertise default routes.

```
[~RouterE-bgp] ipv4-family unicast
[~RouterE-bgp-af-ipv4] peer 200.1.2.2 default-route-advertise
[~RouterE-bgp-af-ipv4] commit
```

# Configure Router F to advertise default routes.

```
[~RouterF-bgp] ipv4-family unicast
[~RouterF-bgp-af-ipv4] peer 200.1.3.2 default-route-advertise
[~RouterF-bgp-af-ipv4] commit
```

# Check the routing table of Router B.

```
[~RouterB] display bgp routing-table

 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 7
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>i  0.0.0.0            200.1.2.1       0          100        0      300i
 * i                     200.1.3.1       0          100        0      400i
 *>i  10.1.1.0/24        200.1.2.1       0          100        0      300i
 *>i  11.1.1.0/24        200.1.3.1       0          100        0      400i
 *>   200.1.1.0          0.0.0.0         0                     0      i
 *>i  200.1.2.0          3.3.3.3         0          100        0      i
 *>i  200.1.3.0          4.4.4.4         0          100        0      i
```

In the routing table, you can view the default routes and all specific routes to AS 300 and AS 400.

Step 5 Configure inbound routing policies.

# Configure an IP prefix list named **default** on Router C to allow only default routes to be accepted.

```
[~RouterC] ip ip-prefix default permit 0.0.0.0 0
[~RouterC] commit
[~RouterC] bgp 200
[~RouterC-bgp] peer 200.1.2.1 ip-prefix default import
[~RouterC-bgp] commit
```

# Configure a route-policy named **set-default-low** on Router D to allow default routes and all specific routes to be accepted, and then set Local_Pref values for the accepted default routes.

```
[~RouterD] ip as-path-filter 10 permit ^(400_)+$
[~RouterD] ip as-path-filter 10 permit ^(400_)+_[0-9]+$
[~RouterD] ip ip-prefix default permit 0.0.0.0 0
[~RouterD] route-policy set-default-low permit node 10
[~RouterD-route-policy] if-match ip-prefix default
[~RouterD-route-policy] apply local-preference 80
[~RouterD-route-policy] quit
[~RouterD] route-policy set-default-low permit node 20
[~RouterD-route-policy] quit
[~RouterD] commit
[~RouterD] bgp 200
[~RouterD-bgp] peer 200.1.3.1 as-path-filter 10 import
[~RouterD-bgp] peer 200.1.3.1 route-policy set-default-low import
[~RouterD-bgp] commit
```

# Check the routing table of Router B.

```
[~RouterB] display bgp routing-table

 BGP Local router ID is 2.2.2.2
```

```
Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 6
    Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

*>i  0.0.0.0            200.1.2.1       0          100        0      300i
* i                     200.1.3.1       0          80         0      400i
*>i  11.1.1.0/24        200.1.3.1       0          100        0      400i
*>   200.1.1.0          0.0.0.0         0                     0      i
*>i  200.1.2.0          3.3.3.3         0          100        0      i
*>i  200.1.3.0          4.4.4.4         0          100        0      i
```

In the routing table, you can view that Router B receives only the default routes to AS 300 and the default routes and all specific routes to AS 400 and sets the Local_Pref of the accepted default routes destined for AS 400 to 80.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
 interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 200.1.1.2 as-number 200
 #
 ipv4-family unicast
  peer 200.1.1.2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 9.1.3.2 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 200
 peer 3.3.3.3 as-number 200
 peer 3.3.3.3 connect-interface LoopBack0
 peer 4.4.4.4 as-number 200
 peer 4.4.4.4 connect-interface LoopBack0
 peer 200.1.1.1 as-number 100
 #
 ipv4-family unicast
  network 200.1.1.0 255.255.255.0
  peer 3.3.3.3 enable
  peer 4.4.4.4 enable
  peer 200.1.1.1 enable
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.3.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 9.1.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 9.1.2.1 255.255.255.0
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 200
 peer 2.2.2.2 as-number 200
 peer 2.2.2.2 connect-interface LoopBack0
 peer 200.1.2.1 as-number 300
 #
 ipv4-family unicast
  network 200.1.2.0 255.255.255.0
  peer 2.2.2.2 enable
  peer 200.1.2.1 enable
  peer 200.1.2.1 ip-prefix default import
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.2.0 0.0.0.255
#
ip ip-prefix default index 10 permit 0.0.0.0 0
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 9.1.2.2 255.255.255.0
#
```

```
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 9.1.3.1 255.255.255.0
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 200
 peer 2.2.2.2 as-number 200
 peer 2.2.2.2 connect-interface LoopBack0
 peer 200.1.3.1 as-number 400
 #
 ipv4-family unicast
  network 200.1.3.0 255.255.255.0
  peer 2.2.2.2 enable
  peer 200.1.3.1 enable
  peer 200.1.3.1 as-path-filter 10 import
  peer 200.1.3.1 route-policy set-default-low import
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 9.1.2.0 0.0.0.255
  network 9.1.3.0 0.0.0.255
#
route-policy set-default-low permit node 10
 if-match ip-prefix default
 apply local-preference 80
#
route-policy set-default-low permit node 20
#
ip ip-prefix default index 10 permit 0.0.0.0 0
#
 ip as-path-filter 10 permit ^(400_)+$
 ip as-path-filter 10 permit ^(400_)+_[0-9]+$
#
return
```

- Configuration file of Router E

```
#
sysname RouterE
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 10.1.1.1 255.255.255.0
#
 interface LoopBack0
 ip address 5.5.5.5 255.255.255.255
#
bgp 300
 peer 200.1.2.2 as-number 200
 #
 ipv4-family unicast
  network 10.1.1.0 255.255.255.0
  peer 200.1.2.2 enable
  peer 200.1.2.2 default-route-advertise
#
return
```

- Configuration file of Router F

```
#
sysname RouterF
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 11.1.1.1 255.255.255.0
#
 interface LoopBack0
 ip address 6.6.6.6 255.255.255.255
#
bgp 400
 peer 200.1.3.2 as-number 200
 #
 ipv4-family unicast
  network 11.1.1.0 255.255.255.0
  peer 200.1.3.2 enable
  peer 200.1.3.2 default-route-advertise
#
return
```

8.21.3 Example for Configuring BGP to Interact with an IGP

During route exchange, configuring BGP route summarization can simplify the routing table.

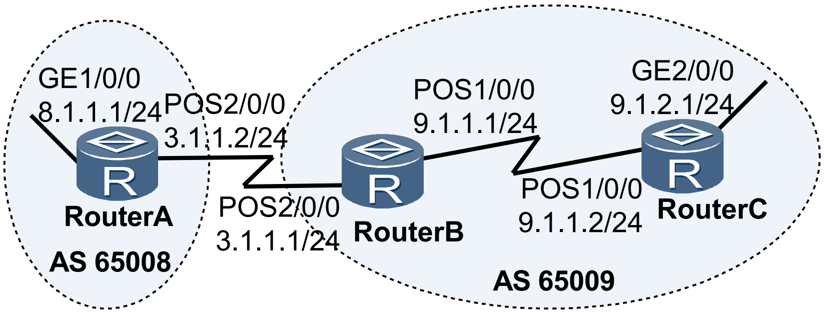
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-8**, OSPF is used as an IGP in AS 65009; an EBGP connection is established between Router A and Router B; Router C is a non-BGP router in the AS.

Figure 8-8 Networking diagram of configuring BGP to interact with an IGP



Configuration Notes

None.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on Router B and Router C.
2. Establish an EBGP connection between Router A and Router B.
3. Configure BGP and OSPF to import routes from each other on Router B and then check the routes.
4. Configure BGP route summarization on Router B to simplify the BGP routing table.

Data Preparation

To complete the configuration, you need the following data:

- Router ID and AS number of Router A
- Router IDs and AS numbers of Router B and Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure OSPF.

# Configure Router B.

```
[~RouterB] ospf 1
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] quit
```

# Configure Router C.

```
[~RouterC] ospf 1
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.1.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] network 9.1.2.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
[~RouterC-ospf-1] quit
```

Step 3 Configure an EBGP connection.

# Configure Router A.

```
[~RouterA] bgp 65008
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 3.1.1.1 as-number 65009
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] network 8.1.1.0 255.255.255.0
[~RouterA-bgp-af-ipv4] commit
```

# Configure Router B.

```
[~RouterB] bgp 65009
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 3.1.1.2 as-number 65008
[~RouterB-bgp] commit
```

Step 4 Configure BGP to interact with an IGP.

# Configure BGP to import OSPF routes on Router B.

```
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] import-route ospf 1
[~RouterB-bgp-af-ipv4] commit
```

```
[~RouterB-bgp-af-ipv4] quit
[~RouterB-bgp] quit
```

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 3
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   8.1.1.0/24         0.0.0.0         0                     0      i
 *>   9.1.1.0/24         3.1.1.1         1                     0      65009?
 *>   9.1.2.0/24         3.1.1.1         2                     0      65009?
```

# Configure OSPF to import BGP routes on Router B.

```
[~RouterB] ospf
[~RouterB-ospf-1] import-route bgp
[~RouterB-ospf-1] commit
[~RouterB-ospf-1] quit
```

# Check the routing table of Router C.

```
[~RouterC] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables : _public_
         Destinations : 7        Routes : 7

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        8.1.1.0/24  O_ASE  150  1             D  9.1.1.1         Pos1/0/0
        9.1.1.0/24  Direct 0    0             D  9.1.1.2         Pos1/0/0
        9.1.1.2/32  Direct 0    0             D  127.0.0.1       InLoopBack0
        9.1.2.0/24  Direct 0    0             D  9.1.2.1
GigabitEthernet2/0/0
        9.1.2.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
```

Step 5 Configure automatic route summarization.

# Configure Router B.

```
[~RouterB] bgp 65009
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] summary automatic
[~RouterB-bgp-af-ipv4] commit
```

# Check the BGP routing table of Router A.

```
[~RouterA] display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 2
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   8.1.1.0/24         0.0.0.0         0                     0      i
 *>   9.0.0.0            3.1.1.1                               0      65009?
```

# Verify the configuration by using the **ping** command.

```
[~RouterA] ping -a 8.1.1.1 9.1.2.1
  PING 9.1.2.1: 56  data bytes, press CTRL_C to break
    Reply from 9.1.2.1: bytes=56 Sequence=1 ttl=254 time=15 ms
    Reply from 9.1.2.1: bytes=56 Sequence=2 ttl=254 time=31 ms
    Reply from 9.1.2.1: bytes=56 Sequence=3 ttl=254 time=47 ms
    Reply from 9.1.2.1: bytes=56 Sequence=4 ttl=254 time=46 ms
    Reply from 9.1.2.1: bytes=56 Sequence=5 ttl=254 time=47 ms
  --- 9.1.2.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 15/37/47 ms
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 8.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 3.1.1.2 255.255.255.0
#
bgp 65008
 router-id 1.1.1.1
 peer 3.1.1.1 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
  network 8.1.1.0 255.255.255.0
  peer 3.1.1.1 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 3.1.1.1 255.255.255.0
#
bgp 65009
 router-id 2.2.2.2
 peer 3.1.1.2 as-number 65008
 #
 ipv4-family unicast
  undo synchronization
  summary automatic
  import-route ospf 1
  peer 3.1.1.2 enable
#
ospf 1
 import-route bgp
 area 0.0.0.0
  network 9.1.1.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 9.1.2.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 9.1.1.2 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.2.0 0.0.0.255
#
return
```

8.21.4 Example for Configuring the MED Attribute to Control Route Selection

By setting the MED attribute, you can flexibly control BGP route selection.

Networking Requirements

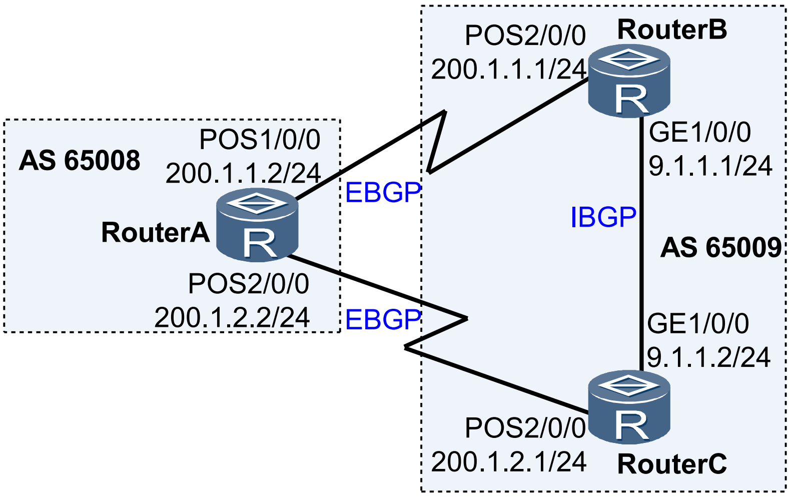
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-9**, all routers are configured with BGP; Router A resides in AS 65008; Router B and Router C reside in AS 65009. EBGP connections are established between Router A and Router B, and between Router A and Router C. An IBGP connection is established between Router B and Router C.

It is required to control route selection by configuring the MED attribute.

Figure 8-9 Networking diagram of configuring the MED attribute to control route selection



Configuration Notes

None.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Router A and Router B, and between Router A and Router C, and establish an IBGP connection between Router B and Router C.
2. Set the MED value for the route sent from Router B to Router A by using a routing policy.

Data Preparation

To complete the configuration, you need the following data:

- Router ID and AS number of Router A
- Router IDs and AS numbers of Router B and Router C and default MED value on Router B

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure BGP connections.

# Configure Router A.

```
[~RouterA] bgp 65008
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.1 as-number 65009
[~RouterA-bgp] peer 200.1.2.1 as-number 65009
[~RouterA-bgp] commit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 65009
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.1.2 as-number 65008
[~RouterB-bgp] peer 9.1.1.2 as-number 65009
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] network 9.1.1.0 255.255.255.0
[~RouterB-bgp-af-ipv4] commit
[~RouterB-bgp-af-ipv4] quit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 65009
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.2.2 as-number 65008
[~RouterC-bgp] peer 9.1.1.1 as-number 65009
[~RouterC-bgp] ipv4-family unicast
[~RouterC-bgp-af-ipv4] network 9.1.1.0 255.255.255.0
[~RouterC-bgp-af-ipv4] commit
[~RouterC-bgp-af-ipv4] quit
[~RouterC-bgp] quit
```

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table 9.1.1.0 24

 BGP local router ID : 1.1.1.1
 Local AS number : 65008
 Paths:   2 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.1.1 (2.2.2.2)
 Route Duration: 0d00h00m56s
 Direct Out-interface: Pos1/0/0
 Original nexthop: 200.1.1.1
 Qos information : 0x0
 AS-path 65009, origin igp, MED 0, pref-val 0, valid, external, best, select, pre
255
 Advertised to such 2 peers:
    200.1.1.1
    200.1.2.1

 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.2.1 (3.3.3.3)
 Route Duration: 0d00h00m06s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 AS-path 65009, origin igp, MED 0, pref-val 0, valid, external, pre 255, not
selected for router ID
 Not advertised to any peers yet
```

In the routing table, you can view that there are two valid routes to the destination 9.1.1.0/24. The route with the next hop being 200.1.1.1 is the optimal route because the router ID of Router B is smaller.

Step 3 Configure the MED attribute.

# Set the MED value for the route sent from Router B to Router A by using a routing policy.

```
[~RouterB] route-policy 10 permit node 10
[~RouterB-route-policy] apply cost 100
[~RouterB-route-policy] commit
[~RouterB-route-policy] quit
[~RouterB] bgp 65009
[~RouterB-bgp] peer 200.1.1.2 route-policy 10 export
[~RouterB-bgp] commit
```

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table 9.1.1.0 24

BGP local router ID : 1.1.1.1
 Local AS number : 65008
 Paths:   2 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.2.1 (3.3.3.3)
 Route Duration: 0d00h07m45s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 AS-path 65009, origin igp, MED 0, pref-val 0, valid, external, best, select, pre
255
 Advertised to such 2 peers:
    200.1.1.1
    200.1.2.1

 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.1.1 (2.2.2.2)
 Route Duration: 0d00h00m08s
 Direct Out-interface: Pos1/0/0
 Original nexthop: 200.1.1.1
 Qos information : 0x0
 AS-path 65009, origin igp, MED 100, pref-val 0, valid, external, pre 255, not s
```

```
elected for MED
 Not advertised to any peers yet
```

In the routing table, you can view that the MED value of the next hop 200.1.2.1 (Router B) is 100, and the MED value of the next hop 200.1.1.1 is 0. Therefore, the route with the smaller MED value is selected.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
bgp 65008
 router-id 1.1.1.1
 peer 200.1.1.1 as-number 65009
 peer 200.1.2.1 as-number 65009
 #
 ipv4-family unicast
  peer 200.1.1.1 enable
  peer 200.1.2.1 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
bgp 65009
 router-id 2.2.2.2
 peer 9.1.1.2 as-number 65009
 peer 200.1.1.2 as-number 65008
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 9.1.1.2 enable
  peer 200.1.1.2 enable
  peer 200.1.1.2 route-policy 10 export
#
route-policy 10 permit node 10
 apply cost 100
#
return
```

- Configuration file of Router C

```
#
```

```
sysname RouterC
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 9.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
bgp 65009
 router-id 3.3.3.3
 peer 9.1.1.1 as-number 65009
 peer 200.1.2.2 as-number 65008
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 9.1.1.1 enable
  peer 200.1.2.2 enable
#
return
```

8.21.5 Example for Configuring BGP Route Filtering

By applying route filtering policies, you can improve network performance.

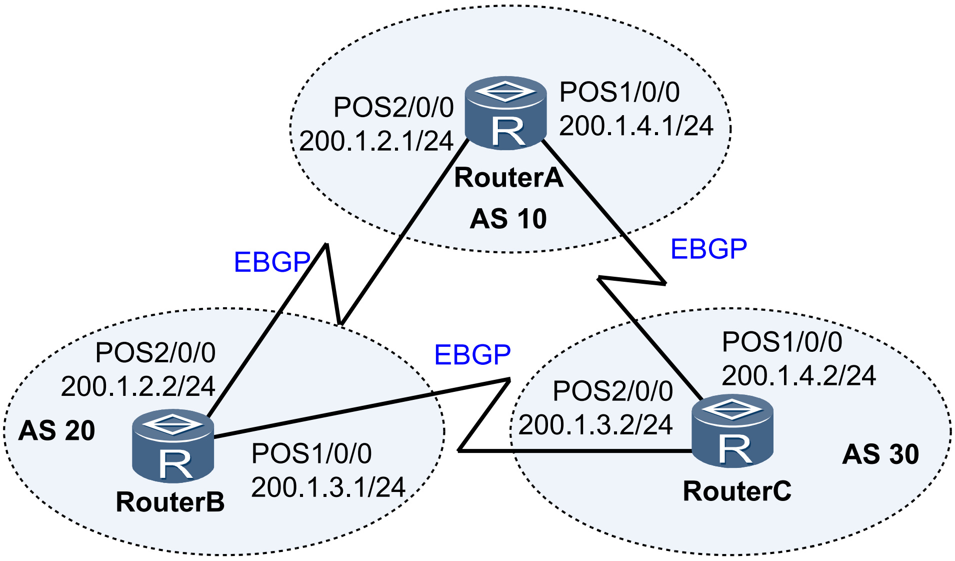
Networking Requirements

⚠ **CAUTION**

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-10**, EBGP connections are established between Router A, Router B, and Router C. The AS_Path filter is configured on Router B to ensure that AS 20 does not advertise routes of AS 30 to AS 10 or advertise routes of AS 10 to AS 30.

Figure 8-10 Networking diagram of configuring BGP route filtering



Configuration Notes

When configuring BGP route filtering, pay attention to the following:

- The relationship between multiple filtering rules of the same filter is OR.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Router A and Router B, between Router B and Router C, and between Router C and Router A, and then import direct routes.
2. Configure the AS_Path filter on Router B and then apply its filtering rules.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, and Router C
- Number of the AS_Path filter

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure EBGP connections.

# Configure Router A.

```
[~RouterA] bgp 10
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.4.2 as-number 30
[~RouterA-bgp] peer 200.1.2.2 as-number 20
[~RouterA-bgp] import-route direct
```

```
[~RouterA-bgp] commit
```

# Configure Router B.

```
[~RouterB] bgp 20
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.2.1 as-number 10
[~RouterB-bgp] peer 200.1.3.2 as-number 30
[~RouterB-bgp] import-route direct
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 30
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.3.1 as-number 20
[~RouterC-bgp] peer 200.1.4.1 as-number 10
[~RouterC-bgp] import-route direct
[~RouterC-bgp] commit
[~RouterC-bgp] quit
```

# Check the routing table advertised by Router B. Take the routing table advertised by Router B to Router C as an example. You can view that Router B advertises the routes destined for the network segment between Router A and Router C.

```
<RouterB> display bgp routing-table peer 200.1.3.2 advertised-routes
 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 9
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          0.0.0.0         0                     0      ?
 *>   200.1.2.1/32       0.0.0.0         0                     0      ?
 *>   200.1.2.2/32       200.1.2.1       0                     0      10?
 *>   200.1.3.0          0.0.0.0         0                     0      ?
 *>   200.1.3.1/32       200.1.3.2       0                     0      30?
 *>   200.1.3.2/32       0.0.0.0         0                     0      ?
 *>   200.1.4.0          200.1.2.1       0                     0      10?
 *>   200.1.4.1/32       200.1.3.2       0                     0      30?
 *>   200.1.4.2/32       200.1.2.1       0                     0      10?
```

Check the routing table of Router C, and you can view that Router C learns the two routes advertised by Router B.

```
<RouterC> display bgp routing-table
 BGP Local router ID is 3.3.3.3
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 18
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          200.1.4.1       0                     0      10?
 *                       200.1.3.1       0                     0      20?
 *>   200.1.2.1/32       200.1.3.1       0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.2.2/32       200.1.4.1       0                     0      10?
 *                       200.1.3.1                             0      20 10?
 *>   200.1.3.0          0.0.0.0         0                     0      ?
 *                       200.1.3.1       0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.3.1/32       0.0.0.0         0                     0      ?
 *>   200.1.3.2/32       200.1.3.1       0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.4.0          0.0.0.0        0                     0      ?
 *                       200.1.4.1      0                     0      10?
 *                       200.1.3.1                            0      20 10?
 *>   200.1.4.1/32       0.0.0.0        0                     0      ?
 *>   200.1.4.2/32       200.1.4.1      0                     0      10?
 *                       200.1.3.1                            0      20 10?
```

Step 3 Configure the AS_Path filter on Router B and then apply the filter on the outbound interface of Router B.

# Create AS_Path filter 1 to deny the routes carrying AS 30. The regular expression "_30_" indicates any AS list that contains AS 30 and "*" matches any character.

```
[~RouterB] ip as-path-filter 1 deny _30_
[~RouterB] ip as-path-filter 1 permit .*
[~RouterB] commit
```

# Create AS_Path filter 2 to deny the routes carrying AS 10.

```
[~RouterB] ip as-path-filter 2 deny _10_
[~RouterB] ip as-path-filter 2 permit .*
[~RouterB] commit
```

# Apply the AS_Path filter on two outbound interfaces of Router B.

```
[~RouterB] bgp 20
[~RouterB-bgp] peer 200.1.2.1 as-path-filter 1 export
[~RouterB-bgp] peer 200.1.3.2 as-path-filter 2 export
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

Step 4 Verify the configuration.

# Check the routing table advertised by Router B, and you can view that the advertised routes to the network segment between Router A and Router C do not exist. Take the routes advertised by Router B to Router C as an example.

```
<RouterB> display bgp routing-table peer 200.1.3.2 advertised-routes

 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 6
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          0.0.0.0        0                     0      ?
 *>   200.1.2.1/32       0.0.0.0        0                     0      ?
 *>   200.1.3.0          0.0.0.0        0                     0      ?
 *>   200.1.3.1/32       200.1.3.2      0                     0      30?
 *>   200.1.3.2/32       0.0.0.0        0                     0      ?
 *>   200.1.4.1/32       200.1.3.2      0                     0      30?
```

Similarly, the BGP routing table of Router C does not have these routes.

```
<RouterC> display bgp routing-table

 BGP Local router ID is 3.3.3.3
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 15
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          200.1.4.1      0                     0      10?
 *                       200.1.3.1      0                     0      20?
 *>   200.1.2.1/32       200.1.3.1      0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.2.2/32       200.1.4.1       0                     0      10?
 *>   200.1.3.0          0.0.0.0         0                     0      ?
 *                       200.1.3.1       0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.3.1/32       0.0.0.0         0                     0      ?
 *>   200.1.3.2/32       200.1.3.1       0                     0      20?
 *                       200.1.4.1                             0      10 20?
 *>   200.1.4.0          0.0.0.0         0                     0      ?
 *                       200.1.4.1       0                     0      10?
 *>   200.1.4.1/32       0.0.0.0         0                     0      ?
 *>   200.1.4.2/32       200.1.4.1       0                     0      10?
```

Check the routing table advertised by Router B, and you can view that the advertised routes to the network segment between Router A and Router C do not exist. Take the routes advertised by Router B to Router A as an example.

```
<RouterB> display bgp routing-table peer 200.1.2.1 advertised-routes
 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 4
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          0.0.0.0         0                     0      ?
 *>   200.1.2.1/32       0.0.0.0         0                     0      ?
 *>   200.1.2.2/32       200.1.2.1       0                     0      10?
 *>   200.1.3.0          0.0.0.0         0                     0      ?
 *>   200.1.3.2/32       0.0.0.0         0                     0      ?
 *>   200.1.4.0          200.1.2.1       0                     0      10?
 *>   200.1.4.2/32       200.1.2.1       0                     0      10?
```

Similarly, the BGP routing table of Router A does not have these routes.

```
<RouterA> display bgp routing-table

 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 14
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          0.0.0.0         0                     0      ?
 *                       200.1.2.2       0                     0      20?
 *>   200.1.2.1/32       200.1.2.2       0                     0      20?
 *                       200.1.4.2                             0      30 20?
 *>   200.1.2.2/32       0.0.0.0         0                     0      ?
 *>   200.1.3.0          200.1.2.2       0                     0      20?
 *                       200.1.4.2       0                     0      30?
 *>   200.1.3.1/32       200.1.4.2       0                     0      30?
 *>   200.1.3.2/32       200.1.2.2       0                     0      20?
 *                       200.1.4.2                             0      30 20?
 *>   200.1.4.0          0.0.0.0         0                     0      ?
 *                       200.1.4.2       0                     0      30?
 *>   200.1.4.1/32       200.1.4.2       0                     0      30?
 *>   200.1.4.2/32       0.0.0.0         0                     0      ?
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
```

```
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.4.1 255.255.255.0
#
bgp 10
 router-id 1.1.1.1
 peer 200.1.2.2 as-number 20
 peer 200.1.4.2 as-number 30
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 200.1.2.2 enable
  peer 200.1.4.2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
bgp 20
 router-id 2.2.2.2
 peer 200.1.2.1 as-number 10
 peer 200.1.3.2 as-number 30
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 200.1.2.1 enable
  peer 200.1.2.1 as-path-filter 1 export
  peer 200.1.3.2 enable
  peer 200.1.3.2 as-path-filter 2 export
#
 ip as-path-filter 1 deny _30_
 ip as-path-filter 1 permit .*
 ip as-path-filter 2 deny _10_
 ip as-path-filter 2 permit .*
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.4.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.2 255.255.255.0
#
bgp 30
 router-id 3.3.3.3
 peer 200.1.3.1 as-number 20
 peer 200.1.4.1 as-number 10
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 200.1.3.1 enable
  peer 200.1.4.1 enable
#
return
```

8.21.6 Example for Configuring Prefix-based BGP ORF

After prefix-based BGP ORF is configured, on-demand route advertisement can be implemented.

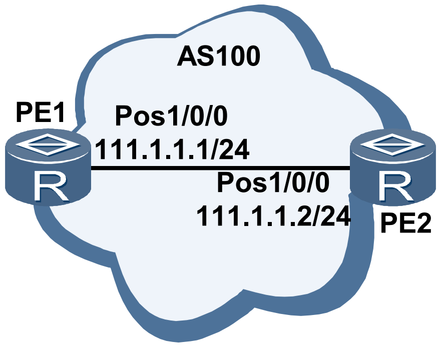
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-11**, PE1 and PE2 are in AS 100; PE1 requires PE2 to send only the routes matching the inbound policy of PE1.

Figure 8-11 Networking diagram of configuring prefix-based BGP ORF



Configuration Roadmap

The configuration roadmap is as follows:

1. Establish an IPv4 unicast peer relationship between PE1 and PE2.
2. Apply prefix-based inbound policy to PE1 and configure PE1 to import routes from PE2. Then, check the sent routes and received routes.

3. Check the sent and received routes after configuring prefix-based BGP ORF.

Data Preparation

To complete the configuration, you need the following data:

- Router ID and AS number of PE1 (in this example, the router ID of PE1 is 1.1.1.1, and the AS number of PE1 is 100)
- Router ID and AS number of PE2 (in this example, the router ID of PE2 is 2.2.2.2, and the AS number of PE2 is 100)

Procedure

Step 1 Establish an IPv4 unicast peer relationship between PE1 and PE2.

# Configure PE1.

```
<HUAWEI> system-view
[~HUAWEI] sysname PE1
[~PE1] interface pos 1/0/0
[~PE1-Pos1/0/0] ip address 111.1.1.1 255.255.255.0
[~PE1-Pos1/0/0] quit
[~PE1] bgp 100
[~PE1-bgp] peer 111.1.1.2 as-number 100
[~PE1-bgp] commit
[~PE1-bgp] quit
```

# Configure PE2.

```
<HUAWEI> system-view
[~HUAWEI] sysname PE2
[~PE2] interface pos 1/0/0
[~PE2-Pos1/0/0] ip address 111.1.1.2 255.255.255.0
[~PE2-Pos1/0/0] quit
[~PE2] bgp 100
[~PE2-bgp] peer 111.1.1.1 as-number 100
[~PE2-bgp] commit
[~PE2-bgp] quit
```

Step 2 Apply the prefix-based inbound policy on PE1.

# Configure PE1.

```
[~PE1] ip ip-prefix 1 permit 4.4.4.0 24 greater-equal 32
[~PE1] bgp 100
[~PE1-bgp] peer 111.1.1.2 ip-prefix 1 import
[~PE1-bgp] commit
[~PE1-bgp] quit
```

# Configure PE2.

```
[~PE2] ip route-static 3.3.3.3 255.255.255.255 NULL0
[~PE2] ip route-static 4.4.4.4 255.255.255.255 NULL0
[~PE2] ip route-static 5.5.5.5 255.255.255.255 NULL0
[~PE2] bgp 100
[~PE2-bgp] import-route static
[~PE2-bgp] commit
[~PE2-bgp] quit
```

# Check the routes sent by PE2 to PE1.

```
[~PE2] display bgp routing peer 111.1.1.1 advertised-routes

 BGP Local router ID is 111.1.1.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 3
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   3.3.3.3/32         0.0.0.0         0                     0      ?
 *>   4.4.4.4/32         0.0.0.0         0                     0      ?
 *>   5.5.5.5/32         0.0.0.0         0                     0      ?
```

# Check the routes received by PE1 from PE2.

```
[~PE1] display bgp routing-table peer 111.1.1.2 received-routes

 BGP Local router ID is 111.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>i  4.4.4.4/32         111.1.1.2       0          100        0      ?
```

When prefix-based BGP ORF is not enabled, PE2 sends routes 3.3.3.3, 4.4.4.4, and 5.5.5.5 to PE1. Because the prefix-based inbound policy is applied on PE1, PE1 receives only route 4.4.4.4.

Step 3 Enable prefix-based BGP ORF.

# Enable prefix-based BGP ORF on PE1.

```
[~PE1] bgp 100
[~PE1-bgp] peer 111.1.1.2 capability-advertise orf ip-prefix both
[~PE1-bgp] commit
[~PE1-bgp] quit
```

# Enable prefix-based BGP ORF on PE2.

```
[~PE2] bgp 100
[~PE2-bgp] peer 111.1.1.1 capability-advertise orf ip-prefix both
[~PE2-bgp] commit
[~PE2-bgp] quit
```

Step 4 Verify the configuration.

# Check the negotiation of prefix-based BGP ORF.

```
<PE1> display bgp peer 111.1.1.2 verbose

        BGP Peer is 111.1.1.2,  remote AS 100
        Type: IBGP link
        BGP version 4, Remote router ID 111.1.1.2
        Update-group ID: 2
        BGP current state: Established, Up for 00h01m22s
        BGP current event: KATimerExpired
        BGP last state: OpenConfirm
        BGP Peer Up count: 8
        Received total routes: 1
        Received active routes total: 1
        Advertised total routes: 0
        Port:  Local - 54845    Remote - 179
        Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
        Received  : Active Hold Time: 180 sec
        Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
        Peer optional capabilities:
        Peer supports bgp multi-protocol extension
        Peer supports bgp route refresh capability
        Peer supports bgp outbound route filter capability
        Support Address-Prefix: IPv4-UNC address-family, rfc-compatible, both
        Peer supports bgp 4-byte-as capability
        Address family IPv4 Unicast: advertised and received
 Received: Total 5 messages
                 Update messages                1
                 Open messages                  1
                 KeepAlive messages             2
                 Notification messages          0
                 Refresh messages               1
 Sent: Total 4 messages
                 Update messages                0
                 Open messages                  1
                 KeepAlive messages             2
                 Notification messages          0
                 Refresh messages               1
 Authentication type configured: None
 Last keepalive received: 2010/03/30 13:37:25 UTC-08:00
 Minimum route advertisement interval is 15 seconds
 Optional capabilities:
 Route refresh capability has been enabled
 Outbound route filter capability has been enabled
 Enable Address-Prefix: IPv4-UNC address-family, rfc-compatible, both
 4-byte-as capability has been enabled
 Peer Preferred Value: 0
 Routing policy configured:
 No import update filter list
 No export update filter list
 Import prefix list is: 1
 No export prefix list
 No import route policy
 No export route policy
 No import distribute policy
 No export distribute policy
```

# Check the routes sent by PE2 to PE1.

```
<PE2> display bgp routing peer 111.1.1.1 advertised-routes

 BGP Local router ID is 111.1.1.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   4.4.4.4/32         0.0.0.0         0                     0      ?
```

# Check the routes received by PE1 from PE2.

```
<PE1> display bgp routing-table peer 111.1.1.2 received-routes

 BGP Local router ID is 111.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>i  4.4.4.4/32         111.1.1.2       0          100        0      ?
```

After being enabled with prefix-based BGP ORF, PE2 sends only route 4.4.4.4 matching the inbound policy of PE1.

----End

Configuration Files

- Configuration file of PE1

```
#
sysname PE1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 111.1.1.1 255.255.255.0
#
bgp 100
 peer 111.1.1.2 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  peer 111.1.1.2 enable
  peer 111.1.1.2 ip-prefix 1 import
  peer 111.1.1.2 capability-advertise orf ip-prefix both
 #
#
route-policy 1 permit node 10
#
ip ip-prefix 1 index 10 permit 4.4.4.0 24 greater-equal 32 less-equal 32
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 111.1.1.2 255.255.255.0
#
bgp 100
 peer 111.1.1.1 as-number 100
#
 ipv4-family unicast
 undo synchronization
  import-route static
  peer 111.1.1.1 enable
  peer 111.1.1.1 capability-advertise orf ip-prefix both
#
ip route-static 3.3.3.3 255.255.255.255 NULL0
ip route-static 4.4.4.4 255.255.255.255 NULL0
ip route-static 5.5.5.5 255.255.255.255 NULL0
#
return
```

8.21.7 Example for Configuring BGP Load Balancing

By properly configuring load balancing, you can fully utilize network resources and thus reduce network congestion.

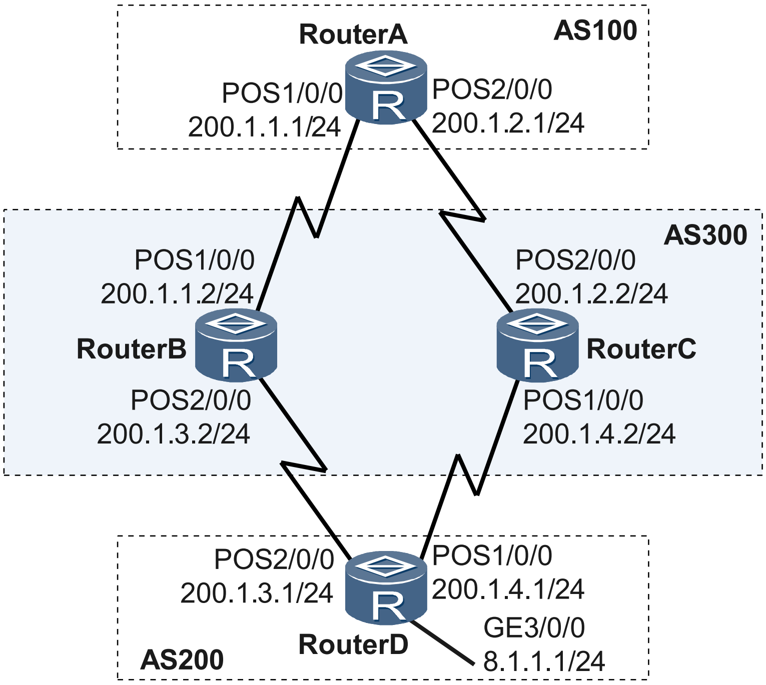
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-12**, all routers are configured with BGP; Router A resides in AS 100; Router B and Router C reside in AS 200; Router D resides in AS 300. EBGP connections are established between Router A and Router B, between Router A and Router C, between Router D and Router B, and between Router D and Router C. On Router A, there are two BGP routes to the destination 8.1.1.0/24. Traffic to the destination 8.1.1.0/24 can reach the destination through Router B or Router C. It is required to configure BGP load balancing to fully utilize network resources and thus reduce network congestion.

Figure 8-12 Networking diagram of configuring BGP load balancing



Configuration Notes

When configuring BGP load balancing, pay attention to the following:

- By setting BGP attributes, you can implement load balancing among routes. For example, you can ignore the comparison of MED values, route types, or IGP metrics. You can perform these configurations only when you can ensure that no routing loops occur, for example, in the scenario of L3VPN. This solution is not recommended on the public network, especially on the transit AS.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Router A and Router B, and between Router A and Router C, and establish an IBGP connection between Router B and Router C.
2. Establish EBGP connections between Router D and Router B, and between Router D and Router C.
3. Configure load balancing on Router A and then check routes.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, Router C, and Router D
- Number of routes for load balancing

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure BGP connections.

# Configure Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.2 as-number 300
[~RouterA-bgp] peer 200.1.2.2 as-number 300
[~RouterA-bgp] commit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 300
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.1.1 as-number 100
[~RouterB-bgp] peer 200.1.3.1 as-number 200
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 300
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.2.1 as-number 100
[~RouterC-bgp] peer 200.1.4.1 as-number 200
[~RouterC-bgp] commit
[~RouterC-bgp] quit
```

# Configure Router D.

```
[~RouterD] bgp 200
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 200.1.3.2 as-number 300
[~RouterD-bgp] peer 200.1.4.2 as-number 300
[~RouterD-bgp] ipv4-family unicast
[~RouterD-bgp-af-ipv4] network 8.1.1.0 255.255.255.0
[~RouterD-bgp-af-ipv4] commit
[~RouterD-bgp-af-ipv4] quit
[~RouterD-bgp] quit
```

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table 8.1.1.0 24

 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Paths : 2 available, 1 best, 1 select
 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.1.2 (2.2.2.2)
 Route Duration: 0d00h00m50s
 Direct Out-interface: Pos1/0/0
 Original nexthop: 200.1.1.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, best, select, pre 255
 Advertised to such 2 peers:
    200.1.1.2
    200.1.2.2

 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.2.2 (3.3.3.3)
 Route Duration: 0d00h00m51s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, pre 255, not selected
for router ID
 Not advertised to any peers yet
```

In the routing table, you can view that there are two valid routes from Router A to the destination 8.1.1.0/24. The route with the next hop being 200.1.1.2 is the optimal route because the router ID of Router B is smaller.

Step 3 Configure load balancing.

# Configure load balancing on Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] maximum load-balancing 2
[~RouterA-bgp-af-ipv4] commit
[~RouterA-bgp-af-ipv4] quit
[~RouterA-bgp] quit
```

Step 4 Verify the configuration.

# Check the routing table of Router A.

```
[~RouterA] display bgp routing-table 8.1.1.0 24

 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Paths : 2 available, 1 best, 2 select
 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.1.2 (2.2.2.2)
 Route Duration: 0d00h03m55s
 Direct Out-interface: Pos1/0/0
 Original nexthop: 200.1.1.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, best, select, pre 255
 Advertised to such 2 peers:
    200.1.1.2
    200.1.2.2

 BGP routing table entry information of 8.1.1.0/24:
 From: 200.1.2.2 (3.3.3.3)
 Route Duration: 0d00h03m56s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.2
 Qos information : 0x0
 AS-path 200 300, origin igp, pref-val 0, valid, external, select, pre 255, not
selected for router ID
 Not advertised to any peers yet
```

In the routing table, you can view that the BGP route to 8.1.1.0/24 has two next hops, 200.1.1.2 and 200.1.2.2, both of which are preferred.

----End

Configuration Files

- Configuration file of Router A

```
#
```

```
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 router-id 1.1.1.1
 peer 200.1.1.2 as-number 300
 peer 200.1.2.2 as-number 300
 #
 ipv4-family unicast
  maximum load-balancing 2
  peer 200.1.1.2 enable
  peer 200.1.2.2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
interface Pos2/0/0
 link-protocol ppp
 ip address 200.1.3.2 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 300
 router-id 2.2.2.2
 peer 200.1.1.1 as-number 100
 peer 200.1.3.1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.1.1 enable
  peer 200.1.3.1 enable
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.4.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
interface LoopBack0
```

```
 ip address 3.3.3.3 255.255.255.255
#
bgp 300
 router-id 3.3.3.3
 peer 200.1.2.1 as-number 100
 peer 200.1.4.1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.2.1 enable
  peer 200.1.4.1 enable
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.4.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 8.1.1.1 255.255.255.0
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 200
 router-id 4.4.4.4
 peer 200.1.3.2 as-number 300
 peer 200.1.4.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  network 8.1.1.0 255.255.255.0
  peer 200.1.3.2 enable
  peer 200.1.4.2 enable
#
return
```

8.21.8 Example for Configuring BGP Route Flap Dampening

Configuring BGP route flap dampening can improve network stability.

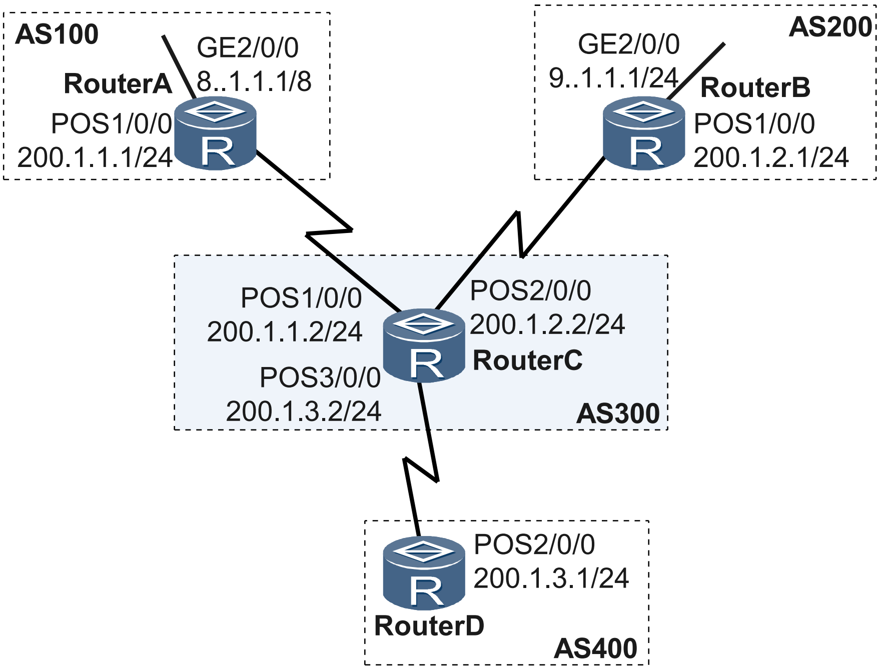
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-13**, all routers are configured with BGP; Router A resides in AS 100; Router B resides in AS 200; Router C resides in AS 300; Router D resides in AS 400. EBGP runs between Router C and Router A, between Router C and Router B, and between Router C and Router D. For routes from different EBGP neighbors, Router C applies different route flap dampening policies. It is required to configure BGP route flap dampening to suppress unstable routes and thus improve network stability.

Figure 8-13 Networking diagram of configuring BGP route flap dampening



Configuration Notes

When configuring BGP route flap dampening, pay attention to the following:

- BGP route flap dampening takes effect on only EBGP routes.
- For the routes with the shorter destination address mask, you need to set a shorter MaxSuppressTime.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Router A and Router C, between Router B and Router C, and between Router D and Router C.
2. Apply a route flap dampening policy to Router C and then check routes.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, Router C, and Router D
- Name of the route flap dampening policy to be applied to Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure BGP connections.

# Configure Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.2 as-number 300
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] network 8.0.0.0 255.0.0.0
[~RouterA-bgp-af-ipv4] commit
[~RouterA-bgp-af-ipv4] quit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 200
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.2.2 as-number 300
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] network 9.1.1.0 255.255.255.0
[~RouterB-bgp-af-ipv4] commit
[~RouterB-bgp-af-ipv4] quit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 300
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.1.1 as-number 100
[~RouterC-bgp] peer 200.1.2.1 as-number 200
[~RouterC-bgp] peer 200.1.3.1 as-number 400
[~RouterC-bgp] commit
[~RouterC-bgp] quit
```

# Configure Router D.

```
[~RouterD] bgp 400
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 200.1.3.2 as-number 300
[~RouterD-bgp] commit
[~RouterD-bgp] quit
```

# Check the BGP peer of Router C.

```
[~RouterC] display bgp peer

 BGP local router ID : 3.3.3.3
 Local AS number : 300
 Total number of peers : 3         Peers in established state : 3

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State
PrefRcv
  200.1.1.1       4         100        3        3     0 00:00:01 Established    0
  200.1.2.1       4         200        3        3     0 00:00:00 Established    0
  200.1.3.1       4         400        3        3     0 00:00:01 Established    0
```

You can view that the BGP connection status of Router C is **Established**.

Step 3 Configure BGP route flap dampening policies.

# Configure an IP prefix list named **prefix-a** on Router C to allow the routes with the prefix being 8.0.0.0/8 to pass the filtering.

```
[~RouterC] ip ip-prefix prefix-a index 10 permit 8.0.0.0 8
```

```
[~RouterC] commit
```

# Configure an IP prefix list named **prefix-b** on Router C to allow the routes with the prefix being 9.1.1.0/24 to pass the filtering.

```
[~RouterC] ip ip-prefix prefix-b index 20 permit 9.1.1.0 24
[~RouterC] commit
```

# Configure a route-policy named **dampen-policy** on Router C and then apply different route flap dampening policies to the routes with different prefix lengths.

```
[~RouterC] route-policy dampen-policy permit node 10
[~RouterC-route-policy] if-match ip-prefix prefix-a
[~RouterC-route-policy] apply dampening 10 1000 2000 5000
[~RouterC-route-policy] commit
[~RouterC-route-policy] quit
[~RouterC] route-policy dampen-policy permit node 20
[~RouterC-route-policy] if-match ip-prefix prefix-b
[~RouterC-route-policy] apply dampening 10 800 3000 10000
[~RouterC-route-policy] commit
[~RouterC-route-policy] quit
```

# Apply route flap dampening policies to the routes that flap.

```
[~RouterC] bgp 300
[~RouterC-bgp] ipv4-family unicast
[~RouterC-bgp-af-ipv4] dampening route-policy dampen-policy
[~RouterC-bgp-af-ipv4] commit
[~RouterC-bgp] quit
```

# Check the configured route flap dampening parameters on Router C.

```
[~RouterC] display bgp routing-table dampening parameter

 Maximum Suppress Time(in second) : 3973
 Ceiling Value                    : 16000
 Reuse Value                      : 750
 HalfLife Time(in  second)        : 900
 Suppress-Limit                   : 2000
 Route-policy                     : dampen-policy
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 8.1.1.1 255.0.0.0
#
bgp 100
 router-id 1.1.1.1
 peer 200.1.1.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  network 8.0.0.0 255.0.0.0
  peer 200.1.1.2 enable
#
```

```
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
bgp 200
 router-id 2.2.2.2
 peer 200.1.2.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 200.1.2.2 enable
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.2 255.255.255.0
#
bgp 300
 router-id 3.3.3.3
 peer 200.1.1.1 as-number 100
 peer 200.1.2.1 as-number 200
 peer 200.1.3.1 as-number 400
 #
 ipv4-family unicast
  undo synchronization
  dampening route-policy dampen-policy
  peer 200.1.1.1 enable
  peer 200.1.2.1 enable
  peer 200.1.3.1 enable
#
route-policy dampen-policy permit node 10
 if-match ip-prefix prefix-a
 apply dampening 10 1000 2000 5000
#
route-policy dampen-policy permit node 20
 if-match ip-prefix prefix-b
 apply dampening 10 800 3000 10000
#
ip ip-prefix prefix-a index 10 permit 8.0.0.0 8
#
ip ip-prefix prefix-b index 20 permit 9.1.1.0 24
#
```

```
return
```

- Configuration file of Router D

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.1 255.255.255.0
#
bgp 400
 router-id 4.4.4.4
 peer 200.1.3.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.3.2 enable
#
return
```

8.21.9 Example for Configuring the BGP Community Attribute

By setting the community attribute, you can flexibly control BGP route selection.

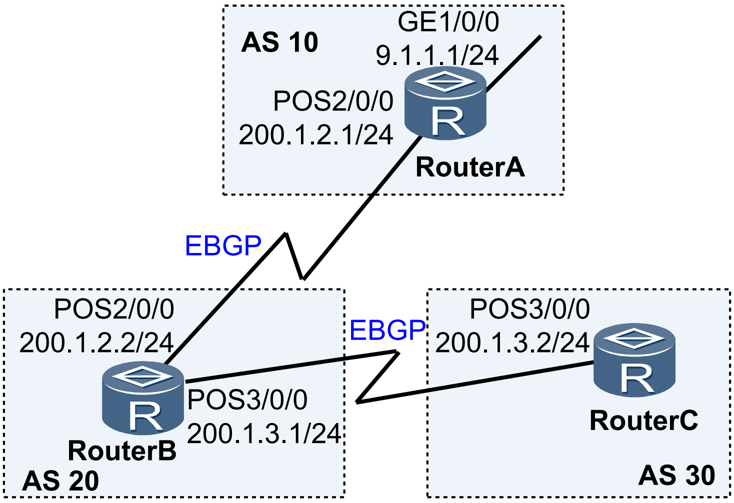
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-14**, EBGP connections are established between Router B and Router A, and between Router B and Router C. With the community No_Export attribute configured on Router A, the routes advertised from AS 10 to AS 20 are not advertised to other ASs by AS 20.

Figure 8-14 Networking diagram of configuring the BGP community attribute



Configuration Notes

None.

Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Router A andRouter B, and between Router B and Router C.
2. Configure a routing policy on Router A to advertise the community No_Export attribute.

Data Preparation

To complete the configuration, you need the following data:

- Router ID and AS number of Router A
- Router ID and AS number of Router B
- Router ID and AS number of Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure EBGP connections.

# Configure Router A.

```
[~RouterA] bgp 10
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.2.2 as-number 20
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] network 9.1.1.0 255.255.255.0
[~RouterA-bgp-af-ipv4] commit
[~RouterA-bgp-af-ipv4] quit
```

# Configure Router B.

```
[~RouterB] bgp 20
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.2.1 as-number 10
[~RouterB-bgp] peer 200.1.3.2 as-number 30
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] bgp 30
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 200.1.3.1 as-number 20
[~RouterC-bgp] commit
[~RouterC-bgp] quit
```

# Check the routing table of Router B.

```
[~RouterB] display bgp routing-table 9.1.1.0
BGP local router ID : 2.2.2.2
 Local AS number : 20
 Paths:   1 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.2.1 (1.1.1.1)
 Route Duration: 0d00h00m37s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 AS-path 10, origin igp, MED 0, pref-val 0, valid, external, best, select, pre 255
 Advertised to such 2 peers:
    200.1.2.1
    200.1.3.2
```

You can view that Router B advertises the received routes to Router C in AS 30.

# Check the routing table of Router C.

```
[~RouterC] display bgp routing-table
 BGP Local router ID is 3.3.3.3
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   9.1.1.0/24         200.1.3.1                            0      20 10i
```

In the routing table, you can view that Router C learns the route to 9.1.1.0/24 from Router B.

Step 3 Configure the BGP community attribute.

# Configure a routing policy on Router A to ensure that the routes advertised by Router A to Router B are not advertised by Router B to any other AS.

```
[~RouterA] route-policy comm_policy permit node 10
[~RouterA-route-policy] apply community no-export
[~RouterA-route-policy] commit
[~RouterA-route-policy] quit
```

# Apply the routing policy.

```
[~RouterA] bgp 10
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] peer 200.1.2.2 route-policy comm_policy export
[~RouterA-bgp-af-ipv4] peer 200.1.2.2 advertise-community
[~RouterA-bgp-af-ipv4] commit
```

# Check the routing table of Router B.

```
[~RouterB] display bgp routing-table 9.1.1.0
BGP local router ID : 2.2.2.2
 Local AS number : 20
 Paths:   1 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 200.1.2.1 (1.1.1.1)
 Route Duration: 0d00h00m12s
 Direct Out-interface: Pos2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 Community:no-export
 AS-path 10, origin igp, MED 0, pref-val 0, valid, external, best, select, pre 255
 Not advertised to any peers yet
```

In the BGP routing table of Router B, you can view the configured community attribute. Then, there is no route to 9.1.1.0/24 in the BGP routing table of Router C.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
```

```
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.1 255.255.255.0
#
bgp 10
 router-id 1.1.1.1
 peer 200.1.2.2 as-number 20
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 200.1.2.2 enable
  peer 200.1.2.2 route-policy comm_policy export
  peer 200.1.2.2 advertise-community
#
route-policy comm_policy permit node 10
 apply community no-export
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.2.2 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.1 255.255.255.0
#
bgp 20
 router-id 2.2.2.2
 peer 200.1.2.1 as-number 10
 peer 200.1.3.2 as-number 30
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.2.1 enable
  peer 200.1.3.2 enable
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 200.1.3.2 255.255.255.0
#
bgp 30
 router-id 3.3.3.3
 peer 200.1.3.1 as-number 20
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.3.1 enable
#
```

```
    return
```

8.21.10 Example for Configuring a BGP Route Reflector

Configuring a BGP route reflector can avoid establishing fully meshed connections between IBGP peers, thus simplifying the network.

Networking Requirements

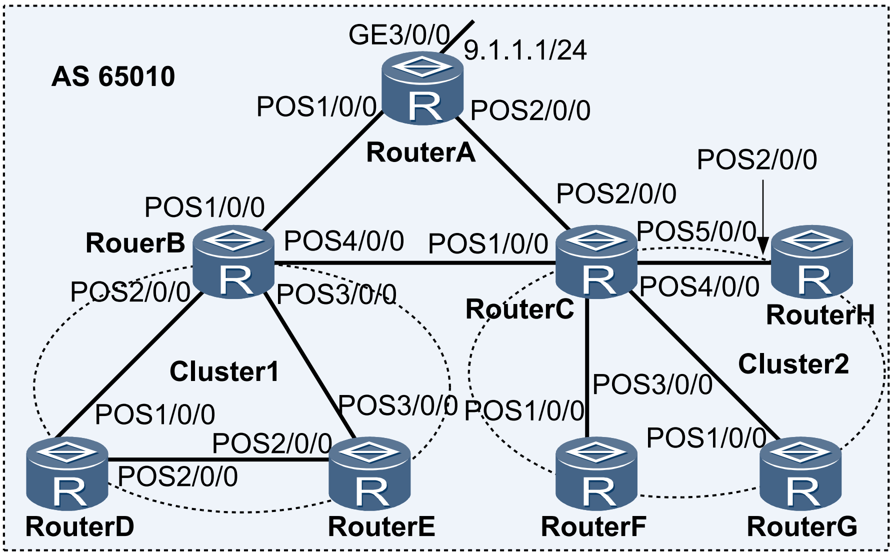
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-15**, Router A is a non-client; Router B is the route reflector (RR) of Cluster 1; Router D and Router E are two clients of Cluster 1. An IBGP connection is established between Router D and Router E, and thus route reflection is not required between they. Router C is the RR of Cluster 2; Router F, Router G, and Router H are the clients of Cluster 2.

It is required to configure peer groups to facilitate configuration and management.

Figure 8-15 Networking diagram of configuring a BGP RR



Device	Interface	IP Address	Device	Interface	IP Address
Router A	GE 3/0/0	9.1.1.1/24	Router C	POS 4/0/0	10.1.8.1/24
	POS 1/0/0	10.1.1.2/24		POS 5/0/0	10.1.9.1/24
	POS 2/0/0	10.1.3.2/24	Router D	POS 1/0/0	10.1.4.2/24
Router B	POS 1/0/0	10.1.1.1/24		POS 2/0/0	10.1.6.1/24
	POS 2/0/0	10.1.4.1/24	Router E	POS 2/0/0	10.1.6.2/24
	POS 3/0/0	10.1.5.1/24		POS 3/0/0	10.1.5.2/24

	POS 4/0/0	10.1.2.1/24	Router F	POS 1/0/0	10.1.7.2/24
Router C	POS 1/0/0	10.1.2.2/24	Router G	POS 1/0/0	10.1.8.2/24
	POS 2/0/0	10.1.3.1/24	Router H	POS 2/0/0	10.1.9.2/24
	POS 3/0/0	10.1.7.1/24			

Configuration Notes

When configuring a BGP RR, pay attention to the following:

- If a cluster has multiple RRs, you can use the **reflector cluster-id** command to set the same cluster ID for these RRs to prevent routing loops.
- When specifying a peer group, ensure that the peer group name is case sensitive.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IBGP connections between the clients and the RR, and between the non-clients and the RR.
2. Configure route reflection on Router B and Router C, specify the client, and check routes.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, Router C, Router D, Router E, Router F, Router G, and Router H
- Cluster ID of Router B

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure IBGP connections between the clients and the RR, and between the non-clients and the RR.

Step 3 Configure RRs.

# Configure Router B.

```
[~RouterB] bgp 65010
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] group in_rr internal
[~RouterB-bgp] peer 10.1.4.2 group in_rr
[~RouterB-bgp] peer 10.1.5.2 group in_rr
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] peer in_rr reflect-client
[~RouterB-bgp-af-ipv4] undo reflect between-clients
[~RouterB-bgp-af-ipv4] reflector cluster-id 1
[~RouterB-bgp-af-ipv4] commit
[~RouterB-bgp-af-ipv4] quit
```

# Configure Router C.

```
[~RouterC] bgp 65010
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] group in_rr internal
[~RouterC-bgp] peer 10.1.7.2 group in_rr
[~RouterC-bgp] peer 10.1.8.2 group in_rr
[~RouterC-bgp] peer 10.1.9.2 group in_rr
[~RouterC-bgp] ipv4-family unicast
[~RouterC-bgp-af-ipv4] peer in_rr reflect-client
[~RouterC-bgp-af-ipv4] reflector cluster-id 2
[~RouterC-bgp-af-ipv4] commit
[~RouterC-bgp-af-ipv4] quit
```

# Check the routing table of Router D.

```
[~RouterD] display bgp routing-table 9.1.1.0
BGP local router ID : 4.4.4.4
 Local AS number : 65010
 Paths:   1 available, 0 best, 0 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 10.1.4.1 (2.2.2.2)
 Route Duration: 00h00m14s
 Relay IP Nexthop: 0.0.0.0
 Relay IP Out-Interface:
 Original nexthop: 10.1.1.2
 Qos information : 0x0
 AS-path Nil, origin igp, MED 0, localpref 100, pref-val 0, internal, pre 255
 Originator:  1.1.1.1
 Cluster list: 0.0.0.1
 Not advertised to any peer yet
```

In the routing table, you can view that Router D has learned from Router B the route advertised by Router A. In addition, you can view the Originator and Cluster_ID attributes of this route.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.3.2 255.255.255.0
#
bgp 65010
 router-id 1.1.1.1
 peer 10.1.1.1 as-number 65010
 peer 10.1.3.1 as-number 65010
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 10.1.1.1 enable
  peer 10.1.3.1 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.4.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.5.1 255.255.255.0
#
interface Pos4/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.1 255.255.255.0
#
bgp 65010
 router-id 2.2.2.2
 peer 10.1.1.2 as-number 65010
 peer 10.1.2.2 as-number 65010
 group in_rr internal
 peer 10.1.4.2 as-number 65010
 peer 10.1.4.2 group in_rr
 peer 10.1.5.2 as-number 65010
 peer 10.1.5.2 group in_rr
 #
 ipv4-family unicast
  undo synchronization
  undo reflect between-clients
  reflector cluster-id 1
  peer 10.1.1.2 enable
  peer 10.1.2.2 enable
  peer in_rr enable
  peer in_rr reflect-client
  peer 10.1.4.2 enable
  peer 10.1.4.2 group in_rr
  peer 10.1.5.2 enable
  peer 10.1.5.2 group in_rr
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.2.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.3.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.7.1 255.255.255.0
#
interface Pos4/0/0
```

```
 undo shutdown
 link-protocol ppp
 ip address 10.1.8.1 255.255.255.0
#
interface Pos5/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.9.1 255.255.255.0
#
bgp 65010
 router-id 3.3.3.3
 peer 10.1.2.1 as-number 65010
 peer 10.1.3.2 as-number 65010
 group in_rr internal
 peer 10.1.7.2 as-number 65010
 peer 10.1.7.2 group in_rr
 peer 10.1.8.2 as-number 65010
 peer 10.1.8.2 group in_rr
 peer 10.1.9.2 as-number 65010
 peer 10.1.9.2 group in_rr
 #
 ipv4-family unicast
  undo synchronization
  reflector cluster-id 2
  peer 10.1.2.1 enable
  peer 10.1.3.2 enable
  peer in_rr enable
  peer in_rr reflect-client
  peer 10.1.7.2 enable
  peer 10.1.7.2 group in_rr
  peer 10.1.8.2 enable
  peer 10.1.8.2 group in_rr
  peer 10.1.9.2 enable
  peer 10.1.9.2 group in_rr
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.4.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.6.1 255.255.255.0
#
bgp 65010
 router-id 4.4.4.4
 peer 10.1.4.1 as-number 65010
 peer 10.1.6.2 as-number 65010
 #
 ipv4-family unicast
  undo synchronization
  peer 10.1.4.1 enable
  peer 10.1.6.2 enable
#
return
```

NOTE

Configuration files of other routers are the same as the configuration file of Router D and are not mentioned here.

8.21.11 Example for Configuring BFD for BGP

After BFD for BGP is configured, BFD can fast detect the fault on the link between BGP peers and notify it to BGP so that service traffic can be transmitted through the backup link.

Networking Requirements

CAUTION

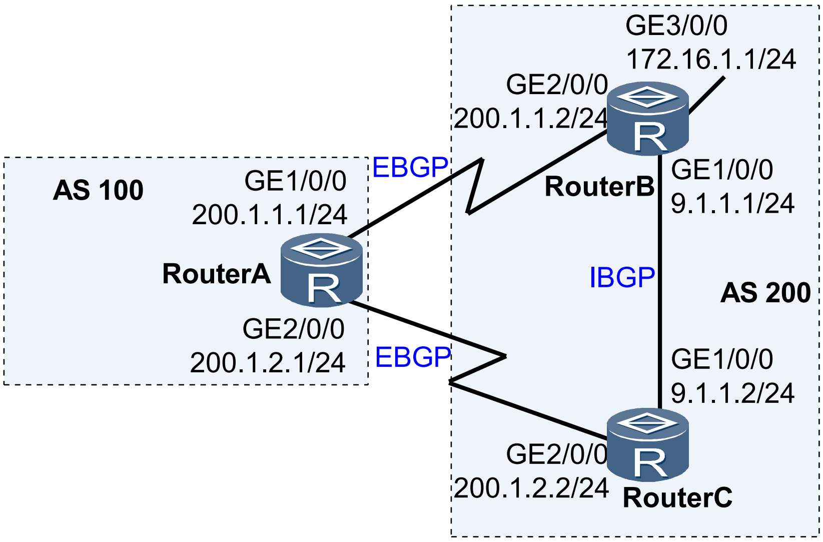
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-16**, Router A belongs to AS 100; Router B and Router C belong to AS 200; EBGP connections are established between Router A and Router B, and between Router A and Router C.

Service traffic is transmitted on the primary link Router A→Router B. The link Router A→ Router C→Router B functions as the backup link.

BFD is configured to detect the BGP neighbor relationship between Router A and Router B. When the link between Router A and Router B becomes faulty, BFD can fast detect the fault and notify it to BGP. Then, service traffic is transmitted through the backup link.

Figure 8-16 Networking diagram of configuring BFD for BGP



Configuration Notes

When configuring BFD for BGP, pay attention to the following:

- Before configuring BFD for BGP, enable BFD globally.

- When configuring BFD for BGP, ensure that parameters configured on the two ends of a BFD session are consistent.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP functions on each router.
2. Configure the MED attribute to control route selection.
3. Enable BFD on Router A and Router B

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, and Router C
- Peer IP address to be detected by BFD
- Minimum interval for sending BFD Control packets, minimum interval for receiving BFD Control packets, and local detection multiplier

Procedure

Step 1 Configure an IP address for each interface on the routers. The configuration details are not mentioned here.

Step 2 Configure basic BGP functions, establish EBGP connections between Router A and Router B, and between Router A and Router C, and establish an IBGP connection betweenRouter B and Router C.

# Configure Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 200.1.1.2 as-number 200
[~RouterA-bgp] peer 200.1.2.2 as-number 200
[~RouterA-bgp] commit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 200
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 200.1.1.1 as-number 100
[~RouterB-bgp] peer 9.1.1.2 as-number 200
[~RouterB-bgp] network 172.16.1.0 255.255.255.0
[~RouterB-bgp] commit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~Routerc] bgp 200
[~Routerc-bgp] router-id 3.3.3.3
[~Routerc-bgp] peer 200.1.2.1 as-number 100
[~Routerc-bgp] peer 9.1.1.1 as-number 200
[~Routerc-bgp] commit
[~Routerc-bgp] quit
```

# On Router A, you can view that the BGP neighbor relationship has been established.

```
<RouterA> display bgp peer
 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Total number of peers : 2                 Peers in established state : 2
  Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  200.1.1.2       4   200        2        5     0 00:01:25 Established       0
  200.1.2.2       4   200        2        4     0 00:00:55 Established       0
```

Step 3 Configure the MED attribute.

# Set the MED value for the routes sent from Router B and Router C to Router A by using a routing policy.

# Configure Router B.

```
[~RouterB] route-policy 10 permit node 10
[~RouterB-route-policy] apply cost 100
[~RouterB-route-policy] commit
[~RouterB-route-policy] quit
[~RouterB] bgp 200
[~RouterB-bgp] peer 200.1.1.1 route-policy 10 export
[~RouterB-bgp] commit
```

# Configure Router C.

```
[~RouterC] route-policy 10 permit node 10
[~RouterC-route-policy] apply cost 150
[~RouterC-route-policy] commit
[~RouterC-route-policy] quit
[~RouterC] bgp 200
[~RouterC-bgp] peer 200.1.2.1 route-policy 10 export
[~RouterC-bgp] commit
```

# Check the BGP routing table of Router A.

```
<RouterA> display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 2
      Network           NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   172.16.1.0/24     200.1.1.2      100                   0      200i
 *                      200.1.2.2      150                   0      200i
```

In the BGP routing table, you can view that the next-hop address of the route to 172.16.1.0/24 is 200.1.1.2, and traffic is transmitted on the primary link Router A→Router B.

Step 4 Configure BFD, and set the interval for sending BFD Control packets, the interval for receiving BFD Control packets, and the local detection multiplier.

# Enable BFD on Router A, and specify the minimum interval for sending BFD Control packets to 100 ms, the minimum interval for receiving BFD Control packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] bgp 100
[~RouterA-bgp] peer 200.1.1.2 bfd enable
[~RouterA-bgp] peer 200.1.1.2 bfd min-tx-interval 100 min-rx-interval 100 detect-
multiplier 4
[~RouterA-bgp] commit
```

# Enable BFD on Router B, and specify the minimum interval for sending BFD Control packets to 100 ms, the minimum interval for receiving BFD Control packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterB] bfd
[~RouterB-bfd] quit
```

```
[~RouterB] bgp 200
[~RouterB-bgp] peer 200.1.1.1 bfd enable
[~RouterB-bgp] peer 200.1.1.1 bfd min-tx-interval 100 min-rx-interval 100 detect-
multiplier 4
[~RouterB-bgp] commit
```

# Check all the BFD sessions established on Router A.

```
<RouterA> display bgp bfd session all
--------------------------------------------------------------------------------
  Local_Address      Peer_Address       Interface
  200.1.1.1          200.1.1.2          GigibitEthernet1/0/0
  Tx-interval(ms)    Rx-interval(ms)    Multiplier  Session-State
  100                100                4           Up
--------------------------------------------------------------------------------
```

Step 5 Verify the configuration.

# Run the **shutdown** command on GE 2/0/0 of Router B to simulate a fault on the primary link.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-GigabitEthernet2/0/0] shutdown
[~RouterB-GigabitEthernet2/0/0] commit
```

# Check the BGP routing table of Router A.

```
<RouterA> display bgp routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>   172.16.1.0/24      200.1.2.2       150                   0      200i
```

In the BGP routing table, you can view that the backup link Router A→Router C→Router B takes effect after the primary link fails, and the next-hop address of the route to 172.16.1.0/24 becomes 200.1.2.2.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 200.1.2.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 200.1.1.1 255.255.255.0
#
bgp 100
 router-id 1.1.1.1
 peer 200.1.1.2 as-number 200
 peer 200.1.1.2 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier
4
 peer 200.1.1.2 bfd enable
 peer 200.1.2.2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.1.2 enable
  peer 200.1.2.2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 200.1.1.2 255.255.255.0
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.16.1.1 255.255.255.0
#
bgp 200
 router-id 2.2.2.2
 peer 9.1.1.2 as-number 200
 peer 200.1.1.1 as-number 100
 peer 200.1.1.1 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier
4
 peer 200.1.1.1 bfd enable
 #
 ipv4-family unicast
  undo synchronization
  network 172.16.1.0 255.255.255.0
  peer 9.1.1.2 enable
  peer 200.1.1.1 enable
  peer 200.1.1.1 route-policy 10 export
#
route-policy 10 permit node 10
 apply cost 100
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 200.1.2.2 255.255.255.0
#
interface GigabitEthernet2/0/0
undo shutdown
 ip address 9.1.1.2 255.255.255.0
#
bgp 200
 router-id 3.3.3.3
 peer 9.1.1.1 as-number 200
 peer 200.1.2.1 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  peer 9.1.1.1 enable
  peer 200.1.2.1 enable
  peer 200.1.2.1 route-policy 10 export
#
route-policy 10 permit node 10
 apply cost 150
#
```

```
return
```

8.21.12 Example for Configuring BGP Auto FRR

Configuring BGP Auto FRR can provide information about the backup route. This improves network reliability.

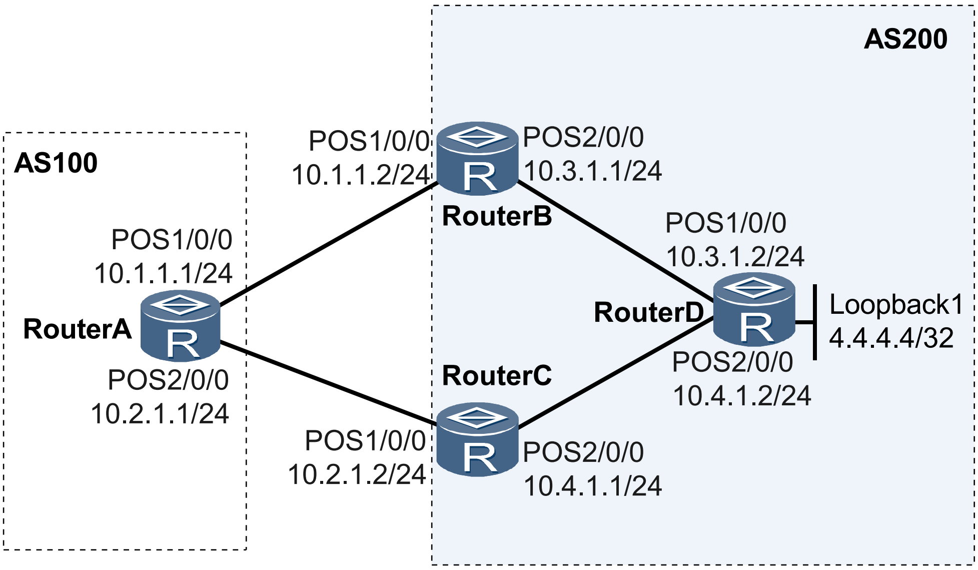
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 8-17**, Router A belongs to AS 100; Router B, Router C, and Router D belong to AS 200. BGP Auto FRR needs to be configured to ensure that the route from Router A to RouterD has the backup route.This improves network reliability.

Figure 8-17 Networking diagram of configuring BGP Auto FRR



Configuration Notes

When configuring BGP Auto FRR, pay attention to the following:

- When configuring BGP FRR, ensure that there are at least two routes to the same destination network segment.
- When specifying a routing policy, ensure that the routing policy name is case sensitive.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure EBGP connections between Router A and Router B, and between Router A and Router C. Configure IBGP connections between Router D and Router B, and between Router D and Router C.
2. Configure routing policies on Router B and Router C to change the MED values of routes to Router D to facilitate route selection.
3. Configure BGP Auto FRR on Router A.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, Router C, and Router D
- Names of routing policies and MED values of routes on Router B and Router C

Procedure

Step 1 Configure an IP address for each interface. The configuration details are not mentioned here.

Step 2 Configure EBGP connections between Router A and Router B, and between Router A and Router C, and configure IBGP connections between Router B and Router D, and between Router C and Router D.

# Configure EBGP connections on Router A.

```
<RouterA> system-view
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 10.1.1.2 as-number 200
[~RouterA-bgp] peer 10.2.1.2 as-number 200
[~RouterA-bgp] commit
```

NOTE

The configurations on Router B and Router C are similar to the configuration on Router A, and the detailed configurations are not mentioned here.

# Configure IBGP connections on Router D.

```
<RouterD> system-view
[~RouterD] bgp 200
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 10.3.1.1 as-number 200
[~RouterD-bgp] peer 10.4.1.1 as-number 200
[~RouterD-bgp] commit
```

NOTE

The configurations on Router B and Router C are similar to the configuration on Router D, and the detailed configurations are not mentioned here.

Step 3 Configure routing policies on Router B and Router C to ensure that the MED values of routes to Router D are different.

# Configure a routing policy on Router B.

```
<RouterB> system-view
[~RouterB] route-policy rtb permit node 10
[~RouterB-route-policy] apply cost 80
[~RouterB-route-policy] quit
[~RouterB] bgp 200
[~RouterB-bgp] ipv4-family unicast
[~RouterB-bgp-af-ipv4] peer 10.1.1.1 route-policy rtb export
[~RouterB-bgp-af-ipv4] commit
[~RouterB-bgp-af-ipv4] quit
```

# Configure a routing policy on Router C.

```
<RouterC> system-view
[~RouterC] route-policy rtc permit node 10
[~RouterC-route-policy] apply cost 120
[~RouterC-route-policy] quit
[~RouterC] bgp 200
[~RouterC-bgp] ipv4-family unicast
[~RouterC-bgp-af-ipv4] peer 10.2.1.1 route-policy rtc export
[~RouterC-bgp-af-ipv4] commit
[~RouterC-bgp-af-ipv4] quit
```

# Advertise a route to 4.4.4.4/32 on Router D.

```
[~RouterD] bgp 200
[~RouterD-bgp] ipv4-family unicast
[~RouterD-bgp] network 4.4.4.4 32
[~RouterD-bgp] commit
```

# Run the **display ip routing-table verbose** command on Router A to check detailed information about the route to 4.4.4.4/32 it learns.

```
<RouterA> display ip routing-table 4.4.4.4 32 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 4.4.4.4/32
     Protocol: BGP             Process ID: 0
   Preference: 255                   Cost: 80
      NextHop: 10.1.1.2         Neighbour: 10.1.1.2
        State: Active Adv             Age: 00h00m12s
          Tag: 0                 Priority: low
        Label: NULL               QoSInfo: 0x0
   IndirectID: 0x4
 RelayNextHop: 0.0.0.0          Interface: Pos1/0/0
     TunnelID: 0x0                  Flags:  D
```

Because the MED value of the route learned from Router B is smaller, on Router A, the route to 4.4.4.4/32 selects the path Router A→Router B→Router D. Because FRR is not configured, no information about the backup route is available.

Step 4 Enable BGP Auto FRR on Router A, and check the routing information.

# Enable BGP Auto FRR on Router A.

```
<RouterA> system-view
[~RouterA] bgp 100
[~RouterA-bgp] ipv4-family unicast
[~RouterA-bgp-af-ipv4] auto-frr
[~RouterA-bgp-af-ipv4] commit
[~RouterA-bgp-af-ipv4] quit
```

# After the configuration, run the **display ip routing-table verbose** command on Router A to check the routing information.

```
<RouterA> display ip routing-table 4.4.4.4 32 verbose
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : _public_
Summary Count : 1

Destination: 4.4.4.4/32
     Protocol: BGP             Process ID: 0
   Preference: 255                   Cost: 80
      NextHop: 10.1.1.2         Neighbour: 10.1.1.2
        State: Active Adv             Age: 00h52m45s
```

```
         Tag: 0                 Priority: low
       Label: NULL               QoSInfo: 0x0
  IndirectID: 0x4
RelayNextHop: 0.0.0.0          Interface: Pos1/0/0
    TunnelID: 0x0                  Flags:  D
   BkNextHop: 10.2.1.2       BkInterface: Pos2/0/0
     BkLabel: NULL           SecTunnelID: 0x0
BkPETunnelID: 0x0        BkPESecTunnelID: 0x0
BkIndirectID: 0x2
```

The preceding information shows that Router A has a backup next hop and a backup outbound interface to 4.4.4.4/32.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.1 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.2.1.1 255.255.255.0
#
bgp 100
 router-id 1.1.1.1
 peer 10.1.1.2 as-number 200
 peer 10.2.1.2 as-number 200
#
 ipv4-family unicast
  auto-frr
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.1.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.3.1.1 255.255.255.0
#
bgp 200
 router-id 2.2.2.2
 peer 10.1.1.1 as-number 100
 peer 10.3.1.2 as-number 200
#
 ipv4-family unicast
  peer 10.1.1.1 route-policy rtb export
#
route-policy rtb permit node 10
 apply cost 80
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.2.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.4.1.1 255.255.255.0
#
bgp 200
 router-id 3.3.3.3
 peer 10.2.1.1 as-number 100
 peer 10.4.1.2 as-number 200
 #
 ipv4-family unicast
  peer 10.2.1.1 route-policy rtc export
#
route-policy rtc permit node 10
 apply cost 120
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.3.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.4.1.2 255.255.255.0
#
interface LoopBack1
 ip address 4.4.4.4 255.255.255.255
#
bgp 200
 router-id 4.4.4.4
 peer 10.3.1.1 as-number 200
 peer 10.4.1.1 as-number 200
 #
 ipv4-family unicast
  network 4.4.4.4 255.255.255.255
#
return
```

9 BGP4+ Configuration

About This Chapter

By transmitting routing information between Autonomous Systems (ASs), BGP4+ is useful for large-scale and complex IPv6 networks.

9.11 Configuring BGP4+ Connection Parameters
By configuring BGP4+ connection parameters, you can optimize BGP4+ network performance.

9.12 Configuring BGP4+ RRs
BGP4+ RRs resolve the problem of full-mesh connections between multiple IBGP peers, which reduces network costs.

9.13 Configuring BFD for BGP4+
BFD for BGP4+ provides BGP4+ with a mechanism for quickly detecting faults, which speeds up network convergence.

9.14 Configuring BGP4+ Auto FRR
As a protection measure against link faults, BGP4+ Auto Fast Reroute (FRR) is applicable to the network topology with primary and backup links. BGP4+ Auto FRR is applicable to services that are very sensitive to packet loss and delays.

9.15 Configuring BGP4+ GR Helper
A BGP4+ GR helper helps its neighbor complete BGP4+ GR.

9.16 Configuring BGP4+ Authentication
BGP4+ authentication can be configured to enhance security of BGP4+ networks.

9.17 Configuring BGP4+ 6PE
BGP4+ 6PE enables separated IPv6 networks to communicate by using the MPLS tunneling technology.

9.18 Maintaining BGP4+
Maintaining BGP4+ involves resetting BGP4+ connections, clearing BGP4+ statistics, and debugging BGP4+.

9.19 Configuration Examples
This chapter provides several BGP4+ configuration examples. In each configuration example, the networking requirements, network diagrams, precautions, configuration roadmap, and configuration procedures are provided.

9.1 BGP4+ Overview

BGP4+ controls route advertisement and makes routing decisions.

As an extension of BGP, BGP4+ is a dynamic routing protocol used between ASs.

Traditional BGP4 only manages IPv4 routing information. The inter-AS transmission of packets encapsulated by other network layer protocols (such as IPv6) is restricted.

To support multiple network layer protocols, the Internet Engineering Task Force (IETF) has extended BGP4 to make BGP4+, which is defined in RFC 2858 (Multiprotocol Extensions for BGP-4).

To support IPv6, BGP4 must have the additional ability to associate an IPv6 protocol with the next hop information and network layer reachable information (NLRI).

Two NLRI attributes are introduced in BGP4+.

- Multiprotocol Reachable NLRI (MP_REACH_NLRI): carries the set of reachable destinations and the next-hop information used for forwarding packets to these destinations.
- Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI): carries the set of unreachable destinations.

The Next_Hop attribute in BGP4+ is in the format of an IPv6 address, which can be either a globally unique IPv6 address or a next-hop link-local address.

Using multiple protocol extensions for BGP4, BGP4+ can carry IPv6 routing information without changing the messaging and routing mechanisms of BGP4.

9.2 BGP4+ Features Supported by the NE5000E

The NE5000E supports various BGP4+ features, including load balancing, manual route aggregation, route dampening, community usage, route reflector usage, BGP4+ accounting, 6PE, BFD for BGP4+, BGP4+ Graceful Restart (GR), and BGP4+ Non-Stop Routing (NSR).

6PE

The 6PE function uses the Multi-Protocol Label Switching (MPLS) tunneling technology to connect IPv6 networks separated by IPv4 or MPLS networks. The Internet Service Provider's (ISP's) PEs support the IPv4/IPv6 dual stack to form a tunnel in 6PE mode. The tunnel identifies IPv6 routes by the label assigned to those routes by the Multiprotocol Border Gateway Protocol (MP-BGP), and implements IPv6 forwarding by using LSPs between PEs.

Other Features

Most of the BGP4+ features supported by the NE5000E are similar to the BGP features supported by the NE5000E.

For details about the BGP features supported by the NE5000E, see the chapter "BGP Configuration" in the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Routing*. BGP4+ does not support automatic route aggregation or MP-BGP.

9.3 Configuring Basic BGP4+ Functions

Basic BGP4+ functions must be configured before you configure subsequent BGP4 functions on a BGP4+ network.

Applicable Environment

Basic BGP4+ functions must be configured first when you configure BGP4+ to implement inter-AS communication.

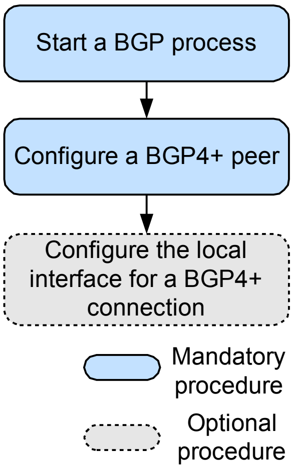
Pre-configuration Tasks

Before configuring basic BGP4+ functions, complete the following task:

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

Configuration Procedures

Figure 9-1 Flowchart for configuring basic BGP4+ functions



Related Tasks

9.19.1 Example for Configuring Basic BGP4+ Functions

9.3.1 Starting a BGP Process

A BGP process must be started before configuring BGP functions. When starting a BGP process, specify the number of the AS to which the device belongs.

Context

CAUTION

Changing router IDs on BGP peers will cause the reestablishment of the BGP connection between routers.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

BGP is enabled (the local AS number is specified), and the BGP view is displayed.

Step 3 (Optional) Run:

```
router-id ipv4-address
```

The router ID is set.

By default, BGP automatically selects the router ID in the system view as its router ID. For the rules governing the selection of router IDs in the system view, see the *HUAWEI NetEngine5000E Core Router Command Reference*.

NOTE

If the router ID of the router is the IP address of a physical interface, the change in the IP address will cause route flapping on the network. To enhance network stability, configuring the address of a loopback interface as the router ID is recommended.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.3.2 Configuring IPv6 Peers

Devices can exchange BGP4+ routing information only after they are configured as IPv6 peers and the IPv6 peer relationship is established among them.

Context

Because BGP4+ uses TCP connections, the IPv6 addresses for peers must be specified when you configure BGP4+. A BGP4+ peer may not be a neighboring router, and a BGP4+ peer relationship can be created by using a logical link. Using the addresses of loopback interfaces to set up BGP4+ peer relationships can improve the stability of BGP4+ connections and is thus recommended.

IBGP peer relationships need to be established among the devices within an AS; EBGP peer relationships need to be established among the devices in different ASs.

Procedure

- Configure IBGP peers.

  Do as follows on the routers between which an IBGP peer relationship needs to be set up:

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     peer ipv6-address as-number as-number
     ```
     The address of the remote peer and the AS to which the remote peer belongs are configured.

     The number of the AS where the specified peer resides should be the same as that of the local AS.

     The IP address of the specified peer can be one of the following types:

     - IPv6 address of an interface on a directly-connected peer
     - IP address of a loopback interface on a reachable peer
     - IPv6 address of a sub-interface on a directly-connected peer
     - Link-local address of an interface on a directly-connected peer

     If the IPv6 address used to set up a BGP4+ peer relationship is a loopback interface address, the procedure described in **Configuring the Local Interface for the BGP4+ Connection** needs to be followed. After that, the BGP4+ peer relationship can be set up successfully.
  4. (Optional) Run:
     ```
     peer { ipv6-address | group-name } listen-only
     ```
     The local peer (group) is configured to only listen to connection requests, not to send connection requests.

     After this command is run, the existing peer relationship is interrupted. The local peer will wait for a connection request from the remote peer to reestablish a peer relationship. This command enables only one peer to send connection requests, preventing a connection request conflict.

     NOTE

     This command can only be run on one of two peers. If this command is run on both of the two peers, the connection between the two peers cannot be reestablished.
  5. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  6. Run:
     ```
     peer ipv6-address enable
     ```
     The IPv6 peer is enabled.

After configuring a BGP4+ peer in the BGP view, enable the peer in the IPv6 unicast address family view.

7. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure EBGP peers.

  Do as follows on the routers between which an EBGP peer relationship needs to be set up:

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   peer ipv6-address as-number as-number
   ```

   The IPv6 address of the remote peer and the AS to which the remote peer belongs are configured.

   The number of the AS where the specified peer resides should be different from that of the local AS.

   The IP address of the specified peer can be one of the following types:

   - IPv6 address of an interface on a directly-connected peer
   - IP address of a loopback interface on a reachable peer
   - IPv6 address of a sub-interface on a directly-connected peer
   - Link-local address of an interface on a directly-connected peer

   If the IP address used to set up a BGP4+ peer relationship is a loopback interface address, or a local-link address, the procedure described in **Configuring the Local Interface for the BGP4+ Connection** needs to be followed. After that, the BGP4+ peer relationship can be set up successfully.

4. Run:

   ```
   peer { ipv6-address | group-name } ebgp-max-hop [ hop-count ]
   ```

   The maximum number of hops is configured for establishing an EBGP connection.

   A direct physical link must be available between EBGP peers. If such a link does not exist, the **peer ebgp-max-hop** command must be used to allow EBGP peers to establish a TCP connection over multiple hops.

   NOTE

   If loopback interfaces are used to establish an EBGP peer relationship, the **peer ebgp-max-hop** command must be run with *hop-count* greater than or equal to 2; otherwise, the peer relationship cannot be established.

5. (Optional) Run:

   ```
   peer { ipv6-address | group-name } listen-only
   ```

   The local peer (group) is configured to only listen to connection requests, not to send connection requests.

After this command is run, the existing peer relationship is interrupted. The local peer will wait for the connection request from the remote peer to reestablish a peer relationship. This configuration can prevent connection requests from conflicting.

NOTE

This command can only be run on one of two peers. If this command is run on both of the two peers, the connection between the two peers cannot be reestablished.

6. Run:

   ```
   ipv6-family unicast
   ```

   The IPv6 unicast address family view is displayed.

7. Run:

   ```
   peer ipv6-address enable
   ```

   The IPv6 peer is enabled.

   After configuring a BGP4+ peer in the BGP view, enable the peer in the IPv6 unicast address family view.

8. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.3.3 (Optional) Configuring the Local Interface for a BGP4+ Connection

When loopback interfaces are used to establish a BGP4+ connection between two peers, the local interfaces used for the BGP4+ connection need to be configured on the devices.

Context

To improve the reliability and stability of a BGP4+ connection, the local interfaces used for the BGP4+ connection should be configured as loopback interfaces. When redundant links are available on the network, the BGP connection will not be torn down if an interface or a link fails.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
peer { ipv6-address | group-name } connect-interface interface-type interface-number [ ipv6-source-address ]
```

The source interface and source address used to set up a TCP connection are specified.

Usually, BGP4+ uses the physical interface that is directly connected to the peer as the local interface of a TCP connection.

NOTE

When multiple peer relationships between two routers are established by using various links, running the **peer connect-interface** command to specify the local interface for a BGP4+ connection is recommended.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.3.4 Checking the Configuration

After basic BGP4+ functions are configured, BGP4+ peer information can be viewed.

Prerequisite

The configurations of basic BGP4+ functions are complete.

Procedure

- Run the **display bgp ipv6 peer** *ipv6-address* **verbose** command to check detailed information about BGP4+ peers.
- Run the **display bgp ipv6 peer** *ipv6-address* **log-info** command to check the log information of BGP4+ peers.

----End

Example

# Run the **display bgp ipv6 peer** command, and you view information about BGP4+ peers. For example:

```
<HUAWEI> display bgp ipv6 peer
BGP Local router ID : 20.0.0.1
local AS number : 100
 Total number of peers : 1                 Peers in established state : 1

 Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
 20::21          4   200       17       19     0 00:09:59 Established       3
```

9.4 Configuring a BGP4+ Peer Group

Configuring a BGP4+ peer group simplifies the management of routing policies and improves the efficiency of route advertisement.

Applicable Environment

A large-scale BGP4+ network has a large number of peers. It is difficult to configure and maintain them. To address this problem, peer groups can be configured to simplify the management of routing policies and improve the efficiency of route advertisement. Peer groups are classified as IBGP peer groups and EBGP peer groups depending upon whether the peers reside in the same AS or not, respectively. EBGP peer groups are further divided into pure EBGP peer groups and mixed EBGP peer groups. If all the peers in an EBGP peer group reside in the same external AS, the group is a pure EBGP peer group; otherwise, the group is a mixed EBGP peer group.

Pre-configuration Tasks

Before configuring a BGP4+ peer group, complete the following task:

- **Configuring Basic BGP4+ Functions**

Procedure

- Configure an IBGP peer group.

  If BGP4+ has multiple IBGP peers, adding them to an IBGP peer group can simplify routing policy management. When creating an IBGP peer group, you do not need to specify the AS number.

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     group group-name [ internal ]
     ```
     An IBGP peer group is created.
  4. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  5. Run:
     ```
     peer group-name enable
     ```
     The IBGP peer group is enabled.
  6. Run:
     ```
     peer { ipv4-address | ipv6-address } group group-name
     ```
     An IPv6 peer is added to the peer group.

     NOTE

     After an IBGP peer is added to a peer group, the system automatically creates an IPv6 peer in the BGP view and enables the IPv6 peer in the IPv6 address family view.
  7. Run:
     ```
     commit
     ```
     The configuration is committed.

- Configure a pure EBGP peer group.

  If BGP4+ has multiple EBGP peers that belong to the same AS, adding them to a pure EBGP peer group can simplify routing policy management. In a pure EBGP peer group, all the peers must reside in the same AS.

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.

2. Run:
   bgp *as-number*

   The BGP view is displayed.
3. Run:
   group *group-name* **external**

   An EBGP peer group is created.
4. Run:
   peer *group-name* **as-number** *as-number*

   The AS number is specified for the peer group.
5. Run:
   ipv6-family unicast

   The IPv6 unicast address family view is displayed.
6. Run:
   peer *group-name* **enable**

   The EBGP peer group is enabled.
7. Run:
   peer *ipv6-address* **group** *group-name*

   An IPv6 peer is added to the peer group.

   After an EBGP peer is added to a peer group, the system automatically creates an EBGP peer in the BGP view and enables the EBGP peer in the IPv6 address family view.

   An AS number needs to be specified for the pure EBGP peer group.

   If the peer group already has peers in it, no AS number needs to be specified for the peer group.
8. Run:
   commit

   The configuration is committed.

- Configure a mixed EBGP peer group.

  If BGP4+ has multiple EBGP peers that belong to different ASs, adding them to a mixed EBGP peer group can simplify the routing policy management. When creating a mixed EBGP peer group, specify an AS number for each peer.

1. Run:
   system-view

   The system view is displayed.
2. Run:
   bgp *as-number*

   The BGP view is displayed.
3. Run:
   group *group-name* **external**

   An EBGP peer group is created.

4. Run:

   peer *ipv6-address* **as-number as-number** *as-number*

   The AS number is specified for an IPv6 peer.

5. Run:

   ipv6-family unicast

   The IPv6 unicast address family view is displayed.

6. Run:

   peer *group-name* **enable**

   The EBGP peer group is enabled.

7. Run:

   peer *ipv6-address* **group** *group-name*

   The created IPv6 peer is added to the peer group.

   After the EBGP peer is added to the peer group, the system automatically enables the EBGP peer in the IPv6 address family view.

   When creating a mixed EBGP peer group, you need to create peers and specify their AS numbers separately. No AS number needs to be specified for the mixed EBGP peer group.

8. Run:

   commit

   The configuration is committed.

----End

Checking the Configuration

After the configuration is complete, you can check if the configuration has taken effect.

- Run the **display bgp ipv6 group** [*group-name*] command to check information about peer groups.

# Display information about all BGP IPv6 peer groups.

```
<HUAWEI> display bgp ipv6 group
 BGP peer-group: in
 Remote AS: 100
 Authentication type configured: None
 Type : internal
 PeerSession Members:
   2::1
 Peer Members:
   1::1              2::1
 ***********************
 BGP peer-group is ex
 Remote AS number not specified
 Authentication type configured: None
 Type : external
 PeerSession Members:
   20::1
 Peer Members:
   10::1             20::1
```

9.5 Controlling Route Import

BGP4+ can import the routes discovered by other protocols into the BGP4+ routing table. When importing IGP routes, BGP can filter routes by routing protocol.

Applicable Environment

BGP4+ cannot discover routes by itself. Instead, it imports routes discovered by other protocols such as OSPFv3 or static routes into the BGP4+ routing table. These imported routes are then transmitted within an AS or between ASs. When importing routes, BGP4+ can filter these routes by routing protocol.

Pre-configuration Tasks

Before controlling route import, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Choose certain configuration tasks from the following (except "Checking the Configuration") as needed for the particular usage scenario.

9.5.1 Configuring BGP4+ to Import Routes

BGP4+ can import routes from other protocols. When routes are imported from dynamic routing protocols, the process IDs of the routing protocols must be specified.

Context

BGP4+ cannot discover routes by itself. Instead, it imports routes discovered by other protocols such as OSPFv3 or static routes to the BGP4+ routing table. These imported routes are then transmitted within an AS or between ASs.

BGP4+ can import routes in either Import or Network mode:

- In Import mode, BGP4+ imports routes from a specific routing protocol. RIP routes, OSPF routes, Intermediate System-to-Intermediate System (IS-IS) routes, static routes, or direct routes can be imported into the BGP4+ routing table.
- In Network mode, routes with the specified prefix and mask are imported into the BGP4+ routing table. Compared with the Import mode, the Network mode imports more specific routes.

Procedure

- Configure BGP4+ to import routes in Import mode.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     bgp as-number
     ```

The BGP view is displayed.

3. Run:

   ipv6-family unicast

   The BGP-IPv6 unicast address family view is displayed.

4. Run:

   import-route *protocol* [*process-id*] [**med** *med* | **route-policy** *route-policy-name*] *

   BGP4+ is configured to import routes from other protocols.

   NOTE

   The process ID of a routing protocol needs to be specified to import IS-IS, OSPF, or RIP routes.

5. Run:

   commit

   The configuration is committed.

- Configure BGP4+ to import routes in Network mode.

1. Run:

   system-view

   The system view is displayed.

2. Run:

   bgp *as-number*

   The BGP view is displayed.

3. Run:

   ipv6-family unicast

   The BGP-IPv6 unicast address family view is displayed.

4. Run:

   network *ipv6-address prefix-length* [**route-policy** *route-policy-name*]

   A local IPv6 route is advertised.

   If no mask or mask length is specified, the IP address is processed as a classful address.

   In order for a local route to be advertised, it must be in the local IP routing table. Routing policies can be used to control what routes are advertised with more flexibility.

   NOTE

   - The destination address and mask specified in the **network** command must be consistent with the corresponding entry in the local IP routing table. Otherwise, the specified route will not be advertised.
   - When using the **undo network** command to clear the existing configuration, remember to specify the correct mask.

5. Run:

   commit

   The configuration is committed.

----End

9.5.2 Configuring BGP4+ to Import Default Routes

This section describes how to configure BGP4+ to import default routes from the local routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
default-route imported
```

BGP4+ is configured to import default routes.

To import default routes, run both the **default-route imported** command and the **import-route** command. If only the **import-route** command is used, default routes cannot be imported. In addition, the **default-route imported** command is used to import only the default routes that exist in the local routing table.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.5.3 Checking the Configuration

After controlling route import is configured, you can view information about filtered and imported routes.

Prerequisite

The configurations of controlling route import are complete.

Procedure

- Run the **display bgp ipv6 routing-table** [*ipv6-address prefix-length*] command to check BGP4+ routes.
- Run the **display bgp ipv6 routing-table as-path-filter** *as-path-filter-number* command to check routes filtered by the AS_Path attribute.

----End

Example

# Run the **display bgp ipv6 routing-table** command, and you can view information about BGP4+ routes. For example:

```
<HUAWEI> display bgp ipv6 routing-table

 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 1
 *>  Network  : 7::                                    PrefixLen : 64
     NextHop  : 10::2                                  LocPrf    :
     MED      : 150                                    PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
```

9.6 Controlling Route Advertisement

BGP4+ can filter or apply routing policies to the routes to be advertised to a peer or peer group.

Applicable Environment

BGP4+ advertises routes to peers based on the network plan for exchange of routing information between devices. The routes can be filtered or processed based on a routing policy before being advertised to a peer or peer group.

Pre-configuration Tasks

Before controlling route advertisement, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Choose certain configuration tasks from the following (except "Checking the Configuration") as needed in the applicable environment.

9.6.1 Configuring BGP4+ Route Summarization

Configuring route summarization can reduce the size of a routing table on a peer.

Context

On a large-scale BGP4+ network, configuring route summarization can reduce the number of advertised route prefixes and improve BGP4+ stability.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run one of the following commands to configure route summarization as needed.

- To advertise all summarized routes and specific routes, run:

  ```
  aggregate ipv6-address { mask | mask-length }
  ```

- To advertise only summarized routes, run:

  ```
  aggregate ipv6-address { mask | mask-length } detail-suppressed
  ```

- To advertise some of the specific routes, run:

  ```
  aggregate ipv6-address { mask | mask-length } suppress-policy route-policy-name
  ```

  Using the **peer route-policy** command can also advertise some of the specific routes.

- To generate the summarized route used for loop detection, run:

  ```
  aggregate ipv6-address { mask | mask-length } as-set
  ```

- To configure the attributes of summarized routes, run:

  ```
  aggregate ipv6-address { mask | mask-length } attribute-policy route-policy-name
  ```

  Using the **peer route-policy** command can also configure the attributes of summarized routes.

  If **as-set** is configured in the **aggregate** command, the AS_Path attribute configured in the **apply as-path** command does not take effect.

- To generate summarized routes based on some of the specific routes, run:

  ```
  aggregate ipv6-address { mask | mask-length } origin-policy route-policy-name
  ```

Manual summarization is valid for routing entries in the local BGP4+ routing table. For example, if a route with a mask longer than 64, such as 9:3::1/128, does not exist in the BGP4+ routing table, BGP4+ does not advertise the summarized route after the aggregate 9:3::1 64 command is used to summarize routes.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.6.2 Configuring BGP4+ to Advertise Default Routes to Peers

A device sends a default route with the local address as the next hop address to the specified peer for load balancing purpose, regardless of whether there are default routes in the local routing table. This greatly reduces the number of routes on the network.

Context

Default routes can be used in the networks that have the following characteristics:

- There are multiple EBGP peers, and each peer can receive full Internet routes.
- There are multiple Route Reflectors (RRs), and each RR can receive full Internet routes.

When load balancing is not performed on the network, a BGP4+ peer receives at most one copy of active full Internet routes. After load balancing is performed on the network, the number of active routes received by the BGP4+ peer doubles, which causes the number of routes on the network to sharply increase. In this case, you can configure the local device to advertise only default routes to its BGP4+ peer and use default routes for load balancing, which can greatly reduce the number of routes on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
peer { ipv4-address | ipv6-address | group-name } default-route-advertise [ route-policy route-policy-name ]
```

Default routes are sent to a remote peer or peer group.

NOTE

After the **peer default-route-advertise** command is used, BGP4+ sends a default route with the local address as the next hop to the specified peer, regardless of whether there are default routes in the routing table.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.6.3 Configuring BGP4+ to Advertise the Community Attribute

The community attribute is used to simplify routing policy management. Compared with a peer group, the community attribute has a larger management scope. It can control routing policies for multiple BGP4+ devices.

Context

The community attribute is transmitted between BGP4+ peers, and its transmission is not restricted within ASs. With the community attribute, a group of routes can share the same routing policy. Before sending a route with the community attribute to peers, BGP4+ can change the original community attribute carried by the route.

Procedure

- Configure BGP4+ to send the community attribute to peers.

  Do as follows on a BGP4+ device:

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  4. Run the following command as needed.
     - To send the standard community attribute to a peer group, run:
       ```
       peer { ipv4-address | ipv6-address | group-name } advertise-community
       ```
     - To send the extended community attribute to a peer group, run:
       ```
       peer { ipv4-address | ipv6-address | group-name } advertise-ext-community
       ```
  5. Run:
     ```
     commit
     ```
     The configuration is committed.

- Apply a routing policy to the routes to be advertised.

  Do as follows on a BGP4+ device:

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  4. Run:
     ```
     peer { ipv4-address | ipv6-address | group-name } route-policy route-policy-name export
     ```
     An export routing policy is configured.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.

NOTE

- When configuring a BGP4+ community, define a specific community attribute by using a routing policy. After that, apply the routing policy to the routes to be advertised.
- For details on routing policy configurations, see **Routing Policy Configuration**.

----End

9.6.4 Setting the Interval for Sending Update Packets

When routes change, the router sends Update packets to notify its peers. If a route changes frequently, to prevent the router from sending Update packets for every change, you need to set the interval for sending Update packets upon changes of this route.

Context

Do as follows on a BGP4+ device:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
peer ipv6-address route-update-interval interval
```

The interval for sending Update packets is set.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.6.5 Checking the Configuration

After controlling route advertisement is configured, you can view information about filtered and advertised routes.

Prerequisite

The configurations of controlling route advertisement are complete.

Procedure

- Run the **display bgp ipv6 network** command to check routing information advertised by BGP4+.
- Run the **display bgp routing-table cidr** command to check Classless InterDomain Routing (CIDR) information.
- Run the **display bgp ipv6 routing-table community** [*aa:nn* &<1-13>] [**internet** | **no-advertise** | **no-export** | **no-export-subconfed**] * [**whole-match**] command to check information about the routes carrying the specified BGP4+ community attribute.
- Run the **display bgp ipv6 peer** *ipv4-address* **verbose** command to check information about BGP4+ peers.
- Run the **display bgp ipv6 peer** *ipv6-address* { **log-info** | **verbose** } command to check the log information about BGP4+ peers.

----End

Example

# Run the **display bgp ipv6 peer** command, and you can view information about BGP4+ peers. For example:

```
<HUAWEI> display bgp ipv6 peer
BGP Local router ID : 20.0.0.1
local AS number : 100
 Total number of peers : 1                 Peers in established state : 1

 Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
 20::21          4   200       17       19     0 00:09:59 Established       3
```

9.7 Controlling BGP4+ Route Selection

The policy on BGP4+ route selection can be changed by configuring BGP4+ route attributes.

Applicable Environment

BGP4+ has many route attributes. These attributes are used to define the routing policy and describe the BGP4+ route prefix. Configuring these attributes can change the policy used by BGP4+ for route selection.

Pre-configuration Tasks

Before controlling BGP4+ route selection, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Choose certain configuration tasks from the following (except "Checking the Configuration") as needed in the applicable environment.

9.7.1 Setting the BGP4+ Preference

Setting the BGP4+ preference can affect route selection among BGP4+ and other routing protocols.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
preference { external internal local | route-policy route-policy-name }
```

The BGP4+ preference is set.

Default BGP preferences are as follows:

- EBGP external routes: 255
- IBGP internal routes: 255
- BGP local routes: 255

There are three types of BGP4+ routes, as listed below:

- EBGP routes learned from peers in other ASs
- IBGP routes learned from peers in the same AS
- Locally generated routes, which refer to the routes summarized by using the **aggregate** command

Different preferences can be set for these three types of routes.

In addition, routing policies can also be used to set the preferences for the routes that match the policies. The routes that do not meet the rules use the default preferences.

NOTE

At present, the **peer route-policy** command cannot be used to set the BGP4+ preference.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.7.2 Setting the Preferred Value for BGP4+ Routes

After the preferred value is set for BGP4+ routes, the route with the greatest value is preferred when multiple routes to the same destination exist in the BGP4+ routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
peer { ipv4-address | ipv6-address | group-name } preferred-value value
```

A preferred value is set for all the routes learned from a specified peer.

The original preferred value of a route learned from a peer defaults to 0.

After the **peer preferred-value** command is run, all the routes learned from a specified peer have the same preferred value.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.7.3 Setting the Default Local_Pref Attribute for the Local Device

After the Local_Pref attribute is set for BGP4+ routes, the route with the greatest attribute value is preferred when multiple routes to the same destination exist in the BGP4+ routing table. The preferred value takes precedence over the Local_Pref attribute.

Context

The Local_Pref attribute is used to determine the optimal route for the traffic that leaves an AS. If a BGP device obtains multiple routes from different IBGP peers and these routes have different next hops to the same destination, the BGP device will select the route with the greatest Local_Pref value.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
default local-preference local-preference
```

The default Local_Pref attribute is set for the local device.

By default, the Local_Pref value of BGP is 100.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.7.4 Setting the MED Attribute

The MED attribute is equal to the metric used in IGP. After the MED attribute is set for routes, an EBGP peer can select a route with the smallest MED value for the traffic that enters an AS.

Context

The MED serves as the metric used by an IGP. It is used to determine the optimal route when traffic enters an AS. When a BGP4+ router obtains multiple routes to the same destination address but with different next hops through EBGP peers, the route with the smallest MED value is selected as the optimal route.

Procedure

- Set the default MED value on the local device.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  4. Run:
     ```
     default med med
     ```
     The default MED value is set.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.
- Compare the MED values of the routes from different ASs.

1. Run:
   ```
   system-view
   ```
   The system view is displayed.
2. Run:
   ```
   bgp as-number
   ```
   The BGP view is displayed.
3. Run:
   ```
   ipv6-family unicast
   ```
   The IPv6 unicast address family view is displayed.
4. Run:
   ```
   compare-different-as-med
   ```
   The MED values of routes from different ASs are compared.

   By default, the BGP4+ router compares only the MED values of the routes from different peers in the same AS. This command enables BGP4+ to compare the MED values of routes from different ASs.
5. Run:
   ```
   commit
   ```
   The configuration is committed.

- Configure the method used by BGP4+ when there is no MED attribute in the route attributes.

1. Run:
   ```
   system-view
   ```
   The system view is displayed.
2. Run:
   ```
   bgp as-number
   ```
   The BGP view is displayed.
3. Run:
   ```
   ipv6-family unicast
   ```
   The IPv6 unicast address family view is displayed.
4. Run:
   ```
   bestroute med-none-as-maximum
   ```
   The maximum MED value is used for a route when there is no MED attribute in the route attributes.

   If this command is not run, BGP4+ uses 0 as the MED value for a route when there is no MED attribute in the route attributes.
5. Run:
   ```
   commit
   ```
   The configuration is committed.

----End

9.7.5 Setting the Next_Hop Attribute

BGP route selection can be flexibly controlled by setting the Next_Hop attribute.

Context

Different from that in IGP, the next-hop address in BGP4+ may not be the IPv6 address of a peer router.

Procedure

- Change the next-hop address when advertising a route to an IBGP peer.

  Do as follows on the IBGP router:

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  4. Run:
     ```
     peer { ipv6-address | group-name } next-hop-local
     ```
     The local address is configured as the next-hop address for route advertisement.

     In certain networks, to ensure that an IBGP peer can find the correct next hop, you can configure the local device to change the next-hop address of a route to be its own address when the local device advertises the route to its IBGP peer.

     By default, a device does not change the next-hop address when advertising a route to its IBGP peer.

     NOTE

     If BGP load balancing is configured, the local router changes the next-hop address to be its own address when advertising routes to IBGP peer groups, regardless of whether the **peer next-hop-local** command is used.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.

- Not to change the next-hop address when advertising a route to an EBGP peer.

  Do as follows on a PE:

  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv6-family vpnv6 [ unicast ]
```

The BGP-VPNv6 sub-address family view is displayed.

4. Run:

```
peer { ipv6-address | group-name } next-hop-invariable
```

The PE is configured not to change the next-hop address when advertising a route to an EBGP peer.

By default, PEs residing in different ASs set up EBGP relationships between each other and they will change the next-hop address when advertising routes to their EBGP peers.

5. Run:

```
commit
```

The configuration is committed.

----End

9.7.6 Setting the AS_Path Attribute

The AS_Path attribute is used to prevent routing loops and control route selection.

Procedure

- Set the AS_Path attribute in the IPv6 address family view.

  Do as follows on a BGP4+ device:

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     ipv6-family unicast
     ```

     The IPv6 unicast address family view is displayed.

  4. Run the following command as needed.

     - To allow repeated local AS numbers, run:

       ```
       peer { ipv6-address | group-name } allow-as-loop [ number ]
       ```

     - To exclude the AS_Path attribute from being a route selection rule, run:

       ```
       bestroute as-path-ignore
       ```

     - To allow the AS_Path attribute to carry only the public AS number, run:

       ```
       peer { ipv6-address | group-name } public-as-only
       ```

     The commands in Step 4 are optional and can be used in random order.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure a fake AS number.

  Do as follows on a BGP4+ device:

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     peer { ipv6-address | group-name } fake-as fake-as-number [ dual-as ]
     ```

     A fake AS number is configured.

     The actual AS number can be hidden by using this command. EBGP peers in other ASs can only learn this fake AS number of the BGP4+ device. This means that the fake AS number is used for the BGP4+ device when it is being specified on the peers in other ASs.

     NOTE

     This command is applicable to EBGP peers only.

  4. Run:

     ```
     commit
     ```

     The configuration is committed.

- Replace the AS number in the AS_Path attribute.

  If the AS_Path attribute of a route contains the AS number of the peer, this number needs to be replaced with the local AS number before the route is advertised. On a VPN, if CEs at different sites use the same AS number, you must run the **peer substitute-as** command.

  CAUTION

  Exercise caution when running the **peer substitute-as** command because improper use of the command may cause routing loops.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

     ```
     ipv6-family vpn-instance vpn-instance-name
     ```

The VPNv6 instance view is displayed.

4. Run:

   ```
   peer { ipv6-address | group-name } substitute-as
   ```

   The AS number in the AS_Path attribute is replaced.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.7.7 Checking the Configuration

After BGP4+ route attributes are configured, you can view information about the route attributes.

Prerequisite

The configurations of BGP4+ route attributes are complete.

Procedure

- Run the **display bgp ipv6 paths** [*as-regular-expression*] command to check the AS_Path information.
- Run the **display bgp ipv6 routing-table different-origin-as** command to check the routes that have the same destination address but different source ASs.
- Run the **display bgp ipv6 routing-table regular-expression** *as-regular-expression* command to check the routes that match the AS regular expression.
- Run the **display bgp ipv6 routing-table community** [*aa:nn* &<1-13>] [**internet** | **no-advertise** | **no-export** | **no-export-subconfed**] * [**whole-match**] command to check the routes carrying the specified BGP4+ community attribute in the routing table.
- Run the **display bgp ipv6 routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check the routes that meet the specified BGP4+ community filtering conditions.

----End

Example

# Run the **display bgp ipv6 routing-table** command, and you can view information about BGP4+ routes. For example:

```
<HUAWEI> display bgp ipv6 routing-table

 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 1
 *>  Network  : 7::                                      PrefixLen : 64
     NextHop  : 10::2                                    LocPrf    :
     MED      : 150                                      PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
```

# Run the **display bgp ipv6 paths** command, and you view the AS_Path information. For example:

```
<HUAWEI> display bgp ipv6 paths

Total Number of Routes: 6
Total Number of Paths: 4

    Address       Refcount  MED         Path/Origin
    0x41036F0     1         0            i
    0x4103758     1         0            i
    0x41037C0     1         0            i
    0x41038F8     3         0            200i
```

9.8 Configuring BGP4+ Routing Policies

BGP4+ routing policies can be configured to flexibly control the sending and receiving of routes.

Applicable Environment

Routing policies can set or re-set BGP4+ route attributes by using some predefined conditions, which provides a flexible and effective method to control BGP4+ route selection. The sending and receiving of routes can be flexibly controlled by applying BGP4+ routing policies.

Based on the import (or export) routing policy specified by the peer, the associated import (or export) routing conditions (**if-match** clauses) can be configured to filter routes, and **apply** clauses can be configured to set or modify route attributes. The routes that match the routing policy will be received (or sent).

Pre-configuration Tasks

Before configuring BGP4+ routing policies, complete the following tasks:

- Configuring IP addresses for interfaces to ensure IP connectivity between neighboring nodes
- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Choose certain configuration tasks from the following (except "Checking the Configuration") as needed in the applicable environment.

9.8.1 Configuring Related BGP4+ Access Lists

BGP4+ access lists can be used when BGP4+ status is displayed or routing policies are configured.

Context

BGP4+ has two specific access lists, the AS_Path filter and the community filter. Both of them can be used when BGP4+ status is displayed or routing policies are configured.

- AS_Path filter

  The AS_Path filter filters BGP4+ routes by the AS_Path attribute. Multiple rules (permit or deny) can be specified in a filter.

- Community filter

  The community filter consists of multiple community attribute lists. There are two types of community attribute lists, the standard community access list and the extended community access list.

Procedure

- Configure the AS_Path filter.
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     ip as-path-filter as-path-filter-number { permit | deny } regular-expression
     ```

     The AS_Path filter is configured.

     After the **peer as-path-filter** command is used to apply a routing policy to BGP4+ routes, the AS-Path filter filters out unqualified routes.

     The AS_Path filter uses the regular expression to define matching rules. A regular expression consists of the following parts:

     - Metacharacter: defines matching rules.
     - General character: defines matching objects.

Table 9-1 Description of metacharacters

Metacharacter	Description
\	Escape character.
.	Matches any single character except "\n", including spaces.
*	An asterisk after a character will match zero or more occurrences of that character.
+	A plus sign after a character will match one or more occurrences of that character.
|	Matches either expression it separates.
^	At the beginning of a line a circumflex matches the start of a line.
$	At the end of a line a dollar sign matches the end of a line.
[xyz]	Characters in square brackets will match any one of the enclosed characters.
[^xyz]	A circumflex at the start of an expression within brackets will match any character except one of the enclosed characters.
[a-z]	Matches any character within the specified range.
[^a-z]	Matches any character out of the specified range.

Metacharacter	Description
{n}	"n" is a non-negative integer. It refers to repeating exactly n times.
{n,}	"n" is a non-negative integer. It refers to repeating at least n times.
{n,m}	"m" and "n" are both non-negative integers, and "n" is equal to or smaller than "m". It refers to repeating the times ranging from "n" to "m". Note that there is no space between "n" and comma, or between comma and "m".

For example, ^10 indicates that only the AS_Path attribute starting with 10 is matched. A circumflex (^) indicates that the beginning of a character string is matched.

Multiple rules, permit or deny, can be specified in a filter. The relationship between theses rules is "OR". This means that if a route meets one of the matching rules, it will pass the AS_Path-based filtering.

NOTE

For details on the regular expression, see the *HUAWEI NetEngine5000E Core Router Configuration Guide - Basic Configurations*.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure the community filter.

  Community filters are classified into two types, the standard community filter and the advanced community filter. The advanced community filter supports regular expressions and is more flexible than the standard community filter.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run the following command as needed.

     - To configure the standard community filter, run:

       ```
       ip community-filter { basic comm-filter-name { permit | deny }
       [ community-number | aa:nn ] * &<1-9> | basic-comm-filter-num { permit
       | deny } [ community-number | aa:nn ] * &<1-16> } [ internet | no-export-
       subconfed | no-advertise | no-export ] *
       ```

     - To configure the advanced community filter, run:

       ```
       ip community-filter { advanced comm-filter-name | adv-comm-filter-num }
       { permit | deny } regular-expression
       ```

  3. Run:

     ```
     commit
     ```

     The configuration is committed.

- Configure the extended community filter.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   ip extcommunity-filter extcomm-filter-number { permit | deny } rt { as-number : nn | ipv4-address : nn } &<1-16>
   ```

   The extended community filter is configured.

   Multiple entries can be defined in an extended community filter. The relationship between the entries is "OR". A route that meets any of the matching rules will pass the filtering.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.8.2 Configuring a Route-Policy

A route-policy is used to match routes or route attributes, and to change route attributes when the matching rules are met.

Context

A route-policy is used to match routes or route attributes, and to change route attributes when the matching rules are met.

A route-policy consists of multiple nodes, and each node can comprise the following clauses:

- **if-match** clauses

  The clauses define the matching rules of a route-policy. The matching objects are route attributes.

- **apply** clauses

  The clauses specify actions. Configuration commands are run after a route satisfies the matching rules specified by the **if-match** clauses. The **apply** clauses can change certain route attributes.

NOTE

This section describes only the BGP4+ route-policy. For detailed information about route-policy configurations, see "Routing Policy Configuration."

Procedure

- Create a route-policy.

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     route-policy route-policy-name { permit | deny } node node
     ```

     The node of a route-policy is created, and the view of the route-policy is displayed.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure **if-match** clauses.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   route-policy route-policy-name { permit | deny } node node
   ```

   The route-policy view is displayed.

3. Run the following command to configure the **if-match** clause for the current node in the route-policy as needed.

   - To use the AS_Path attribute of BGP4+ routes as a match criterion, run:

     ```
     if-match as-path-filter as-path-filter-number &<1-16>
     ```

   - To use the community attribute of BGP4+ routes as a match criterion, run:
     - **if-match community-filter** { *basic-comm-filter-num* [**whole-match**] | *adv-comm-filter-num* }* &<1-16>
     - **if-match community-filter** *comm-filter-name* [**whole-match**]

   - To use the extended community attribute of BGP4+ routes as a match criterion, run:

     ```
     if-match extcommunity-filter extcomm-filter-number &<1-16>
     ```

   The commands in Step 3 can be configured in random order. A node may have multiple or no **if-match** clauses.

   NOTE

   - The relationship between the **if-match** clauses for a node of a route-policy is "AND". A route must meet all the matching rules before the action defined by the **apply** clause is performed.
   - If no **if-match** clause is specified, all the routes are matched.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

- Configure the **apply** clause.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   route-policy route-policy-name { permit | deny } node node
   ```

   The route-policy view is displayed.

3. Run the following command as needed to configure the **apply** clause for the current node in the route-policy.

   - To replace the AS number in the AS_Path attribute with the specified AS number or add the specified AS number to the AS_Path attribute of BGP4+ routes, run:

apply as-path *as-number* &<1-10> [**additive**]

- To delete the specified community attribute of BGP4+ routes, run:

  apply comm-filter *comm-filter-number* **delete**

  TIP

  > The **apply comm-filter delete** command deletes the specified community attribute from routes. Running the **ip community-filter** command specifies only one community attribute each time. To delete more than one community attribute, run the **ip community-filter** command multiple times. If multiple community attributes are specified in one filter, none of them can be deleted. For examples, see the *HUAWEI NetEngine5000E Core Router Command Reference*.

- To delete the community attribute of BGP4+ routes, run:
  apply community none

- To set the community attribute for BGP4+ routes, run:
  apply community { { *community-number* | *aa:nn* } &<1-32> | **internet** | **no-advertise** | **no-export** | **no-export-subconfed** }* [**additive**]

- To set the MED, run:
  apply cost { [*apply-type*] *cost* | **inherit** }

- To set the MED value of BGP4+ routes as the IGP cost of the next hop, run:
  apply cost-type internal

- To set the extended community attribute (Route-Target) for BGP4+ routes, run:
  apply extcommunity rt { *as-number:nn* | *ipv4-address:nn* } [**additive**]

- To set the local preference for BGP4+ routes, run:
  apply local-preference *preference*

- To set the Origin attribute for BGP4+ routes, run:
  apply origin { **igp** | **egp** *as-number* | **incomplete** }

- To set the preferred value for BGP4+ routes, run:
  apply preferred-value *preferred-value*

- To set dampening parameters for EBGP routes, run:

  apply dampening *half-life-reach reuse suppress ceiling*

The commands in Step 3 can be configured in random order.

4. Run:
   commit

   The configuration is committed.

----End

9.8.3 Configuring a Policy for Advertising BGP4+ Routes

BGP4+ filters the imported routes, and only the routes that meet the matching rules are added to the local BGP4+ routing table and advertised to BGP4+ peers.

Context

BGP4+ can apply a routing policy to all the routes to be advertised or only the routes to be advertised to a certain peer (group).

Procedure

- Configure BGP to filter all the routes to be advertised.
  1. Run:
     system-view

The system view is displayed.

2. Run:

   bgp *as-number*

   The BGP view is displayed.

3. Run:

   ipv6-family unicast

   The IPv6 unicast address family view is displayed.

4. Run:

   filter-policy { *acl6-number* | **acl6-name** *acl6-name* | **ipv6-prefix** *ipv6-prefix-name* } **export** [*protocol* [*process-id*]]

   The routes to be advertised are filtered.

   BGP4+ filters the routes imported by using the **import-route** command, and only the routes that meet the matching rules are added to the local BGP4+ routing table and advertised to BGP4+ peers.

   If *protocol* is specified, only the routing information of a specified protocol is filtered. If *protocol* is not specified, all BGP routes to be advertised are filtered, including the routes imported by using the **import-route** and **network** commands.

5. Run:

   commit

   The configuration is committed.

- Apply a routing policy to the routes to be advertised to a certain peer (group).

1. Run:

   system-view

   The system view is displayed.

2. Run:

   bgp *as-number*

   The BGP view is displayed.

3. Run:

   ipv6-family unicast

   The IPv6 unicast address family view is displayed.

4. Run the following command as needed to configure BGP4+ to use a specified filter to filter the routes to be advertised to a peer.

   - To use an ACL for route filtering, run:

     peer { *ipv4-address* | *ipv6-address* | *group-name* } **filter-policy** { *acl6-number* | **acl6-name** *acl6-name* } **export**

   - To use the AS_Path attribute for route filtering, run:

     peer { *ipv4-address* | *ipv6-address* | *group-name* } **as-path-filter** *as-path-filter-number* **export**

   - To use a prefix list for route filtering, run:

     peer { *ipv4-address* | *ipv6-address* | *group-name* } **ipv6-prefix** *ipv6-prefix-name* **export**

   - To use a route-policy for route filtering, run:

     peer { *ipv4-address* | *ipv6-address* | *group-name* } **route-policy** *route-policy-name* **export**

A peer group and its members can use different export policies when advertising routes. This means that each member in a peer group can select its own policy when advertising routes.

5. Run:
   ```
   commit
   ```
   The configuration is committed.

----End

9.8.4 Configuring a Policy for Receiving BGP4+ Routes

BGP4+ filters received routes by using a policy. Only the routes that match the policy can be installed into a routing table.

Context

BGP4+ can apply a routing policy to all received routes or only routes received from a specific peer (group).

If a device is under malicious attacks or some network configurations are incorrect, the device will receive a large number of routes from its BGP4+ peers, consuming lots of device resources. Therefore, the administrator must limit resources consumed by BGP4+ devices based on device capacities. BGP4+ provides peer-specific route control to limit the number of routes sent from a peer or peer group.

Procedure

- Configure BGP4+ to filter all received routes.
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     bgp as-number
     ```
     The BGP view is displayed.
  3. Run:
     ```
     ipv6-family unicast
     ```
     The IPv6 unicast address family view is displayed.
  4. Run:
     ```
     filter-policy { acl6-number | acl6-name acl6-name | ipv6-prefix ipv6-
     prefix-name } import
     ```
     A policy is configured to filter all received BGP4+ routes.

     Only the routes that match the policy are installed into a routing table.
  5. Run:
     ```
     commit
     ```
     The configuration is committed.
- Configure BGP4+ to filter the routes received from a specific peer (group).
  1. Run:
     ```
     system-view
     ```

The system view is displayed.

2. Run:

   ```
   bgp as-number
   ```

   The BGP view is displayed.

3. Run:

   ```
   ipv6-family unicast
   ```

   The IPv6 unicast address family view is displayed.

4. Run the following commands as needed to configure BGP4+ to use different filters to filter routes received from peers:

   - To use an ACL for route filtering, run:

     ```
     peer { ipv4-address | ipv6-address | group-name } filter-policy { acl6-number | acl6-name acl6-name } import
     ```

   - To use the AS_Path filter for route filtering, run:

     ```
     peer { ipv4-address | ipv6-address | group-name } as-path-filter as-path-filter-number import
     ```

   - To use an IP prefix list for route filtering, run:

     ```
     peer { ipv4-address | ipv6-address | group-name } ip-prefix ipv6-prefix-name import
     ```

   - To use a route-policy for route filtering, run:

     ```
     peer { ipv4-address | ipv6-address | group-name } route-policy route-policy-name import
     ```

   A peer group and its members can use different inbound policies when receiving routes. This means that each member in a peer group can select its own policy to filter received routes.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.8.5 Configuring BGP4+ Soft Resetting

BGP4+ soft resetting allows the system to refresh a BGP4+ routing table dynamically without tearing down any BGP4+ connection if routing policies are changed.

Procedure

- Enable the route-refresh capability.

  Do as follows on a BGP4+ device:

  1. Run:

     ```
     system-view
     ```

     The system view is displayed.

  2. Run:

     ```
     bgp as-number
     ```

     The BGP view is displayed.

  3. Run:

```
peer { ipv4-address | ipv6-address | group-name } capability-advertise
route-refresh
```

The route-refresh capability is enabled.

Assume that all BGP4+ routers are enabled with the route-refresh capability. If a BGP4+ routing policy is changed, the local router sends route-refresh messages to its peers. After receiving the messages, the peers resend their routing information to the local router. Based on the received routing information, the local router can refresh its BGP4+ routing table dynamically and apply the new routing policy, without tearing down any BGP4+ connections.

By default, the route-refresh capability is enabled.

4. Run:

```
commit
```

The configuration is committed.

- Configure a BGP4+ device to store all the routing updates received from its peers.

Do as follows on a BGP4+ device:

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bgp as-number
```

The BGP view is displayed.

3. Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

4. Run:

```
peer { ipv4-address | ipv6-address | group-name } keep-all-routes
```

The device is configured to store all routing updates received from its peers.

After this command is used, all routing updates sent by a specified peer are stored, regardless of whether a filtering policy is used. When the local routing policy is changed, the routing updates can be used to regenerate BGP4+ routes.

5. Run:

```
commit
```

The configuration is committed.

- Softly reset a BGP4+ connection.

Do as follows on a BGP4+ device:

1. Run:

```
refresh bgp ipv6 { all | ipv4-address | ipv6-address | group group-name |
external | internal } { export | import }
```

A BGP4+ connection is softly reset.

A BGP4+ connection must be softly reset in the user view.

----End

9.8.6 Checking the Configuration

After configuring BGP4+ routing policies, you can view routes that are advertised and received by BGP4+.

Prerequisite

The configurations of BGP4+ routing policies are complete.

Procedure

- Run the **display bgp ipv6 network** command to check the routes imported by using the **network** command.
- Run the **display bgp ipv6 routing-table as-path-filter** *as-path-filter-number* command to check the routes that match a specified AS_Path filter.
- Run the **display bgp ipv6 routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check the routes that match a specified BGP4+ community filter.
- Run the **display bgp ipv6 routing-table peer** *ipv6-address* { **advertised-routes** | **received-routes** } [**statistics**] command to view the routes advertised to or received from a specified BGP4+ peer.

----End

Example

Run the **display bgp network** command. You can view the routes imported by using the **network** command. For example:

```
<HUAWEI> display bgp ipv6 network
  BGP Local Router ID is 5.5.5.5
  Local AS Number is 100(PublicV6)
  Network          Prefix           Route-policy
  100::            64
  200::            64
```

9.9 Configuring BGP4+ Load Balancing

Configuring BGP4+ load balancing better utilizes network resources.

Applicable Environment

BGP4+ load balancing can be performed among routes that have the same AS_Path attribute and conform to the first 10 rules in "Policies for BGP4+ Route Selection" in **9.2 BGP4+ Features Supported by the NE5000E**.

Pre-configuration Tasks

Before configuring BGP4+ load balancing, complete the following task:

- **Configuring Basic BGP4+ Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
maximum load-balancing number
```

The number of routes for BGP4+ load balancing is set.

By default, the number of routes for BGP4+ load balancing is 1.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the configuration is complete, you can check whether the configuration has taken effect.

- Run the **display bgp ipv6 routing-table** *ipv6-address prefix-length* command to check routes in the BGP4+ routing table.

Run the **display bgp ipv6 routing-table** command. The command output shows that two BGP4 + routes have been selected for load balancing. For example:

```
<HUAWEI> display bgp ipv6 routing-table 8::1 64

 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Paths : 2 available, 1 best, 2 select
 BGP routing table entry information of 8::/64:
 From: 20:1::2 (2.2.2.2)
 Route Duration: 0d00h02m00s
 Direct Out-interface: Ethernet3/0/0
 Original nexthop: 20:1::2
 AS-path 300 200, origin igp, pref-val 0, valid, external, best, select, pre 255
 Advertised to such 2 peers:
    20:1::2
    20:2::2

 BGP routing table entry information of 8::/64:
 From: 20:2::2 (3.3.3.3)
 Route Duration: 0d00h02m00s
 Direct Out-interface: Ethernet3/0/1
 Original nexthop: 20:2::2
 AS-path 300 200, origin igp, pref-val 0, valid, external, select, pre 255, not
selected for router ID
 Not advertised to any peers yet
```

9.10 Configuring BGP4+ Route Dampening

Configuring BGP4+ route dampening suppresses instable BGP4+ routes.

Applicable Environment

BGP4+ route dampening is designed to suppress instable routes and improve network stability. After being configured with BGP4+ route dampening, a BGP4+ device does not add any instable routes to its BGP4+ routing table or advertise them to its BGP4+ peers.

A primary cause of route instability is route flapping. A route is considered to be flapping when it repeatedly appears and then disappears in the routing table. BGP4+ is applied to complex networks where routes change frequently. Frequent route flapping consumes lots of bandwidth and CPU resources and even seriously affects network operations. To prevent the impact of frequent route flapping, BGP4+ uses route dampening to suppress instable routes.

Specified-requirement route dampening is a type of route dampening that is used to differentiate BGP4+ routes based on routing policies. This allows BGP4+ to use different route dampening parameters to suppress instable routes. Different route dampening parameters can aslo be configured for different nodes in the same route-policy. When route flapping occurs, BGP4+ can use specific route dampening parameters to suppress the routes that match the route-policy. For example, on a network, a long dampening period of time can be set for routes with a long mask, and a short dampening period of time can be set for routes with a short mask (such as an 8-bit mask).

Pre-configuration Tasks

Before configuring BGP4+ route dampening, complete the following task:

- **Configuring Basic BGP4+ Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
dampening [ half-life-reach reuse suppress ceiling | route-policy route-policy-name ] *
```

BGP4+ route dampening parameters are set.

NOTE

The **dampening** command is valid only for EBGP routes.

The value of *suppress* must be greater than that of *reuse* and smaller than that of *ceiling*.

If routes are differentiated based on policies and the **dampening** command is run to reference a route-policy, BGP4+ allow you to use different route dampening parameters to suppress different routes.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the configuration is complete, you can check whether the configuration has taken effect.

- Run the **display bgp ipv6 routing-table dampened** command to check dampened BGP4 + routes.
- Run the **display bgp ipv6 routing-table dampening parameter** command to check configured BGP4+ route dampening parameters.
- Run the **display bgp ipv6 routing-table flap-info** [**regular-expression** *as-regular-expression* | **as-path-filter** *as-path-filter-number* | *network-address* [{ *mask* | *mask-length* } [**longer-match**]]] command to check route flapping statistics.

Run the **display bgp ipv6 routing-table dampening parameter** command. You can view the configured values of BGP4+ route dampening parameters including **Maximum Suppress Time**, **Ceiling Value**, and **Reuse Value**. For example:

```
<HUAWEI> display bgp ipv6 routing-table dampening parameter
 Maximum Suppress Time(in second) : 3973
 Ceiling Value                    : 16000
 Reuse Value                      : 750
 HalfLife Time(in  second)        : 900
 Suppress-Limit                   : 2000
 Route-policy                     : dampen-policy
```

9.11 Configuring BGP4+ Connection Parameters

By configuring BGP4+ connection parameters, you can optimize BGP4+ network performance.

Applicable Environment

BGP4+ can use various timers to minimize the impact of interface flapping or route flapping.

After establishing a BGP4+ connection, two peers periodically send Keepalive messages to each other to detect the status of the BGP4+ peer relationship. If the router does not receive any Keepalive message or any other type of packets from its peer within a time of period called holdtime period, it considers the BGP4+ connection closed.

When establishing a BGP4+ connection, two peers compare their holdtime periods and use the shorter one. If the used holdtime period is 0, no Keepalive message will be transmitted and the hold timer status will not be detected.

CAUTION

If the timeout period of the hold timer changes, the BGP4+ connection may be interrupted for a short period of time. This is because the peers need to renegotiate the holdtime period.

Pre-configuration Tasks

Before configuring BGP4+ connection parameters, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Choose one or more configuration tasks (excluding "Checking the Configuration") as needed.

9.11.1 Configuring Timers for BGP4+ Peers

Configuring timers properly improves network performance. Changing the timeout periods of timers, however, will interrupt peer relationships.

Context

CAUTION

Changing the timeout period of a timer by using the **peer timer** command will interrupt peer relationships between routers. Therefore, exercise caution when changing the timeout period of a timer.

Do as follows on a BGP4+ device:

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
peer { ipv6-address | group-name } timer keepalive keepalive-time hold hold-time
```

The timeout periods of the holdtime timer and Keepalive timer are set on a peer or peer group.

The value of *hold-time* must be at least three times that of *keepalive-time*. By default, the value of *keepalive-time* is 60 seconds and the value of *hold-time* is 180 seconds.

When setting the values of *keepalive-time* and *hold-time*, note the following points:

- The values of *keepalive-time* and *hold-time* cannot both be set to 0. Otherwise, the BGP timers will fail to function. This means that BGP4+ cannot detect link faults based on the timers.
- The value of *hold-time* such as **timer keepalive 1 hold 65535** is much greater than that of *keepalive-time*. If the value of *hold-time* is too large, BGP4+ cannot detect link faults rapidly.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.11.2 Enabling the Function of Fast Resetting EBGP Connections

After the function of fast resetting EBGP connections is enabled, BGP4+ rapidly detects EBGP link faults and resets EBGP connections on related interfaces.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ebgp-interface-sensitive
```

The function of fast resetting EBGP connections is enabled.

After this function is enabled, BGP4+ rapidly detects EBGP link faults and resets EBGP connections on related interfaces.

When an interface on which a BGP4+ connection is established alternates between Up and Down states, run the **undo ebgp-interface-sensitive** command to prevent repeated re-establishment and deletion of the BGP4+ session in the case of route flapping. This saves network bandwidth.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.11.3 Checking the Configuration

After configuring BGP4+ connection parameters, you can view information about BGP4+ peers or peer groups.

Prerequisite

The configurations of BGP4+ connection parameters are complete.

Procedure

- Run the **display bgp ipv6 peer** [**verbose**] command to check information about BGP4+ peers.

----End

Example

Run the **display bgp ipv6 peer verbose** command. You can view information about BGP4+ peers. For example:

```
<HUAWEI> display bgp ipv6 peer verbose

         BGP Peer is 10::1,  remote AS 10
         Type: EBGP link
         BGP version 4, Remote router ID 2.2.2.2

  Group ID : 1
         BGP current state: Established, Up for 00h00m17s
         BGP current event: RecvUpdate
         BGP last state: Established
         BGP Peer Up count: 1
         Port: Local - 3363        Remote - 179
         Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Received  : Active Hold Time: 180 sec
         Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
         Peer optional capabilities:
         Peer supports bgp multi-protocol extension
         Peer supports bgp route refresh capability
         Peer supports bgp 4-byte-as capability
         Address family IPv6 Unicast: advertised and received
 Received: Total 3 messages
                  Update messages                1
                  Open messages                  1
                  KeepAlive messages             1
                  Notification messages          0
                  Refresh messages               0
 Sent: Total 3 messages
                  Update messages                1
                  Open messages                  1
                  KeepAlive messages             1
                  Notification messages          0
                  Refresh messages               0
 Authentication type configured: None
 Last keepalive received:2011-02-28 03:21:18
 Minimum route advertisement interval is 30 seconds
 Optional capabilities:
 Route refresh capability has been enabled
 4-byte-as capability has been enabled
 Multi-hop ebgp has been enabled
 Peer Preferred Value: 0
 Routing policy configured:
 No routing policy is configured
```

9.12 Configuring BGP4+ RRs

BGP4+ RRs resolve the problem of full-mesh connections between multiple IBGP peers, which reduces network costs.

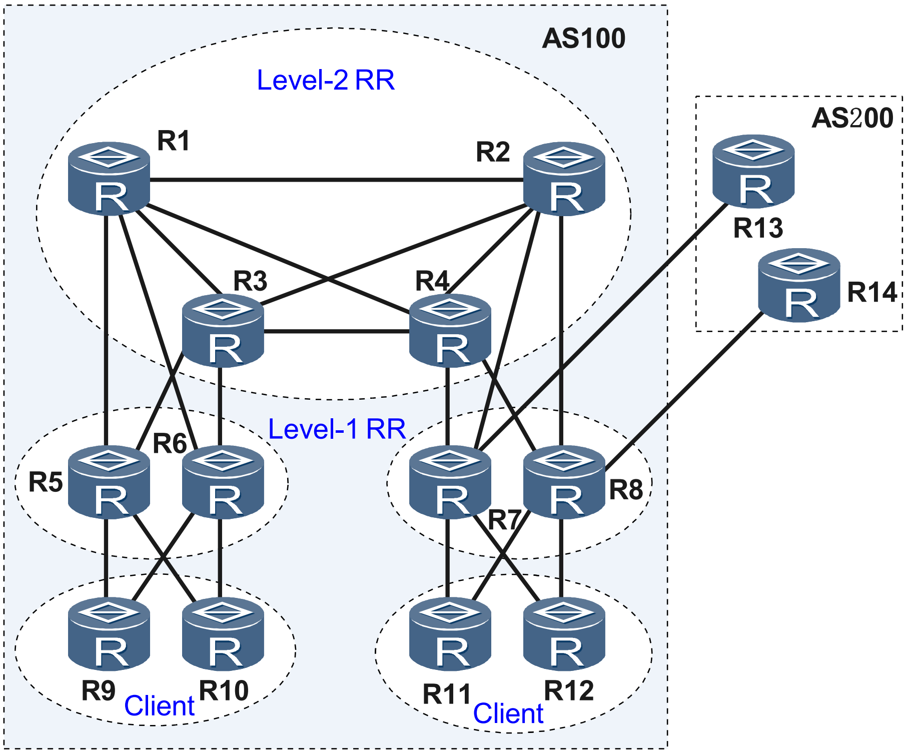
Applicable Environment

Full-mesh connections need to be established between IBGP peers in an AS to ensure the connectivity between IBGP peers. When there are many IBGP peers, establishing a fully-meshed network is costly. An RR can be used to resolve this problem.

Using RRs reduces the total number of IBGP connections. On a large network, however, multiple RRs need to be configured to reduce the number of clients of each RR. Therefore, there are still a large number of IBGP connections on the network because full-mesh connections need to be established between the RRs. Hierarchical BGP RR networking is introduced to further reduce the number of IBGP connections.

Figure 9-2 shows a typical hierarchical RR networking. R1, R2, R3, and R4 function as level-2 RRs; R5, R6, R7, and R8 function as level-1 RRs and the clients of level-2 RRs. Level-2 RRs are not the clients of any RR and must be fully meshed. Level-1 RRs function as the clients of level-2 RRs and do not need to be fully meshed.

Figure 9-2 Networking diagram of hierarchical RRs



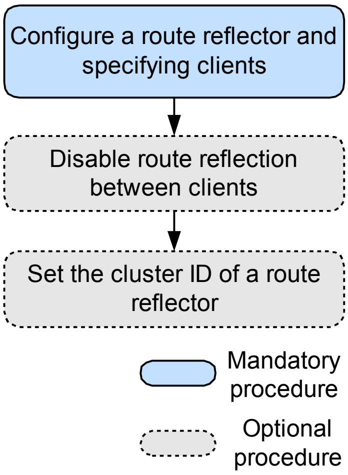
Pre-configuration Tasks

Before configuring BGP4+ RRs, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Figure 9-3 Flowchart for configuring BGP4+ RRs


Configure a route reflector and specifying clients
Disable route reflection between clients
Set the cluster ID of a route reflector
Mandatory procedure
Optional procedure

Related Tasks

9.19.2 Example for Configuring BGP4+ Route Reflection

9.12.1 Configuring an RR and Specifying Its Clients

RRs reflect routes between clients, and therefore IBGP connections do not need to be established between the clients.

Context

In an AS, one router functions as an RR, and the other routers function as its clients. The clients establish IBGP connections with the RR. The RR and its clients form a cluster. The RR transmits or reflects routes among clients, but the clients do not need to establish any IBGP connections between each other.

An RR is easy to configure because it needs to be configured only on the router that functions as a reflector and clients do not need to know that they are clients.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
peer { ipv6-address | ipv4-address | group-name } reflect-client
```

An RR and its clients are configured.

The router where the **peer reflect-client** command is run functions as the RR, and specified peers or peer groups function as the clients.

NOTE

reflect-client configured in an address family is valid in this family address and cannot be inherited by other address families.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.12.2 (Optional) Disabling Route Reflection Between Clients

If the clients of an RR are fully meshed, you can disable route reflection between the clients to reduce network cost.

Context

On some networks, if the clients of an RR have established IBGP peer relationships with each other, they can directly exchange routing information. Route reflection between the clients is unnecessary. Route reflection can be disabled to reduce network cost.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
undo reflect between-clients
```

Route reflection between clients is disabled.

By default, route reflection between clients is enabled.

If the clients of an RR have been fully meshed, you can run the **undo reflect between-clients** command to disable route reflection between the clients to reduce the network cost. The **undo reflect between-clients** command is run only on RRs.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.12.3 (Optional) Configuring a Cluster ID for RRs

If a cluster has multiple RRs, you can configure the same cluster ID for these RRs to prevent routing loops.

Context

Under some circumstances, more than one RR needs to be configured in a cluster to improve network reliability and prevent single-point failures. The same cluster ID needs to be configured for all the RRs in the cluster to reduce the number of routes received by each RR. This reduces network cost.

NOTE

To allow clients to receive routes reflected by RRs, ensure that the cluster ID of the RRs is different from the router ID of any client. If the cluster ID of the RRs is the same as the router ID of a client, the client will discard received routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The IPv6 unicast address family view is displayed.

Step 4 Run:

```
reflector cluster-id cluster-id
```

A cluster ID is configured for RRs.

If a cluster has multiple RRs, you can use this command to set the same cluster ID for these RRs.

The **reflector cluster-id** command is run only on RRs.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.12.4 Checking the Configuration

After configuring BGP4+ RRs, you can view information about BGP4+ routes and peer groups.

Prerequisite

The BGP4+ RR configurations are complete.

Procedure

- Run the **display bgp ipv6 peer** [**verbose**] command to check detailed information about peers and check whether an RR and its clients are successfully configured.
- Run the **display bgp ipv6 routing-table** command to check the routes to an RR.

----End

9.13 Configuring BFD for BGP4+

BFD for BGP4+ provides BGP4+ with a mechanism for quickly detecting faults, which speeds up network convergence.

Applicable Environment

BFD is dedicated to fast detection of forwarding faults to ensure QoS of voice, video, and other video-on-demand services on a network. It enables service providers to provide users with required VoIP and other real-time services of high availability and scalability.

A BGP4+ device periodically sends Keepalive packets to detect the status of its peers. The detection, however, takes more than one second. When the data transmission rate reaches the level of Gbit/s, such slow detection will cause a large amount of data to be lost. As a result, the requirement for high reliability of carrier-class networks cannot be met.

Therefore, BFD for BGP4+ is introduced to fast detect faults on the links between BGP4+ peers and then notify BGP4+ of the faults. This increases the network convergence speed.

Pre-configuration Tasks

Before configuring BFD for BGP4+, complete the following tasks:

- Configuring link layer protocol parameters and IP addresses for interfaces to ensure that the link layer protocol on the interfaces is Up
- **Configuring Basic BGP4+ Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bfd
```

BFD is enabled globally.

Step 3 Run:

```
quit
```

Return to the system view.

Step 4 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 5 Run:

```
peer { group-name | ipv6-address } bfd enable
```

BFD is enabled on a peer or peer group and a BFD session is established by using default BFD parameters.

Step 6 (Optional) Run:

```
peer { group-name | ipv6-address | ipv4-address } bfd { min-tx-interval min-tx-
interval | min-rx-interval min-rx-interval | detect-multiplier multiplier } *
```

Various parameters used for establishing a BFD session are set.

If BFD is configured on a peer group, the peers that are in the peer group but are not configured with the **peer bfd block** command establish BFD sessions.

NOTE

- A BFD session can be established only when the corresponding BGP session is in the Established state.
- The configuration of a peer takes precedence over that of its peer group. If BFD is not configured on a peer while its peer group is enabled with BFD, the peer inherits the BFD configurations of its peer group.

Step 7 (Optional) Run:

```
peer ipv6-address bfd block
```

A peer is prohibited from inheriting the BFD function of its peer group.

If peers are added to a peer group enabled with BFD, the peers inherit the BFD function from the peer group and establish BFD sessions with other devices. If you do not want a peer to inherit BFD of its peer group, you can prevent the peer from inheriting BFD of its peer group.

NOTE

The **peer** *ipv6-address* **bfd block** command and the **peer** *ipv6-address* **bfd enable** command are mutually exclusive. After being configuerd with the **peer bfd block** command, a device automatically deletes the BFD session established with a specified peer.

Step 8 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the configuration is complete, you can check whether the configuration has taken effect.

- Run the **display bgp ipv6 bfd session** { [**vpnv6 vpn-instance** *vpn-instance-name*] **peer** *ipv6-address* | **all** } command to check BFD sessions established by BGP4+.

- Run the **display bgp** [**vpnv6 vpn-instance** *vpn-instance-name*] **peer** [*ipv6-address*] [**verbose**] command to check BGP4+ peers.

Run the **display bgp ipv6 bfd session all** command. You can view the BFD sessions established by BGP4+. For example:

```
<HUAWEI> display bgp ipv6 bfd session all
--------------------------------------------------------------------------------
  Local_Address        Peer_Address         Interface
  8::1                 8::2                 Pos1/0/0
  Tx-interval(ms)      Rx-interval(ms)      Multiplier  Session-State
  100                  100                  4           Up
--------------------------------------------------------------------------------
```

Related Tasks

9.19.3 Example for Configuring BFD for BGP4+

9.14 Configuring BGP4+ Auto FRR

As a protection measure against link faults, BGP4+ Auto Fast Reroute (FRR) is applicable to the network topology with primary and backup links. BGP4+ Auto FRR is applicable to services that are very sensitive to packet loss and delays.

Applicable Environment

BGP4+ Auto FRR is applicable to services that are very sensitive to packet loss and delays. After being enabled with BGP4+ Auto FRR, a device uses the optimal route selected from the routes which carry the same prefix and are learned from multiple peers as the primary link to forward packets and uses the sub-optimal route as the backup link. When the primary link becomes faulty, the device rapidly responds to the notification that the BGP4+ route becomes unreachable, and then switches traffic from the primary link to the backup link.

Pre-configuration Tasks

Before configuring BGP4+ Auto FRR, complete the following tasks:

- Configuring static route or an IGP to make devices routable.
- **Configuring Basic BGP4+ Functions**

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The BGP-IPv6 unicast address family view is displayed.

Step 4 Run:

```
auto-frr
```

BGP4+ Auto FRR is enabled for unicast routes.

By default, BGP4+ Auto FRR is not enabled for any unicast routes.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the configuration is complete, you can check whether the configuration has taken effect.

Run the **display ipv6 routing-table** *ipv6-address prefix-length* [**longer-match**] [**verbose**] command to check the backup forwarding information of routes in the routing table of a device.

Run the **display ipv6 routing-table** *ipv6-address prefix-length* [**longer-match**] [**verbose**] command. You can view backup routes. For example:

```
<HUAWEI> display ipv6 routing-table 1::1 64 verbose
Routing Table : _public_
Summary Count : 1

 Destination  : 1::                             PrefixLength : 64
 NextHop      : 10::1                           Preference   : 255
 Neighbour    : ::                              ProcessID    : 0
 Label        : NULL                            Protocol     : BGP
 State        : Active Adv Relied               Cost         : 80
 Entry ID     : 0                               EntryFlags   : 0x00000000
 Reference Cnt: 0                               Tag          : 0
 IndirectID   : 0xb06                           Age          : 10sec
 RelayNextHop : 10::1                           TunnelID     : 0x0
 Interface    : Pos3/1/1                        Flags        : RD
 BkNextHop    : 9:1::2                        BkInterface  : Ethernet3/0/1
 BkLabel      : NULL                            BkTunnelID   : 0x0
 BkPETunnelID : 0x0                             BkIndirectID : 0xb07
```

9.15 Configuring BGP4+ GR Helper

A BGP4+ GR helper helps its neighbor complete BGP4+ GR.

Applicable Environment

BGP4+ restart causes re-establishment of peer relationships and traffic interruption.

BGP4+ GR needs to be enabled to prevent traffic interruption in the event of BGP4+ restart. A GR restarter and its BGP4+ peer negotiate the GR capability to establish a GR-capable BGP4+ session.

NOTE

Currently, the system supports only the GR helper function.

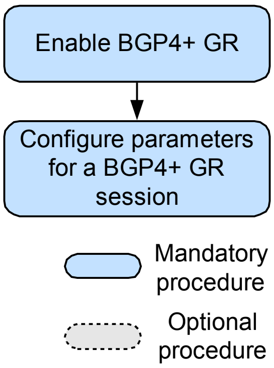
Pre-configuration Tasks

Before configuring BGP4+ GR helper, complete the following task:

- **Configuring Basic BGP4+ Functions**

Configuration Procedures

Figure 9-4 Flowchart for configuring BGP4+ GR helper



9.15.1 Enabling BGP4+ GR

Enabling or disabling BGP4+ GR may delete and re-establish all BGP4+ sessions and instances.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
ipv6-family unicast
```

The BGP-IPv6 unicast address family view is displayed.

Step 4 Run:

```
graceful-restart
```

BGP4+ GR is enabled.

By default, BGP4+ GR is disabled.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

9.15.2 Configuring BGP4+ GR Session Parameters

Changing the BGP4+ restart period re-establishes BGP4+ peer relationships.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
graceful-restart timer wait-for-rib timer
```

The length of time the restarting speaker and receiving speaker wait for End-of-RIB messages is set.

By default, the period for waiting for End-Of-RIB messages is 600s.

NOTE

You can adjust BGP4+ GR session parameter values as needed. Default BGP4+ GR session parameter values are recommended.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.15.3 Checking the Configuration

After configuring BGP4+ GR helper, you can view the BGP4+ GR status.

Prerequisite

The BGP4+ GR helper configurations are complete.

Procedure

- Run the **display bgp ipv6 peer verbose** command to check the BGP4+ GR status.

----End

Example

Run the **display bgp ipv6 peer verbose** command. You can view the BGP4+ GR status. For example:

```
<HUAWEI> display bgp ipv6 peer verbose

BGP Peer is 10:1::1,  remote AS 200
         Type: EBGP link
         BGP version 4, Remote router ID 4.4.4.4

  Group ID : 1
         BGP current state: Established, Up for 00h00m11s
         BGP current event: RecvUpdate
         BGP last state: Established
```

```
        BGP Peer Up count: 2
        Port: Local - 179        Remote - 36664
        Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
        Received  : Active Hold Time: 180 sec
        Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
        Peer optional capabilities:
        Peer supports bgp multi-protocol extension
        Peer supports bgp route refresh capability
        Peer supports bgp 4-byte-as capability
       Graceful Restart Capability: advertised
        Address family IPv6 Unicast: advertised and received
Received: Total 108 messages
                 Update messages                4
                 Open messages                  2
                 KeepAlive messages             102
                 Notification messages          0
                 Refresh messages               0
Sent: Total 124 messages
                 Update messages                4
                 Open messages                  18
                 KeepAlive messages             101
                 Notification messages          1
                 Refresh messages               0
Authentication type configured: None
Last keepalive received: 2010-08-20 10:32:32
Minimum route advertisement interval is 30 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

9.16 Configuring BGP4+ Authentication

BGP4+ authentication can be configured to enhance security of BGP4+ networks.

Applicable Environment

Authenticating the connections between BGP4+ peers can improve security of a BGP4+ network.

BGP4+ uses TCP as the transport layer protocol. Authentication is performed during the establishment of TCP connections to enhance BGP4+ security. TCP connections, not BGP4+ packets, are authenticated and the authentication is implemented by TCP. If authentication fails, no TCP connections can be established.

Pre-configuration Tasks

Before configuring BGP4+ authentication, complete the following task:

- Configuring Basic BGP4+ Functions

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
peer { group-name | ipv6-address } password { cipher | simple } password
```

The MD5 authentication password is set.

In MD5 authentication of BGP4+, MD5 authentication passwords are used only for TCP connections, and the authentication is performed by TCP. If the authentication fails, no TCP connections can be established.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

A peer relationship can be set between two peers that have the same authentication information. Check the peer relationship status.

```
[~HUAWEI] display bgp ipv6 peer
 BGP local router ID : 2.2.2.2
 Local AS number : 65009
 Total number of peers : 3                 Peers in established state : 3
  Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  9:1::2          4 65009        8        9     0 00:05:37 Established       2
  9:3::2          4 65009        2        2     0 00:00:09 Established       2
  10::2           4 65008        9        7     0 00:05:38 Established       2
```

9.17 Configuring BGP4+ 6PE

BGP4+ 6PE enables separated IPv6 networks to communicate by using the MPLS tunneling technology.

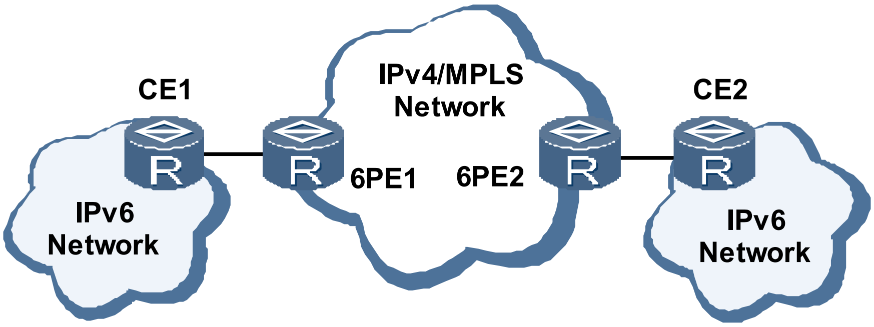
Applicable Environment

6PE enables IPv6 networks separated by IPv4/MPLS networks to communicate.

Separated IPv6 networks can be connected by using various tunneling techniques. A 6PE tunnel is established on Internet Service Provider's (ISP's) PEs that support the IPv4/IPv6 dual stack. The 6PE tunnel identifies IPv6 routes by label assigned by the Multiprotocol Border Gateway Protocol (MP-BGP), and implements IPv6 forwarding by using LSPs between PEs.

As shown in **Figure 9-5**, the IPv6 network where CE1 and CE2 reside are separated by an IPv4/ MPLS network. Configuring 6PE enables CE1 and CE2 to communicate across the IPv4 network.

Figure 9-5 Networking diagram for 6PE



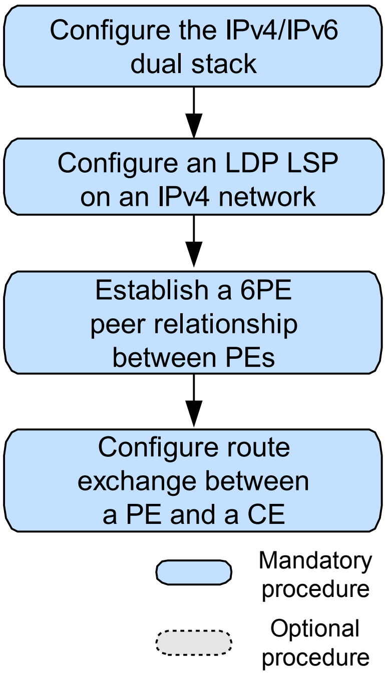
Pre-configuration Tasks

Before configuring BGP4+ 6PE, complete the following tasks:

- Connecting interfaces and setting parameters for the interfaces to ensure that the physical-layer status of the interfaces is Up
- Configuring the link layer protocol parameters for interfaces
- Ensure reachable routes on the IPv4/MPLS backbone network

Configuration Procedures

Figure 9-6 Flowchart for configuring BGP4+ 6PE


Configure the IPv4/IPv6 dual stack
Configure an LDP LSP on an IPv4 network
Establish a 6PE peer relationship between PEs
Configure route exchange between a PE and a CE
Mandatory procedure
Optional procedure

Related Tasks

9.19.4 Example for Configuring BGP4+ 6PE

9.17.1 Configuring the IPv4/IPv6 Dual Stack

The IPv4/IPv6 dual stack needs to be configured on the router at the edge of an IPv6 network and an IPv4 network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The view of the IPv4 network-side interface is displayed.

Step 3 Run:

```
ip address ipv4-address { mask | mask-length }
```

An IPv4 address is configured for the interface.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

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

The view of the IPv6 network-side interface is displayed.

Step 6 Run:

```
ipv6 enable
```

IPv6 is enabled on the interface.

Step 7 Run:

```
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length } eui-64 or
ipv6 address { ipv6-address prefix-length | ipv6-address/prefix-length }
```

An IPv6 address is configured for the interface.

Step 8 Run:

```
commit
```

The configuration is committed.

----End

9.17.2 Configuring an LDP LSP on an IPv4 Network

An LDP LSP is configured on an IPv4/MPLS backbone network to forward IPv6 packets.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
mpls lsr-id lsr-id
```

An LSR ID is configured.

Step 3 Run:

```
mpls
```

MPLS is enabled and the MPLS view is displayed.

Step 4 (Optional) Run:

```
lsp-trigger { all | host | ip-prefix ip-prefix-name | none }
```

An LSP triggering policy is configured.

Currently, the NE5000E automatically distributes labels to host routes with 32-bit masks. The command in this step needs to be run to distribute labels to routes of other types or specific routes.

Step 5 Run:

```
quit
```

Return to the system view.

Step 6 Run:

```
mpls ldp
```

LDP is enabled and the LDP view is displayed.

Step 7 Run:

```
quit
```

Return to the system view.

Step 8 Run:

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

The view of the IPv4 network-side interface is displayed.

Step 9 Run:

```
mpls
```

MPLS is enabled on the interface.

Step 10 Run:

```
mpls ldp
```

LDP is enabled on the interface.

Step 11 Run:

```
commit
```

The configuration is committed.

----End

9.17.3 Establishing a 6PE Peer Relationship Between PEs

6PE peers can exchange IPv6 routes learned from their attached CEs.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bgp as-number
```

The BGP view is displayed.

Step 3 Run:

```
peer ipv4-address as-number as-number
```

The IP address of the peer and the number of the AS where the peer resides are specified.

Step 4 Run:

```
peer ipv4-address connect-interface interface-type interface-number
```

The interface that is used to establish a connection with the remote PE is specified.

Step 5 Run:

```
ipv6-family unicast
```

The BGP-IPv6 unicast address family view is displayed.

Step 6 Run:

```
peer ipv4-address enable
```

A 6PE peer relationship is established.

Step 7 Run:

```
peer ipv4-address label-route-capability
```

The function of sending labeled routes is enabled.

Step 8 Run:

```
commit
```

The configuration is committed.

----End

9.17.4 Configuring Route Exchange Between a PE and a CE

An IPv6 routing protocol needs to be configured on a PE and a CE to enable them to learn IPv6 routes from each other.©

Context

The routing protocol running between a PE and a CE can be EBGP, IBGP, IPv6 static route, IS-IS, RIPng, or OSPFv3. For details, see the configuration of each routing protocol in the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Routing*.

9.17.5 Checking the Configuration

After BGP4+ 6PE is configured successfully, CEs can learn routes to each other.

Procedure

Step 1 Run the **display bgp ipv6 peer** command on each PE to check the status of the BGP4+ peer relationship.

Step 2 Run the **display mpls ldp session** [**vpn-instance** *vpn-instance-name*] [**verbose** | *peer-id*] command on each PE to check the status of the LDP session.

Step 3 Run the **display bgp ipv6 routing-table** *ipv6-address prefix-length* command on each PE or the **display ipv6 routing-table** *ipv6-address prefix-length* [**longer-match**] [**verbose**] command on each CE to check the routes destined for the remote IPv6 network.

----End

Example

- Run the **display bgp ipv6 peer** [**verbose**] command. You can see that BGP4+ peer relationships have been established.

```
<PE> display bgp ipv6 peer

 BGP local router ID : 11.11.11.11
 Local AS number : 100
 Total number of peers : 2         Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State
PrefRcv
  22.22.22.22     4         100      177      178     0 01:59:30
Established        1
  1::1            4       65410      158      163     0 01:30:00
Established        1
```

- Run the **display mpls ldp session** command. You can see that an LDP session has been established between PEs.

```
<PE> display mpls ldp session

 LDP Session(s) in Public Network
 Codes: LAM(Label Advertisement Mode), SsnAge Unit
(DDD:HH:MM)
 A '*' before a session means the session is being deleted.
--------------------------------------------------------------------------
 PeerID              Status      LAM  SsnRole  SsnAge       KASent/Rcv
--------------------------------------------------------------------------
 22.22.22.22:0       Operational DU   Passive  000:02:01   493/686
--------------------------------------------------------------------------
TOTAL: 1 Session(s) Found.
```

- Run the **display bgp ipv6 routing-table** *ipv6-address prefix-length* command on each PE or the **display ipv6 routing-table** command on each CE. You can view the routes destined for the remote IPv6 network.

```
<PE> display bgp ipv6 routing-table 6::6 128

 BGP local router ID : 11.11.11.11
 Local AS number : 100
 Paths : 1 available, 1 best, 1 select
 BGP routing table entry information of 6::6/128:
 Label information (Received/Applied): 2/NULL
 From: 22.22.22.22 (22.22.22.22)
 Route Duration: 0d01h49m43s
 Relay IP Nexthop: 10.1.1.2
 Relay IP Out-interface: Pos3/1/1
 Relay Tunnel Name: 0x0000000001004c4b42
 Original nexthop: ::FFFF:22.22.22.22
 AS-path 65420, origin igp, MED 0, localpref 100, pref-val 0, valid, internal,
b
```

```
est, select, pre 255
 Advertised to such 1 peers:
    1::1
<CE> display ipv6 routing-table 6::6 128
Routing Table : _public_
Summary Count : 1

 Destination  : 6::6                          PrefixLength : 128
 NextHop      : 1::2                            Preference   : 255
 Cost         : 0                               Protocol     : BGP
 RelayNextHop : 1::2                            TunnelID     : 0x0
 Interface    : Pos1/0/0                        Flags        : RD
```

9.18 Maintaining BGP4+

Maintaining BGP4+ involves resetting BGP4+ connections, clearing BGP4+ statistics, and debugging BGP4+.

9.18.1 Resetting BGP4+ Connections

Resetting BGP4+ connections interrupts peer relationships.

Context

CAUTION

BGP4+ peer relationships between routers are deleted if you reset BGP4+ connections by using the **reset bgp ipv6** command. Exercise caution when resetting BGP4+ connections.

When a BGP4+ routing policy (the router does not support the router-fresh capability) is changed, reset BGP connections to make the new configuration take effect. To reset BGP4+ connections, run any of the following commands in the user view as needed.

Procedure

- Run the **reset bgp ipv6 all** command to reset all BGP4+ connections.
- Run the **reset bgp ipv6** *as-number* command to reset a BGP4+ connection between two devices in different ASs.
- Run the **reset bgp ipv6** *ipv4-address* command to reset a BGP4+ connection between specified BGP4+ peers.
- Run the **reset bgp ipv6 external** command to reset all EBGP connections.
- Run the **reset bgp ipv6 internal** command to reset all IBGP connections.

----End

9.18.2 Clearing BGP4+ Statistics

Clearing BGP4+ statistics involves clearing route flapping statistics and route dampening statistics.

Context

CAUTION

BGP4+ statistics cannot be restored after they are cleared. Exercise caution when using reset commands.

Procedure

- Run the **reset bgp ipv6 flap-info** [**regexp** *as-path-regexp* | **as-path-filter** *as-path-filter-number* | *ipv6-address prefix-length*] command in the user view to clear route flapping statistics.
- Run the **reset bgp ipv6** *ipv6-address* **flap-info** command in the user view to clear route flapping statistics on a specified peer.
- Run the **reset bgp ipv6 dampening** [*ipv6-address prefix-length*] command in the user view to clear route dampening statistics and release suppressed routes.

----End

9.19 Configuration Examples

This chapter provides several BGP4+ configuration examples. In each configuration example, the networking requirements, network diagrams, precautions, configuration roadmap, and configuration procedures are provided.

9.19.1 Example for Configuring Basic BGP4+ Functions

Before building BGP4+ networks, you need to configure basic BGP4+ functions.

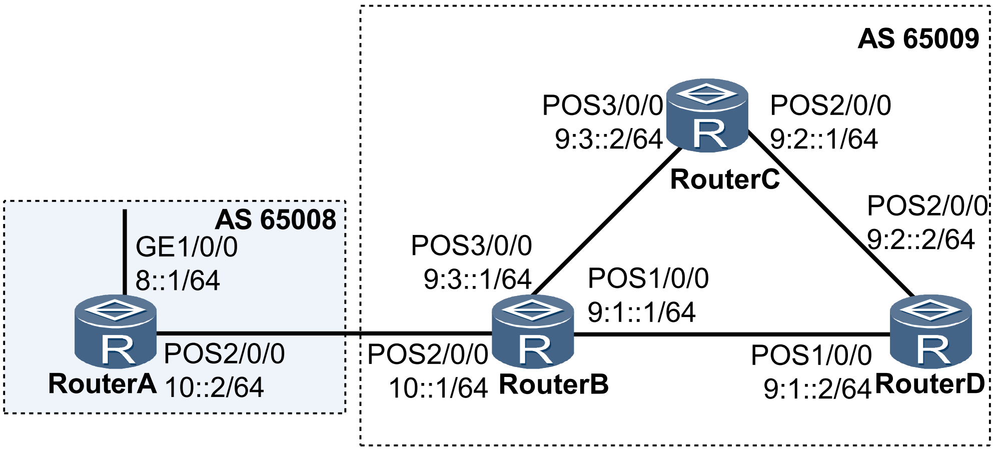
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 9-7**, there are two ASs: AS 65008 and AS 65009. Router A is in AS 65008, and Router B, Router C, and Router D are in AS 65009. BGP4+ must be configured to exchange routing information between the two ASs.

Figure 9-7 Networking diagram for configuring basic BGP4+ functions



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure IBGP connections between Router B, Router C, and Router D.
2. Configure an EBGP relationship between Router A and Router B.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs of Router A, Router B, Router C, and Router D
- AS numbers of Router A, Router B, Router C, and Router D

Procedure

Step 1 Configure an IPv6 address for each interface. The configuration details are not provided here.

Step 2 Configure IBGP connections.

# Configure Router B.

```
[~RouterB] ipv6
[~RouterB] bgp 65009
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 9:1::2 as-number 65009
[~RouterB-bgp] peer 9:3::2 as-number 65009
[~RouterB-bgp] ipv6-family unicast
[~RouterB-bgp-af-ipv6] peer 9:1::2 enable
[~RouterB-bgp-af-ipv6] peer 9:3::2 enable
[~RouterB-bgp-af-ipv6] network 9:1:: 64
[~RouterB-bgp-af-ipv6] network 9:3:: 64
[~RouterB-bgp-af-ipv6] commit
[~RouterB-bgp-af-ipv6] quit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] ipv6
[~RouterC] bgp 65009
```

```
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 9:3::1 as-number 65009
[~RouterC-bgp] peer 9:2::2 as-number 65009
[~RouterC-bgp] ipv6-family unicast
[~RouterC-bgp-af-ipv6] peer 9:3::1 enable
[~RouterC-bgp-af-ipv6] peer 9:2::2 enable
[~RouterC-bgp-af-ipv6] network 9:3:: 64
[~RouterC-bgp-af-ipv6] network 9:2:: 64
[~RouterC-bgp-af-ipv6] commit
[~RouterC-bgp-af-ipv6] quit
[~RouterC-bgp] quit
```

# Configure Router D.

```
[~RouterD] ipv6
[~RouterD] bgp 65009
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 9:1::1 as-number 65009
[~RouterD-bgp] peer 9:2::1 as-number 65009
[~RouterD-bgp] ipv6-family unicast
[~RouterD-bgp-af-ipv6] peer 9:1::1 enable
[~RouterD-bgp-af-ipv6] peer 9:2::1 enable
[~RouterD-bgp-af-ipv6] network 9:2:: 64
[~RouterD-bgp-af-ipv6] network 9:1:: 64
[~RouterD-bgp-af-ipv6] commit
[~RouterD-bgp-af-ipv6] quit
[~RouterD-bgp] quit
```

Step 3 Configure an EBGP connection.

# Configure Router A.

```
[~RouterA] ipv6
[~RouterA] bgp 65008
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 10::1 as-number 65009
[~RouterA-bgp] ipv6-family unicast
[~RouterA-bgp-af-ipv6] peer 10::1 enable
[~RouterA-bgp-af-ipv6] network 10:: 64
[~RouterA-bgp-af-ipv6] network 8:: 64
[~RouterA-bgp-af-ipv6] commit
[~RouterA-bgp-af-ipv6] quit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 65009
[~RouterB-bgp] peer 10::2 as-number 65008
[~RouterB-bgp] ipv6-family unicast
[~RouterB-bgp-af-ipv6] peer 10::2 enable
[~RouterB-bgp-af-ipv6] network 10:: 64
[~RouterB-bgp-af-ipv6] commit
[~RouterB-bgp-af-ipv6] quit
[~RouterB-bgp] quit
```

Step 4 Verify the configuration

# Check the status of BGP4+ connections.

```
[~RouterB] display bgp ipv6 peer
 BGP local router ID : 2.2.2.2
 Local AS number : 65009
 Total number of peers : 3                 Peers in established state : 3
  Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  9:1::2          4 65009        8        9     0 00:05:37 Established       2
  9:3::2          4 65009        2        2     0 00:00:09 Established       2
  10::2           4 65008        9        7     0 00:05:38 Established       2
```

The preceding command output shows that BGP4+ connections have been established between Router B and other routers.

Display the routing table of Router A.

```
[~RouterA] display bgp ipv6 routing-table
 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 6
 *>  Network  : 8::                                      PrefixLen : 64
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
 *>  Network  : 9:1::                                    PrefixLen : 64
     NextHop  : 10::1                                    LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 65009 i
 *>  Network  : 9:2::                                    PrefixLen : 64
     NextHop  : 10::1                                    LocPrf    :
     MED      :                                          PrefVal   : 0
     Label    :
     Path/Ogn : 65009 i
 *>  Network  : 9:3::                                    PrefixLen : 64
     NextHop  : 10::1                                    LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 65009 i
 *>  Network  : 10::                                     PrefixLen : 64
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
  *
     NextHop  : 10::1                                    LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 65009 i
```

The preceding command output shows that Router A has learned routes from its peer in AS 65009. AS 65008 and AS 65009 can exchange routing information.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 8::1/64
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10::2/64
#
bgp 65008
 router-id 1.1.1.1
 peer 10::1 as-number 65009
#
 ipv4-family unicast
```

```
  undo synchronization
#
 ipv6-family unicast
  network 8:: 64
  network 10:: 64
  peer 10::1 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:1::1/64
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 10::1/64
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:3::1/64
#
bgp 65009
 router-id 2.2.2.2
 peer 9:1::2 as-number 65009
 peer 9:3::2 as-number 65009
 peer 10::2 as-number 65008
#
 ipv4-family unicast
  undo synchronization
#
 ipv6-family unicast
  network 9:1:: 64
  network 9:3:: 64
  network 10:: 64
  peer 9:1::2 enable
  peer 9:3::2 enable
  peer 10::2 enable
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
ipv6
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:2::1/64
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:3::2/64
#
```

```
bgp 65009
 router-id 3.3.3.3
 peer 9:2::2 as-number 65009
 peer 9:3::1 as-number 65009
#
 ipv4-family unicast
  undo synchronization
#
 ipv6-family unicast
  network 9:2:: 64
  network 9:3:: 64
  peer 9:2::2 enable
  peer 9:3::1 enable
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:1::2/64
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 9:2::2/64
#
bgp 65009
 router-id 4.4.4.4
 peer 9:1::1 as-number 65009
 peer 9:2::1 as-number 65009
#
 ipv4-family unicast
  undo synchronization
#
 ipv6-family unicast
  network 9:1:: 64
  network 9:2:: 64
  peer 9:1::1 enable
  peer 9:2::1 enable
#
return
```

Related Tasks

9.3 Configuring Basic BGP4+ Functions

9.19.2 Example for Configuring BGP4+ Route Reflection

Configuring BGP4+ RRs simplifies the network configuration because IBGP peers do not need to be fully meshed.

Networking Requirements

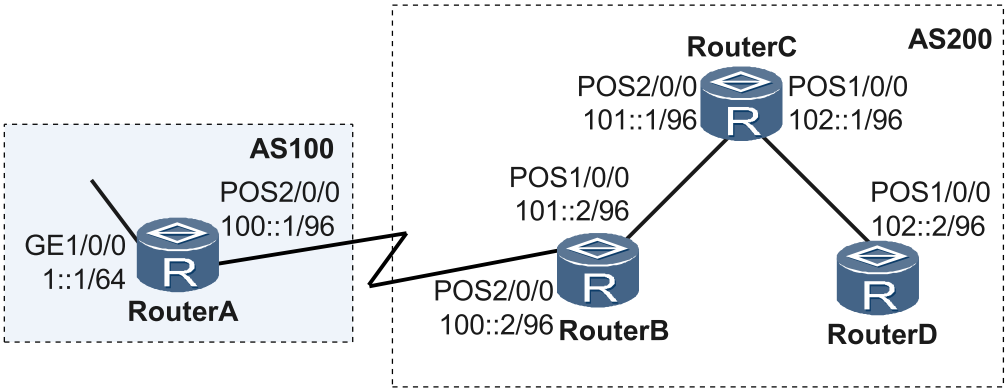
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 9-8**, Router B receives an Update packet from its EBGP peer and forwards it to Router C. Router C is configured as an RR, and has two clients: Router B and Router D.

Router B and Router D do not need to establish an IBGP connection. After receiving an Update packet from Router B, Router C reflects it to Router D. Similarly, after receiving an Update packet from Router D, Router C reflects it to Router B.

Figure 9-8 Networking diagram of configuring BGP4+ route reflection



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP4+ functions on each router.
2. Configure an RR and its clients to establish IBGP connections.
3. Configure Router C as an RR, and then check its routing information.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs of Router A, Router B, Router C, and Router D
- AS numbers of Router A, Router B, Router C, and Router D

Procedure

Step 1 Configure an IPv6 address for each interface. The configuration details are not provided here.

Step 2 Configure basic BGP4+ functions.

# Configure Router A.

```
[~RouterA] ipv6
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 100::2 as-number 200
[~RouterA-bgp] ipv6-family unicast
[~RouterA-bgp-af-ipv6] peer 100::2 enable
[~RouterA-bgp-af-ipv6] network 1:: 64
[~RouterA-bgp-af-ipv6] commit
[~RouterA-bgp-af-ipv6] quit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] ipv6
[~RouterB] bgp 200
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 100::1 as-number 100
[~RouterB-bgp] peer 101::1 as-number 200
[~RouterB-bgp] ipv6-family unicast
[~RouterB-bgp-af-ipv6] peer 100::1 enable
[~RouterB-bgp-af-ipv6] peer 101::1 enable
[~RouterB-bgp-af-ipv6] commit
[~RouterB-bgp-af-ipv6] quit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~RouterC] ipv6
[~RouterC] bgp 200
[~RouterC-bgp] router-id 3.3.3.3
[~RouterC-bgp] peer 101::2 as-number 200
[~RouterC-bgp] peer 102::2 as-number 200
[~RouterC-bgp] ipv6-family unicast
[~RouterC-bgp-af-ipv6] peer 101::2 enable
[~RouterC-bgp-af-ipv6] peer 102::2 enable
[~RouterC-bgp-af-ipv6] commit
[~RouterC-bgp-af-ipv6] quit
[~RouterC-bgp] quit
```

# Configure Router D.

```
[~RouterD] ipv6
[~RouterD] bgp 200
[~RouterD-bgp] router-id 4.4.4.4
[~RouterD-bgp] peer 102::1 as-number 200
[~RouterD-bgp] ipv6-family unicast
[~RouterD-bgp-af-ipv6] peer 102::1 enable
[~RouterD-bgp-af-ipv6] commit
[~RouterD-bgp-af-ipv6] quit
[~RouterD-bgp] quit
```

Step 3 Configure a BGP4+ RR.

# Configure Router C as an RR, with Router B and Router D as its clients.

```
[~RouterC-bgp] ipv6-family unicast
[~RouterC-bgp-af-ipv6] peer 101::2 reflect-client
[~RouterC-bgp-af-ipv6] peer 102::2 reflect-client
[~RouterC-bgp-af-ipv6] commit
```

Step 4 Verify the configuration

# Check the routing table of Router B.

```
[~RouterB] display bgp ipv6 routing-table
 BGP Local router ID is 2.2.2.2
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 6
 *>  Network  : 1::                                      PrefixLen : 64
     NextHop  : 100::1                                   LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 100 i
 *>  Network  : 100::                                    PrefixLen : 96
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
  *
     NextHop  : 100::1                                   LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 100 i
 *>  Network  : 101::                                    PrefixLen : 96
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
    i
     NextHop  : 101::1                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
 *>i Network  : 102::                                    PrefixLen : 96
     NextHop  : 101::1                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
```

# Check the routing table of Router D.

```
[~RouterD] display bgp ipv6 routing-table
 BGP Local router ID is 4.4.4.4
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 Total Number of Routes: 5
 *>i Network  : 1::                                      PrefixLen : 64
     NextHop  : 100::1                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : 100 i
 *>i Network  : 100::                                    PrefixLen : 96
     NextHop  : 101::2                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
 *>i Network  : 101::                                    PrefixLen : 96
     NextHop  : 102::1                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
 *>  Network  : 102::                                    PrefixLen : 96
     NextHop  : ::                                       LocPrf    :
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
    i
     NextHop  : 102::1                                   LocPrf    : 100
     MED      : 0                                        PrefVal   : 0
     Label    :
     Path/Ogn : i
```

The preceding command output shows that Router D and Router B have learned from Router C the routes advertised by Router A.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 1::1/64
#
interface Pos2/0/0
 link-protocol ppp
 ipv6 enable
 ipv6 address 100::1/96
#
bgp 100
 router-id 1.1.1.1
 peer 100::2 as-number 200
#
 ipv6-family unicast
  undo synchronization
  network 1:: 64
  network 100:: 96
  peer 100::2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 101::2/96
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 100::2/96
#
bgp 200
 router-id 2.2.2.2
 peer 100::1 as-number 100
 peer 101::1 as-number 200
#
 ipv6-family unicast
  undo synchronization
  network 100:: 96
  network 101:: 96
  peer 100::1 enable
  peer 101::1 enable
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 102::1/96
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 101::1/96
#
bgp 200
 router-id 3.3.3.3
 peer 101::2 as-number 200
 peer 102::2 as-number 200
#
 ipv6-family unicast
  undo synchronization
  network 101:: 96
  network 102:: 96
  peer 101::2 enable
  peer 101::2 reflect-client
  peer 102::2 enable
  peer 102::2 reflect-client
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
ipv6
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 102::2/96
#
bgp 200
 router-id 4.4.4.4
 peer 102::1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 102:: 96
  peer 102::1 enable
#
return
```

Related Tasks

9.12 Configuring BGP4+ RRs

9.19.3 Example for Configuring BFD for BGP4+

If the primary link between two BGP4+ peers fails, BFD can quickly detect the failure and report it to BGP4+. This allows service traffic to be quickly switched to the backup link.

Networking Requirements

CAUTION

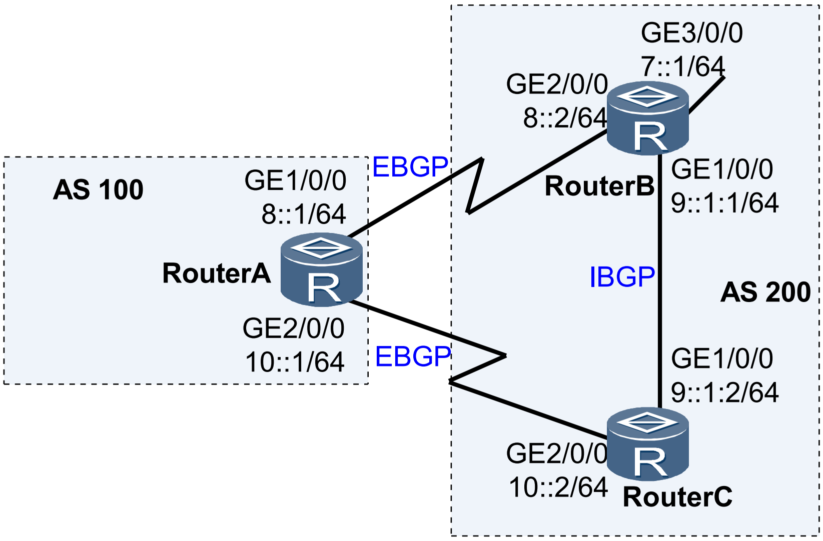
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 9-9**, Router A is in AS 100, and Router B and Router C are in AS 200. EBGP connections are established between Router A and Router B and between Router A and Router C.

Service traffic is transmitted over the primary link Router A -> Router B. The link Router A -> Router C -> Router B serves as the backup link.

BFD is used to detect the BGP4+ peer relationship between Router A and Router B. When the link between Router A and Router B fails, BFD can rapidly detect the failure and report it to BGP4+. This allows service traffic to be quickly switched to the backup link.

Figure 9-9 Networking diagram for configuring BFD for BGP4+



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP4+ functions on each router.

2. Configure the MED attribute on Router A and Router B to control route selection, allowing traffic to be transmitted over the primary link between Router A and Router B.
3. Enable BFD on Router A and Router B.

Data Preparation

To complete the configuration, you need the following data:

- Router IDs and AS numbers of Router A, Router B, and Router C
- IPv6 address of the remote end on a BFD session
- Minimum interval for sending BFD control packets, minimum interval for receiving BFD control packets, and local detection time multiplier

Procedure

Step 1 Configure an IPv6 address for each router. The configuration details are not provided here.

Step 2 Configure basic BGP4+ functions, establish EBGP connections between Router A and Router B and between Router A and Router C, and establish an IBGP connection between Router B and Router C.

# Configure Router A.

```
[~RouterA] bgp 100
[~RouterA-bgp] router-id 1.1.1.1
[~RouterA-bgp] peer 8::2 as-number 200
[~RouterA-bgp] peer 10::2 as-number 200
[~RouterA-bgp] ipv6-family unicast
[~RouterA-bgp-af-ipv6] peer 8::2 enable
[~RouterA-bgp-af-ipv6] peer 10::2 enable
[~RouterA-bgp-af-ipv6] commit
[~RouterA-bgp-af-ipv6] quit
[~RouterA-bgp] quit
```

# Configure Router B.

```
[~RouterB] bgp 200
[~RouterB-bgp] router-id 2.2.2.2
[~RouterB-bgp] peer 8::1 as-number 100
[~RouterB-bgp] peer 9::1:2 as-number 200
[~RouterB-bgp] ipv6-family unicast
[~RouterB-bgp-af-ipv6] peer 8::1 enable
[~RouterB-bgp-af-ipv6] peer 9::1:2 enable
[~RouterB-bgp-af-ipv6] network 7::1 64
[~RouterB-bgp-af-ipv6] commit
[~RouterB-bgp-af-ipv6] quit
[~RouterB-bgp] quit
```

# Configure Router C.

```
[~Routerc] bgp 200
[~Routerc-bgp] router-id 3.3.3.3
[~Routerc-bgp] peer 10::1 as-number 100
[~Routerc-bgp] peer 9::1:1 as-number 200
[~RouterC-bgp] ipv6-family unicast
[~RouterC-bgp-af-ipv6] peer 10::1 enable
[~RouterC-bgp-af-ipv6] peer 9::1:1 enable
[~RouterC-bgp-af-ipv6] commit
[~RouterC-bgp-af-ipv6] quit
[~RouterC-bgp] quit
```

# Display information about the BGP4+ peer relationship on Router A. You can see that the BGP4+ peer relationship has been established on the device.

```
<RouterA> display bgp ipv6 peer

 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Total number of peers : 2                 Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State
PrefRcv

  8::2            4         200       12       11     0 00:07:26 Established   0
  10::2           4         200       12       12     0 00:07:21 Established   0
```

Step 3 Configure the MED attribute.

Set the MEDs to be sent by Router B and Router C to Router A by using related policies.

# Configure Router B.

```
[~RouterB] route-policy 10 permit node 10
[~RouterB-route-policy] apply cost 100
[~RouterB-route-policy] quit
[~RouterB] bgp 200
[~RouterB-bgp] ipv6-family unicast
[~RouterB-bgp-af-ipv6] peer 8::1 route-policy 10 export
[~RouterB-bgp-af-ipv6] quit
[~RouterB-bgp] quit
[~RouterB] commit
```

# Configure Router C.

```
[~RouterC] route-policy 10 permit node 10
[~RouterC-route-policy] apply cost 150
[~RouterC-route-policy] quit
[~RouterC] bgp 200
[~RouterC-bgp] ipv6-family unicast
[~RouterC-bgp-af-ipv6] peer 10::1 route-policy 10 export
[~RouterC-bgp-af-ipv6] quit
[~RouterC-bgp] quit
[~RouterC] commit
```

# Check all BGP4+ routes on Router A.

```
<RouterA> display bgp ipv6 routing-table

 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 2
 *>  Network  : 7::                                      PrefixLen : 64
     NextHop  : 8::2                                     LocPrf    :
     MED      : 100                                      PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
 *
     NextHop  : 10::2                                    LocPrf    :
     MED      : 150                                      PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
```

The preceding command output shows that the next-hop address of the route to 7::1/64 is 8::2 and that traffic is transmitted on the primary link Router A→Router B.

Step 4 Configure BFD, and set the interval for sending BFD control packets, the interval for receiving BFD control packets, and the local detection multiplier.

# Enable BFD on Router A. Specify the minimum interval for sending BFD control packets to 100 ms, the minimum interval for receiving BFD control packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterA] bfd
[~RouterA-bfd] quit
[~RouterA] bgp 100
[~RouterA-bgp] peer 8::2 bfd enable
[~RouterA-bgp] peer 8::2 bfd min-tx-interval 100 min-rx-interval 100 detect-
multiplier 4
[~RouterA-bgp] quit
[~RouterA] commit
```

# Enable BFD on Router B. Specify the minimum interval for sending BFD control packets to 100 ms, the minimum interval for receiving BFD control packets to 100 ms, and the local detection multiplier to 4.

```
[~RouterB] bfd
[~RouterB-bfd] quit
[~RouterB] bgp 200
[~RouterB-bgp] peer 8::1 bfd enable
[~RouterB-bgp] peer 8::1 bfd min-tx-interval 100 min-rx-interval 100 detect-
multiplier 4
[~RouterB-bgp] comit
```

# Display all BFD sessions on Router A.

```
<RouterA> display bgp ipv6 bfd session all
--------------------------------------------------------------------------------
  Local_Address      Peer_Address       Interface
  8::1               8::2               GigabitEthernet1/0/0
  Tx-interval(ms)    Rx-interval(ms)    Multiplier  Session-State
  100                100                4           Up
--------------------------------------------------------------------------------
```

Step 5 Verify the configuration.

# Run the **shutdown** command on GE 2/0/0 of Router B to simulate a fault in the primary link.

```
[~RouterB] interface gigabitethernet 2/0/0
[~RouterB-Gigabitethernet2/0/0] shutdown
```

# Check the BGP4+ routing table on Router A.

```
<RouterA> display bgp ipv6 routing-table

 BGP Local router ID is 1.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 Total Number of Routes: 1
 *>  Network  : 7::                                     PrefixLen : 64
     NextHop  : 10::2                                  LocPrf    :
     MED      : 150                                     PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
```

The preceding command output shows that the next-hop address of the route to 7::1/64 becomes 10::2 and the backup link Router A → Router C → Router B takes effect after the primary link fails.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
ipv6
#
bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 8::1/64
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 10::1/64
#
interface NULL0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 router-id 1.1.1.1
 peer 8::2 as-number 200
 peer 8::2 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
 peer 8::2 bfd enable
 peer 10::2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  peer 8::2 enable
  peer 10::2 enable
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
ipv6
#
bfd
#
interface interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 8::2/64
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 9::1:1/64
#
interface GigabitEthernet3/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 7::1/64
#
interface NULL0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
```

```
#
bgp 200
 router-id 2.2.2.2
 peer 8::1 as-number 100
 peer 8::1 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
 peer 8::1 bfd enable
 peer 9::1:2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 7:: 64
  peer 8::1 enable
  peer 8::1 route-policy 10 export
  peer 9::1:2 enable
#
route-policy 10 permit node 10
 apply cost 100
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
ipv6
#
interface interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 9::1:2/64
#
interface interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 10::2/64
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 200
 router-id 3.3.3.3
 peer 9::1:1 as-number 200
 peer 10::1 as-number 100
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  peer 9::1:1 enable
  peer 10::1 enable
  peer 10::1 route-policy 10 export
#
route-policy 10 permit node 10
 apply cost 150
#
return
```

Related Tasks

9.13 Configuring BFD for BGP4+

9.19.4 Example for Configuring BGP4+ 6PE

BGP4+ 6PE enables separated IPv6 networks to communicate by using the MPLS tunneling technology.

Networking Requirements

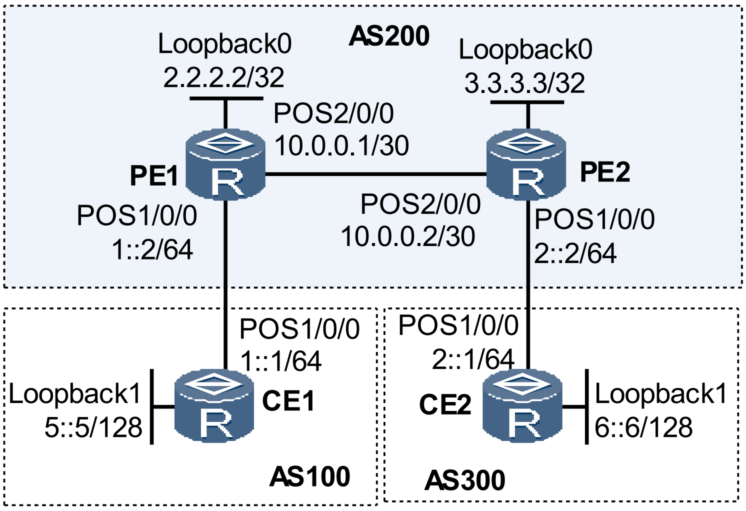
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

6PE enables IPv6 networks separated by IPv4/MPLS networks to communicate.

As shown in **Figure 9-10**, the IPv6 network where CE1 resides and the IPv6 network where CE2 resides are connected by an IPv4/MPLS network in AS 200. A 6PE peer relationship must be established between PE1 and PE2. This allows CE1 and CE2 to communicate across the IPv4/MPLS network. The 6PE peers send IPv6 routes learned from their attached CEs to each other by using MP-BGP, and forward IPv6 data over an LDP LSP.

Figure 9-10 Networking diagram for configuring BGP4+ 6PE



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on PE1 and PE2 to make them learn loopback interface addresses from each other.
2. Enable MPLS and MPLS LDP on the backbone network so that an LDP LSP can be established between PEs.
3. Establish a 6PE peer relationship between PE1 and PE2.

4. Configure BGP4+ on PEs and CEs to exchange IPv6 routes.

Data Preparation

To complete the configuration, you need the following data:

- Router ID of each device
- Number of the AS where each device resides

Procedure

Step 1 Configure IPv4 and IPv6 addresses for each interface. The configuration details are not provided here.

Step 2 Configure OSPF on PE1 and PE2 to make them learn loopback interface addresses from each other. For details, see configuration files in this configuration example.

Step 3 Enable MPLS and MPLS LDP on the backbone network so that an LDP LSP can be established between the PEs.

# Configure PE1.

```
[~PE1] mpls lsr-id 2.2.2.2
[~PE1] mpls
[~PE1-mpls] quit
[~PE1] mpls ldp
[~PE1-mpls-ldp] quit
[~PE1] interface pos2/0/0
[~PE1-Pos2/0/0] mpls
[~PE1-Pos2/0/0] mpls ldp
[~PE1-Pos2/0/0] quit
[~PE1] commit
```

# Configure PE2.

```
[~PE2] mpls lsr-id 3.3.3.3
[~PE2] mpls
[~PE2-mpls] quit
[~PE2] mpls ldp
[~PE2-mpls-ldp] quit
[~PE2] interface pos2/0/0
[~PE2-Pos2/0/0] mpls
[~PE2-Pos2/0/0] mpls ldp
[~PE2-Pos2/0/0] quit
[~PE2] commit
```

After completing the preceding configurations, run the **display mpls ldp session** command on each PE. You can see that an LDP session has been established between the PEs.

```
[~PE1] display mpls ldp session

 LDP Session(s) in Public Network
 Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDD:HH:MM)
 A '*' before a session means the session is being deleted.
--------------------------------------------------------------------------
 PeerID             Status      LAM  SsnRole  SsnAge       KASent/Rcv
--------------------------------------------------------------------------
 3.3.3.3:0         Operational DU   Passive  000:00:35    143/199
--------------------------------------------------------------------------
TOTAL: 1 Session(s) Found.
```

Step 4 Establish a 6PE peer relationship between the PEs.

# Configure PE1.

```
[~PE1] bgp 200
[~PE1-bgp] peer 3.3.3.3 as-number 200
[~PE1-bgp] peer 3.3.3.3 connect-interface LoopBack0
[~PE1-bgp] ipv6-family unicast
[~PE1-bgp-af-ipv6] peer 3.3.3.3 enable
[~PE1-bgp-af-ipv6] peer 3.3.3.3 label-route-capability
[~PE1-bgp-af-ipv6] quit
[~PE1-bgp] quit
[~PE1] commit
```

# Configure PE2.

```
[~PE2] bgp 200
[~PE2-bgp] peer 2.2.2.2 as-number 200
[~PE2-bgp] peer 2.2.2.2 connect-interface LoopBack0
[~PE2-bgp] ipv6-family unicast
[~PE2-bgp-af-ipv6] peer 2.2.2.2 enable
[~PE2-bgp-af-ipv6] peer 2.2.2.2 label-route-capability
[~PE2-bgp-af-ipv6] commit
[~PE2-bgp-af-ipv6] quit
[~PE2-bgp] quit
[~PE2] commit
```

After completing the preceding configurations, run the **display bgp ipv6 peer** command on each PE. You can see that a 6PE peer relationship has been established between the PEs.

```
[~PE1] display bgp ipv6 peer
 BGP local router ID : 2.2.2.2
 Local AS number : 200
 Total number of peers : 2                 Peers in established state : 2
  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  3.3.3.3         4         200     1248     1342      0 18:06:28 Established
1
```

Step 5 Configure BGP4+ on PEs and CEs to exchange IPv4 routes.

# Configure CE1.

```
[~CE1] bgp 100
[~CE1-bgp] peer 1::2 as-number 200
[~CE1-bgp] ipv6-family unicast
[~CE1-bgp-af-ipv6] peer 1::2 enable
[~CE1-bgp-af-ipv6] network 5::5 128
[~CE1-bgp-af-ipv6] commit
[~CE1-bgp-af-ipv6] quit
[~CE1-bgp] quit
```

# Configure PE1.

```
[~PE1] bgp 200
[~PE1-bgp] peer 1::1 as-number 100
[~PE1-bgp] ipv6-family unicast
[~PE1-bgp-af-ipv6] peer 1::1 enable
[~PE1-bgp-af-ipv6] commit
[~PE1-bgp-af-ipv6] quit
[~PE1-bgp] quit
```

# Configure PE2.

```
[~PE2] bgp 200
[~PE2-bgp] peer 2::1 as-number 300
[~PE2-bgp] ipv6-family unicast
[~PE2-bgp-af-ipv6] peer 2::1 enable
[~PE2-bgp-af-ipv6] commit
[~PE2-bgp-af-ipv6] quit
[~PE2-bgp] quit
```

# Configure CE2.

```
[~CE2] bgp 300
```

```
[~CE2-bgp] peer 2::2 as-number 200
[~CE2-bgp] ipv6-family unicast
[~CE2-bgp-af-ipv6] peer 2::2 enable
[~CE2-bgp-af-ipv6] network 6::6 128
[~CE2-bgp-af-ipv6] commit
[~CE2-bgp-af-ipv6] quit
[~CE2-bgp] quit
```

After completing the preceding configurations, run the **display bgp ipv6 peer** command on each PE or CE. You can see that the BGP4+ peer relationships have been established between the PEs and the CEs.

In the following example, the display on PE1 is used.

```
[~PE1] display bgp ipv6 peer

 BGP local router ID : 2.2.2.2
 Local AS number : 100
 Total number of peers : 2         Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State
PrefRcv
  3.3.3.3         4         200       59       60     0 00:35:46 Established
1
  1::1            4         100       40       45     0 00:06:16 Established
1
```

Step 6 Checking the Configuration

After the preceding configurations are complete, CEs can learn the routes to each other's loopback interface, and ping each other.

In the following example, the display on CE1 is used.

```
[~CE1] display ipv6 routing-table
Routing Table : _public_
         Destinations : 8        Routes : 8

 Destination  : ::1                             PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : ::FFFF:127.0.0.0                PrefixLength : 104
 NextHop      : ::FFFF:127.0.0.0                Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : ::FFFF:127.0.0.0                PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : InLoopBack0                     Flags        : D

 Destination  : 1::                             PrefixLength : 64
 NextHop      : 1::1                            Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : Pos1/0/0                        Flags        : D

 Destination  : 1::1                            PrefixLength : 128
 NextHop      : ::1                             Preference   : 0
 Cost         : 0                               Protocol     : Direct
 RelayNextHop : ::                              TunnelID     : 0x0
 Interface    : Pos1/0/0                 Flags         : D

 Destination  : 5::5                            PrefixLength : 128
 NextHop      : ::1                              Preference   : 0
 Cost         : 0                                Protocol     : Direct
 RelayNextHop : ::                               TunnelID     : 0x0
 Interface    : LoopBack2                        Flags        : D

 Destination  : 6::6                             PrefixLength : 128
 NextHop      : 1::2                             Preference   : 255
 Cost         : 0                                Protocol     : BGP
 RelayNextHop : 1::2                             TunnelID     : 0x0
 Interface    : Pos1/0/0                  Flags        : RD

 Destination  : FE80::                           PrefixLength : 10
 NextHop      : ::                               Preference   : 0
 Cost         : 0                                Protocol     : Direct
 RelayNextHop : ::                               TunnelID     : 0x0
 Interface    : NULL0                            Flags        : D
<CE1> ping ipv6 -a 5::5 6::6

  PING 6::6 : 56  data bytes, press CTRL_C to break
    Reply from 6::6
    bytes=56 Sequence=1 hop limit=62 time=8 ms
    Reply from 6::6
    bytes=56 Sequence=2 hop limit=62 time=2 ms
    Reply from 6::6
    bytes=56 Sequence=3 hop limit=62 time=4 ms
    Reply from 6::6
    bytes=56 Sequence=4 hop limit=62 time=3 ms
    Reply from 6::6
    bytes=56 Sequence=5 hop limit=62 time=4 ms
  ---6::6 ping statistics---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=2/4/8 ms
```

After 6PE is configured, the IPv6 network where CE1 resides and the IPv6 network where CE2 resides can communicate by means of the IPv4/MPLS network.

----End

Configuration Files

- Configuration file of CE1

```
#
sysname CE1
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 1::1/64
#
interface LoopBack1
 ipv6 enable
 ipv6 address 5::5/128
#
bgp 100
 peer 1::2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 5:: 64
  peer 1::2 enable
#
```

```
return
```

- Configuration file of PE1

```
#
sysname PE1
#
mpls lsr-id 2.2.2.2
#
mpls
#
mpls ldp
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 1::2/64
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.0.0.1 255.255.255.252
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 200
 peer 3.3.3.3 as-number 200
 peer 3.3.3.3 connect-interface LoopBack0
 peer 1::1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  peer 3.3.3.3 enable
 #
 ipv6-family unicast
  undo synchronization
  peer 3.3.3.3 enable
  peer 3.3.3.3 label-route-capability
  peer 1::1 enable
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 10.0.0.0 0.0.0.3
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
mpls lsr-id 3.3.3.3
#
mpls
#
mpls ldp
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 10.0.0.2 255.255.255.252
 mpls
 mpls ldp
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 2::2/64
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 200
 peer 2.2.2.2 as-number 200
 peer 2.2.2.2 connect-interface LoopBack0
 peer 2::1 as-number 300
#
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.2 enable
 #
 ipv6-family unicast
  undo synchronization
  peer 2.2.2.2 enable
  peer 2.2.2.2 label-route-capability
  peer 2::1 enable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 10.0.0.0 0.0.0.3
#
return
```

- Configuration file of CE2

```
#
sysname CE2
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ipv6 enable
 ipv6 address 2::1/64
#
interface LoopBack1
 ipv6 enable
 ipv6 address 6::6/128
#
bgp 300
 peer 2::2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 6:: 64
  peer 2::2 enable
#
return
```

Related Tasks

9.17 Configuring BGP4+ 6PE

10 Routing Policy Configuration

About This Chapter

Routing policies are applied to routing information to change the path through which network traffic passes.

Routing policy configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

10.1 Routing Policy Overview

By using routing policies, you can strictly control the receiving and advertising of routes on networks.

Routing Policy

Routing policies are applied to routing information to change the path through which network traffic passes. Routing policies can change the path through which network traffic passes by applying route attributes (including reachability).

When advertising or receiving routes, a router uses certain routing policies to filter routes by applying route attributes. Currently, the main method of implementing routing policies is to filter routing information. For example:

- Only the routes that meet the matching rules are received or advertised.
- To enrich routing information, a routing protocol such as the Routing Information Protocol Next Generation (RIPng) may need to import the routes discovered by other routing protocols. When importing routes from other routing protocols, a router may import only the routes that meet the matching rules, and set some attributes of the imported routes to meet the requirements of the protocol.

To implement routing policies, you need to define the characteristics of routes to which routing policies are applied. That is, you need to define a set of matching rules and actions to be performed. Then, you can apply the defined matching rules to the routing policies for advertising, receiving, or importing routes.

Differences Between the Routing Policy and Policy-based Routing

Unlike the routing mechanism that searches the forwarding table for matching routes based on the destination addresses of IP packets, policy-based routing (PBR) is based on the user-defined routing policies. PBR selects routes according to the user-defined routing policies, with reference to the source IP addresses and lengths of incoming packets. PBR can be used for security and load balancing.

Routing policies and PBR are different mechanisms. **Table 10-1** shows the differences between the two mechanisms.

Table 10-1 Differences between the routing policy and PBR

Routing Policy	Policy-based Routing
Forwards packets based on destination addresses in the routing table.	Forwards packets based on the policy. If packets fail to be forwarded based on the policy, the device begins to search the routing table for packet forwarding.
Based on the control plane and serves routing protocols and routing tables.	Based on the forwarding plane and serves forwarding policies.
Combines with routing protocols to form policies.	Needs to be manually configured hop by hop to ensure that packets are forwarded according to the policies.

Routing Policy	Policy-based Routing
Uses the **route-policy** command.	Uses the **policy-based-route** command.

10.2 Routing Policy Features Supported by the NE5000E

When configuring routing policies, you can use these filters: Access Control Lists (ACLs), IP prefix lists, AS_Path filters, community filters, extended community filters, RD filters, and route-policies.

Filter

The NE5000E provides seven types of filters for routing protocols: ACLs, IP prefix lists, AS_Path filters, community filters, extended community filters, RD filters, and route-policies. Details about each type of filter are as follows:

- ACL

  There are ACLs for IPv4 packets and for IPv6 packets. According to the applications, ACLs are classified into three types, that is, interface-based ACLs, basic ACLs, and advanced ACLs. When defining an ACL, you can specify the IP address and subnet range to match the destination network segment address or next-hop address of a route.

  For the configuration of an ACL, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - IP Services*.

- IP prefix list

  There are IPv4 prefix lists and IPv6 prefix lists, which are flexible in filtering routes.

  An IP prefix list is identified by its prefix list name. Each prefix list contains multiple entries. Each entry can specify a matching range in the form of a network prefix. The matching range is identified by an index number that designates the matching sequence.

  During route matching, a router checks the entries identified by index numbers in an ascending order. If a route matches an entry, the route does not continue to match the next entry. For the configuration of an IP prefix list, refer to **Configuring an IP Prefix List**.

- AS_Path filter

  Each BGP route contains an AS_Path attribute. AS_Path filters specify matching rules regarding AS_Path attributes.

  For the configuration of an AS_Path filter, refer to "BGP Configuration."

- Community filter

  Community filters are exclusive to BGP. BGP route contains a community attribute field that identifies a community. Community filters specify matching rules regarding community attributes.

  For the configuration of a community filter, refer to "BGP Configuration."

- Extended community filter

  Extended community filters are exclusive to BGP. The extended community of BGP supports only the route-target (RT) extended community of VPN. Extended community filters specify matching conditions regarding extended community attributes.

  For the configuration of an extended community filter, refer to "BGP Configuration."

- RD filter

  Through a route distinguisher (RD), a VPN instance has an independent address space and distinguishes the IPv4 and IPv6 prefixes in the same address space. RD filters specify matching rules regarding RD attributes.

  For the configuration of an RD filter, refer to the *HUAWEI NetEngine5000E Core Router Configuration Guide - VPN*.

- Route-policy

  A route-policy is a complex filter. With a route-policy, you can obtain the required routes by matching route attributes, and change route attributes when the matching rules are met. A route-policy can use the preceding filters to define the matching rules.

  A route-policy consists of multiple nodes and the relationship between these nodes is OR. The system checks the nodes according to index numbers. If a route matches a node in the route-policy, the route does not continue to match the next node.

  Each node comprises a set of **if-match** and **apply** clauses. The **if-match** clauses define the matching rules that are used to match certain route attributes. The relationship between the **if-match** clauses of a node is AND. A route matches a node only when the route matches all the matching rules defined by the **if-match** clauses of the node. The **apply** clauses specify actions. When a route matches a node, the **apply** clauses set certain attributes for the route. For the configuration of a route-policy, refer to **Configuring a Route-Policy**.

Applications of the Routing Policy

Routing policies are mainly used in the following situations:

- When importing the routes discovered by other routing protocols, a routing protocol imports only the routes that meet the matching rules by applying filters.
- When advertising or receiving the routes discovered by itself, a routing protocol advertises or receives only the routes that meet the matching rules by applying filters.

For the applications of routing policies in routing protocols, refer to the corresponding routing protocol configurations.

NOTE

If a routing policy changes, by default, routing protocols apply a new routing policy immediately.

10.3 Configuring an IP Prefix List

An IP prefix list filters routes according to the destination addresses of the routes.

Applicable Environment

To control the receiving and advertising of routes according to the destination addresses of the routes, you need to configure an IP prefix list.

Each route contains an IP prefix. Therefore, IP prefix lists can be flexibly applied in various networks to filter routes.

Before applying a routing policy, you need to set the matching rules, that is, filters. Similar to an ACL, an IP prefix list is flexible in filtering routes. IP prefix lists filter routes according to the destination addresses of the routes.

Pre-configuration Tasks

None.

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

10.3.1 Configuring an IPv4 Prefix List

An IP prefix list filters routes according to IP address prefixes. An IP address prefix is defined by an IP address and the mask length.

Context

⚠ **CAUTION**

When modifying the configurations of multiple cooperative IP prefix lists, you are recommended to perform the configuration task of **Setting the Delay for Applying a Routing Policy**. Otherwise, an incomplete routing policy will cause route flapping.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

ip ip-prefix *ip-prefix-name* [**index** *index-number*] { **permit** | **deny** } *ip-address mask-length* [**greater-equal** *greater-equal-value*] [**less-equal** *less-equal-value*]

An IPv4 prefix list is configured.

The range of the mask length can be specified as *mask-length* <= *greater-equal-value* <= *less-equal-value* <= 32. If only **greater-equal** is specified, the range of the IP prefix is [*greater-equal-value*, 32]. If only **less-equal** is specified, the range of the IP prefix is [*mask-length*, *less-equal-value*].

An IPv4 prefix list is identified by its list name, and each IP prefix list contains multiple entries. Each entry can specify a matching range in the form of a network prefix. The matching range is identified by an index number. For example, the following is an IPv4 prefix list named **abcd**:

```
#
ip ip-prefix abcd index 10 permit 1.0.0.0 8
ip ip-prefix abcd index 20 permit 2.0.0.0 8
```

During route matching, the system checks the entries identified by index numbers in an ascending order. If a route matches an entry, the route does not continue to match the next entry.

In the NE5000E, all unmatched routes are denied by the IPv4 prefix list by default. If all entries are set to be in **deny** mode, all routes are denied by the IP prefix list. Therefore, you need to define an entry **permit 0.0.0.0 0 less-equal 32** following the entries in **deny** mode to allow all the other IPv4 routes to pass the filtering of the IP prefix list.

NOTE

If more than one IP prefix entry is defined, at least one entry needs to be set in **permit** mode.

Step 3 Run:

commit

The configuration is committed.

----End

10.3.2 Configuring an IPv6 Prefix List

An IPv6 prefix list filters routes according to IPv6 address prefixes. An IPv6 address prefix is defined by an IPv6 address and the mask length.

Context

Do as follows on the router where an IPv6 prefix list needs to be applied:

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

ip ipv6-prefix *ipv6-prefix-name* [**index** *index-number*] { **permit** | **deny** } *ipv6-address prefix-length* [**greater-equal** *greater-equal-value*] [**less-equal** *less-equal-value*]

An IPv6 prefix list is configured.

An IPv6 prefix list is identified by its list name, and each IPv6 prefix list contains multiple entries. Each entry can specify a matching range in the form of a network prefix. The matching range is identified by an index number. For example, the following is an IPv6 prefix list named **abcd**:

```
#
ip ipv6-prefix abcd index 10 permit 1:: 64
ip ipv6-prefix abcd index 20 permit 2:: 64
```

During route matching, the system checks the entries identified by index numbers in an ascending order. If a route matches an entry, the route does not continue to match the next entry.

On the NE5000E, all the unmatched routes are denied by the IPv6 prefix list by default. If all entries are set to be in **deny** mode, all routes are denied by the IPv6 prefix list. Therefore, you need to define an entry **permit :: 0 less-equal 128** following the entries in **deny** mode to allow all the other IPv6 routes to pass the filtering of the IPv6 prefix list.

NOTE

If more than one IPv6 prefix entry is defined, at least one entry needs to be set in **permit** mode.

Step 3 Run:

commit

The configuration is committed.

----End

10.3.3 Checking the Configuration

After an IP prefix list is configured, you can check information about the IP prefix list.

Prerequisite

All configurations of the IP prefix list are complete.

Procedure

- Run the **display ip ip-prefix** [*ip-prefix-name*] command to check information about the IPv4 prefix list.
- Run the **display ip ipv6-prefix** [*ipv6-prefix-name*] command to check information about the IPv6 prefix list.

----End

Example

Run the **display ip ip-prefix p1** command, and you can view information about the IP prefix list named **p1**.

```
<HUAWEI> display ip ip-prefix p1
Prefix-list pl
Permitted 5
Denied 2
index: 10      permit  192.168.0.0/16          ge  17  le  18
```

10.4 Configuring a Route-Policy

Each node of a route-policy can comprise a set of **if-match** and **apply** clauses.

Applicable Environment

A route-policy includes various matching rules and hence can flexibly meet the requirements of various scenarios. Except ACLs, IP prefix lists, and AS_Path filters, other filters need to be used together with a route-policy.

A route-policy is used to match routes or attributes of routes, and to change the attributes when the matching rules are met. The matching rules of a route-policy can use other filters: ACLs, IP prefix lists, AS_Path filters, community filters, extended community filters, and RD filters.

A route-policy can consist of multiple nodes, and each node can comprise the following clauses:

- **if-match** clauses: define the matching rules that are used to match certain route attributes, that is, the conditions that routes need to meet to pass the filtering of the current route-policy.
- **apply** clauses: specify actions, namely, running configuration commands used for modifying some attributes of routes.

For more information about a route-policy, refer to the *HUAWEI NetEngine5000E Core Router Feature Description - IP Routing*.

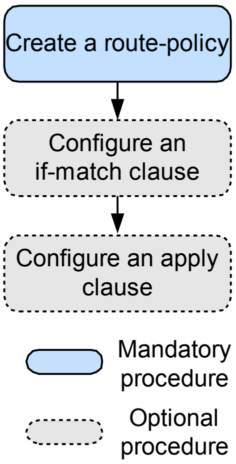
Pre-configuration Tasks

Before configuring a route-policy, complete the following tasks:

- **Configuring an IP Prefix List**
- Configuring a routing protocol

Configuration Procedures

Figure 10-1 Configuration flowchart of a route-policy



Related Tasks

10.9.3 Example for Applying the Routing Policy

10.4.1 Creating a Route-Policy

By applying a route-policy, you can set attributes for the imported routes as required.

Context

CAUTION

When modifying the configurations of multiple cooperative route-policies, you are recommended to perform the configuration task of **Setting the Delay for Applying a Routing Policy**. Otherwise, an incomplete routing policy will cause route flapping.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
route-policy route-policy-name { permit | deny } node node
```

A route-policy is created, and the route-policy view is displayed.

The matching modes of a node include **permit** and **deny**:

- The parameter **permit** specifies the matching mode for a node in a route-policy as permit. In **permit** mode, if a route matches a node, the actions specified by **apply** clauses are performed on the route, and the route does not continue to match the next node. If the route fails to match the node, the route continues to match the next node.
- The parameter **deny** specifies the matching mode for a node in a route-policy as deny. In **deny** mode, the actions specified by **deny** clauses are not performed. In **deny** mode, if a route matches all the **if-match** clauses of a node, the route is denied by the node and does not continue to match the next node. If the route does not match any **if-match** clause of the node, the route continues to match the next node.

NOTE

On the NE5000E, by default, all the unmatched routes are denied by the route-policy. If more than one node is defined in a route-policy, at least one node needs to be set in **permit** mode.

When a route-policy is used to filter routes, note the following: If a route does not match any node in the route-policy, it indicates that the route is denied by the route-policy. If all the nodes in a route-policy are set in **deny** mode, all the routes to be filtered are denied by the route-policy.

When a route-policy is used to filter routes, the node with a smaller value is matched first.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

10.4.2 (Optional) Configuring an if-match Clause

The **if-match** clauses define the matching rules that are used to match certain route attributes.

Context

CAUTION

When modifying a route-policy that contains multiple cooperative **if-match** clauses, you are recommended to perform the configuration task of **Setting the Delay for Applying a Routing Policy**. Otherwise, an incomplete route-policy will cause route flapping.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
route-policy route-policy-name { permit | deny } node node
```

The route-policy view is displayed.

Step 3 Run the following command as required to configure **if-match** clauses in the route-policy:

- Run:

  ```
  if-match acl { acl-number | acl-name }
  ```

  The ACL is configured to match routes.

- Run:

  ```
  if-match cost cost
  ```

  The route cost is set to match routes.

- Run:

  ```
  if-match ip { next-hop | route-source } { acl { acl-number | acl-name } | ip-
  prefix ip-prefix-name }
  ```

  The next hop or source address is configured to match IPv4 routes.

- Run:

  ```
  if-match ip-prefix ip-prefix-name
  ```

  The IP prefix list is configured to match routes.

  NOTE

  For the same route-policy node, the **if-match acl** command and the **if-match ip-prefix** command cannot be configured at the same time. This is because the latest configuration overwrites the previous configuration.

- Run:

  ```
  if-match ipv6 { address | next-hop | route-source } prefix-list ipv6-prefix-name
  ```

  The IPv6 prefix list is configured to match IPv6 routes.

- Run the following command as required to match the type of route:

  - Run:

    ```
    if-match route-type { external-type1 | external-type1or2 | external-type2 |
    internal | nssa-external-type1 | nssa-external-type1or2 | nssa-external-
    type2 }
    ```

    OSPF is set to match routes.

  - Run:

    ```
    if-match route-type { is-is-level-1 | is-is-level-2 }
    ```

    IS-IS is set to match routes.

- Run:

  ```
  if-match tag tag
  ```

  The route tag is set to match routes.

The commands in Step 3 are not listed in sequence. A node can have multiple or no **if-match** clauses.

NOTE

For the same node in a route-policy, the relationship between **if-match** clauses is AND. A route needs to meet all the matching rules before the actions defined by **apply** clauses are performed. The relationship between the **if-match** clauses in the **if-match route-type** command is OR, but the relationship between the **if-match** clauses in the **if-match route-type** command and other commands is AND.

If no **if-match** clause is specified, all routes are matched.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.4.3 (Optional) Configuring an apply Clause

The **apply** clauses specify actions to set certain route attributes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
route-policy route-policy-name { permit | deny } node node
```

The route-policy view is displayed.

Step 3 Run the following command as required to configure **apply** clauses in the route-policy:

- Run:

  ```
  apply cost { [ + | - ] cost | inherit }
  ```

  The route cost is set.

- Run the following command as required to set the cost type of a route:
  - Run:

    ```
    apply cost-type { external | internal }
    ```

    The IS-IS cost type is set.
  - Run:

    ```
    apply cost-type { type-1 | type-2 }
    ```

    The OSPF cost type is set.

- Run:

  ```
  apply dampening half-life-reach reuse suppress ceiling
  ```

  The dampening parameters of EBGP routes is set.

- Run:

  ```
  apply ip-address next-hop ipv4-address
  ```

  The next-hop address of the IPv4 route is set.

- Run:

  ```
  apply ipv6 next-hop ipv6-address
  ```

  The next-hop address of the IPv6 route is set.

- Run:

  ```
  apply isis { level-1 | level-1-2 | level-2 }
  ```

  The level of the IS-IS route is set.

- Run:

  ```
  apply preference preference
  ```

  The preference of the routing protocol is set.

- Run:

  ```
  apply tag tag
  ```

The route tag is set.

The commands in Step 3 are not listed in sequence. A node can have multiple or no **apply** clauses.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.4.4 Checking the Configuration

After a route-policy is configured, you can check information about the route-policy.

Prerequisite

All configurations of the route-policy are complete.

Procedure

- Run the **display route-policy** [*route-policy-name*] command to check information about the route-policy.

----End

Example

Run the **display route-policy policy1** command, and you can view information about the route-policy named **policy1**.

```
<HUAWEI> display route-policy policy1
Route-policy : policy1
  permit : 10
    Match clauses :
        if-match acl 2000
    Apply clauses :
        apply cost 100
        apply tag 100
```

10.5 Applying Filters to the Received Routes

By applying the related filters of routing policies to routing protocols, you can filter the received routes.

Applicable Environment

When exchanging routes on a network, devices need to selectively receive routes as required. After defining the related filters (including the IP prefix list, ACL, and route-policy) of a routing policy, you need to apply these filters in routing protocols. You can run the **filter-policy** command in the related protocol view and apply an ACL or an IP prefix list to filter the received routes. Only the routes that meet the matching rules are received.

You can run the **filter-policy import** command to filter the received routes. For a distance-vector protocol and a link-state protocol, the operations of the **filter-policy** command are different:

- Distance-vector protocol

A distance-vector protocol generates routes based on the routing table. Therefore, filters affect the routes received from neighbors and the routes advertised to neighbors.

- Link-state protocol

  A link-state protocol generates routes based on the link state database (LSDB). The **filter-policy** command does not affect any Link State Advertisement (LSA) or the integrity of any LSDB. Therefore, the inbound policy and outbound policy have different impact on a link-state protocol.

  When routes are being received, the **filter-policy** command just determines which routes to be installed from the protocol routing table into the local core routing table. That is, this command affects the local core routing table rather than the protocol routing table.

NOTE

- BGP has the powerful filtering function. For the configuration of BGP routing policies, refer to "BGP Configuration."
- For details of the **filter-policy** and **import-route** commands and their applications in RIP, OSPF, IS-IS, and BGP, refer to related configurations.

Pre-configuration Tasks

Before applying filters to the received routes, complete the following tasks:

- **Configuring an IP Prefix List**
- Configuring an ACL
- **Configuring a Route-Policy**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

Related Tasks

10.9.1 Example for Filtering the Routes to Be Received or Advertised

10.5.1 Configuring RIP to Filter the Received Routes

You can configure an inbound or outbound filtering policy by specifying Access Control Lists (ACLs) and IP address prefix lists to filter routes to be received and advertised. You can also configure a device to receive only the RIP packets from a specified neighbor.

Context

Devices can filter the routing information. To filter the received and advertised routes, you can configure inbound and outbound filtering policies by specifying the ACL and IP prefix list.

You can also configure a device to receive RIP packets from only a specified neighbor.

For details on how to configure RIP to filter the advertised routes, see **5.6.4 Configuring RIP to Filter the Routes to Be Advertised**.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Configure RIP to filter the received routes as required:

- Run:

  ```
  filter-policy acl-number import [ interface-type interface-number ]
  ```

  The learned routes are filtered based on the ACL.

- Run:

  ```
  filter-policy gateway ip-prefix-name import
  ```

  The routes advertised by neighbors are filtered based on the destination address prefix.

- Run:

  ```
  filter-policy acl-name acl-name import [ interface-type interface-number ]
  ```

  The routes learned by the specified interface are filtered based on the ACL name.

- Run:

  ```
  filter-policy ip-prefix ip-prefix-name [ gateway ip-prefix-name ] import
  [ interface-type interface-number ]
  ```

  The routes learned by the specified interface are filtered based on the destination address prefix and neighbors.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

10.5.2 Configuring OSPF to Filter the Received Routes

After a filtering policy is configured for the OSPF routes that need to be delivered to the routing management module, only the routes that match the policy will be added to the routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } import
```

OSPF is configured to filter the received routes.

- The parameter *acl-number* specifies the number of a basic ACL.

- The parameter **acl-name** *acl-name* specifies the name of an ACL.
- The parameter **ip-prefix** *ip-prefix-name* specifies the name of an IP prefix list.

OSPF is a link-state dynamic routing protocol, with routing information carried in the LSA. Therefore, the **filter-policy import** command cannot be used to filter the advertised or received LSAs.

The **filter-policy import** command is used to filter the routes calculated by OSPF. Only the routes that pass the filtering are added to the routing table. Routes that do not pass the filtering can not added to the OSPF routing table, but can be advertised. Therefore, the LSDB is not affected regardless of whether the received routes pass the filtering.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.5.3 Configuring IS-IS to Filter the Received Routes

By configuring IS-IS to filter the received routes, you can control the number of IS-IS routes to be added to the IP routing table, and thus reduce the size of the IP routing table.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name | route-policy route-policy-name } import
```

IS-IS is configured to filter the received routes that need to be added to the IP routing table.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.5.4 Checking the Configuration

After filters are applied to the received routes, you can check information about the routing table of each protocol.

Prerequisite

Applying filters to the received routes is complete.

Procedure

- Run the **display rip** *process-id* **route** command to check information about the RIP routing table.
- Run the **display ospf** [*process-id*] **routing** command to check information about the OSPF routing table.
- Run the **display isis** [*process-id*] **route** command to check information about the IS-IS routing table.
- Run the **display ip routing-table** command to check information about the IP routing table.

  Run the **display ip routing-table** command on the local router, and you can view that the routes that meet the matching rules set on the neighbor are filtered or the actions defined by **apply** clauses are performed on these routes.

----End

10.6 Applying Filters to the Advertised Routes

By applying the related filters of routing policies to routing protocols, you can filter the advertised routes.

Applicable Environment

After defining the related filters (including the IP prefix list, ACL, and route-policy) of a routing policy, you need to apply these filters to routing protocols. You can use the **filter-policy** command in the related protocol view and apply an ACL or an IP prefix list to filter the advertised routes. Only the routes that meet the matching rules are advertised.

You can run the **filter-policy export** command to filter the advertised routes. For a distance-vector protocol and a link-state protocol, the operations of the **filter-policy** command are different:

- Distance-vector protocol

  A distance-vector protocol generates routes based on the routing table. Therefore, filters affect the routes received from neighbors and the routes advertised to neighbors.
- Link-state protocol

  A link-state protocol generates routes based on the LSDB. The **filter-policy** command does not affect any Link State Advertisement (LSA) or the integrity of any LSDB. Therefore, the inbound policy and outbound policy have different impact on a link-state protocol.

  When advertising routes, you can run the **filter-policy export** command to determine whether to advertise the imported routes (such as the imported RIP routes). Only the LSAs or Link State PDUs (LSPs) that are imported through the **filter-policy import** command are added to the LSDB. This does not affect the LSAs advertised to other routers.

NOTE

- BGP has the powerful filtering function. For the configuration of BGP routing policies, refer to "BGP Configuration."
- For details of the **filter-policy** and **import-route** commands and their applications in RIP, OSPF, IS-IS, and BGP, refer to related configurations.

Pre-configuration Tasks

Before applying filters to the advertised routes, complete the following tasks:

- **Configuring an IP Prefix List**
- Configuring an ACL
- **Configuring a Route-Policy**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

Related Tasks

10.9.1 Example for Filtering the Routes to Be Received or Advertised

10.6.1 Configuring RIP to Filter the Routes to Be Advertised

You can set conditions to filter the routes to be advertised. Only the routes that meet the conditions can be advertised.

Context

Devices can filter the routing information. To filter the advertised routes, you can configure inbound and outbound filtering policies by specifying the ACL and IP prefix list.

For details on how to configure RIP to filter the received routes, see **5.6.3 Configuring RIP to Filter the Received Routes**.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } export
[ protocol process-id | interface-type interface-number ]
```

The advertised routes are filtered based on the ACL and the destination address prefix.

Step 4 Run:

```
commit
```

The configuration is submitted.

----End

10.6.2 Configuring OSPF to Filter the Routes to Be Advertised

After a filtering policy is configured for OSPF routes to be imported, only the routes that match the policy will be advertised.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ospf [ process-id ]
```

The OSPF process view is displayed.

Step 3 (Optional) Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } export
[ protocol [ process-id ] ]
```

OSPF is configured to filter the routes imported through the **import-route** command. Only the routes that pass the filtering are advertised.

- The parameter *acl-number* specifies the number of a basic ACL.
- The parameter **acl-name** *acl-name* specifies the name of an ACL.
- The parameter **ip-prefix** *ip-prefix-name* specifies the name of an IP prefix list.

You can specify the parameter *protocol* [*process-id*] to filter the routes of a certain routing protocol or a certain OSPF process. If *protocol* [*process-id*] is not specified, OSPF filters all the imported routes.

NOTE

- The **import-route** command cannot be used to import external default routes.
- OSPF filters the imported routes, and generates Type 5 LSAs to advertise only external routes that passing the filtering.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.6.3 Configuring IS-IS to Filter the Routes to Be Advertised

By configuring IS-IS to filter the routes to be advertised, you can effectively control the number of IS-IS routes on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Run:

```
filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix -name | route-
policy route-policy-name } export [ protocol [ process-id ] ]
```

IS-IS is configured to filter the imported routes that need to be advertised.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.6.4 Checking the Configuration

After filters are applied to the advertised routes, you can check information about the routing table of each protocol.

Prerequisite

All configurations of applying filters to the advertised routes are complete.

Procedure

- Run the **display rip** *process-id* **route** command to check information about the RIP routing table.
- Run the **display ospf** [*process-id*] **routing** command to check information about the OSPF routing table.
- Run the **display isis** [*process-id*] **route** command to check information about the IS-IS routing table.
- Run the **display ip routing-table** command to check information about the IP routing table.

  Run the **display ip routing-table** command on the neighbor, and you can view that the routes that meet the matching rules set on the neighbor are filtered or the actions defined by **apply** clauses are performed on these routes.

----End

10.7 Applying Filters to the Imported Routes

By applying the related filters of routing policies to routing protocols, you can filter the imported routes.

Applicable Environment

After defining the related filters (including the IP prefix list, ACL, and route-policy) of a routing policy, you need to apply these filters to routing protocols.

You can apply routing policies when external routes are imported:

- You can use the **import-route** command in the related protocol view, import the required external routes to the protocols, and apply the filters of a route-policy to the imported routes.
- After external routes are imported, you can run the **filter-policy export** command to filter the imported external routes. Only the routes that meet the matching rules are advertised.

NOTE

- BGP has the powerful filtering function. For the configuration of BGP routing policies, refer to "BGP Configuration."
- For details of the **filter-policy** and **import-route** commands and their applications in RIP, OSPF, IS-IS, and BGP, refer to related configurations.

Pre-configuration Tasks

Before applying filters to the imported routes, complete the following tasks:

- **Configuring an IP Prefix List**
- Configuring an ACL
- **Configuring a Route-Policy**

Configuration Procedures

You can choose to perform the following configuration tasks (except Checking the Configuration) according to the applicable environment.

Related Tasks

10.9.2 Example for Applying a Routing Policy When Routes Are Imported

10.7.1 Configuring RIP to Import External Routes

RIP can import the routing information from other processes or other routing protocols to enrich the RIP routing table.

Context

On a large-scale network, different routing protocols are configured for devices in different ASs. In this case, you need to import routes learnt by other protocols to the devices.

If RIP needs to advertise the routing information of other routing protocols (direct, static, OSPF, IS-IS, or BGP), you can specify *protocol* to filter the specific routing information. If *protocol* is not specified, the routing information to be advertised is filtered, including the imported routes and local RIP routes.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
rip [ process-id ]
```

The RIP process is created and the RIP view is displayed.

Step 3 (Optional) Run:

```
default-cost cost
```

The default cost is set for the imported routes.

If no cost is specified when external routes are imported, the default cost 0 is used.

Step 4 Run:

```
import-route protocol [ process-id ] [ cost cost | route-policy route-policy-name ]
*
```

External routes are imported.

Step 5 Run:

```
commit
```

The configuration is submitted.

----End

10.7.2 Configuring OSPF to Import External Routes

Importing the routes discovered by other routing protocols can enrich OSPF routing information.

Context

OSPF can ensure loop-free intra-area routes and inter-area routes; however, OSPF cannot protect external routes against loops. Therefore, when configuring OSPF to import external routes, avoid the loops caused by manual configurations.

Do as follows on the router that functions as the ASBR running OSPF:

Procedure

- Configuring OSPF to import the routes discovered by other protocols
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ospf [ process-id ]
     ```
     The OSPF process view is displayed.
  3. Run:
     ```
     import-route { protocol [ process-id ] [ cost cost | route-policy route-
     policy-name | tag tag | type type ] *}
     ```
     The routes discovered by other protocols are imported.
     - The parameter *protocol* specifies the routing protocol whose routes are imported. It can be **direct**, **static**, **rip**, **ospf**, **isis**, or **bgp**.
     - The parameter *process-id* specifies the process ID of the protocol whose routes are imported. The default value is 1.
     - The parameter **cost** *cost* specifies the cost of a route.
     - The parameter **type** *type* specifies the type of the metric. It can be 1 or 2.
     - The parameter **tag** *tag* specifies the tag in the external LSA.
     - The parameter **route-policy** *route-policy-name* indicates that the matching rules of the specified routing policy are applied.
  4. Run:

```
commit
```

The configuration is committed.

- Setting parameters for OSPF to import routes
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     ospf [ process-id ]
     ```
     The OSPF process view is displayed.
  3. Run:
     ```
     default { cost { cost | inherit-metric } | tag tag | type type } *
     ```
     The default values of parameters (the metric of routes, tag, and type) are set for importing routes.

     - The parameter **cost** *cost* specifies the default metric of the external route imported by OSPF.
     - The parameter **inherit-metric** indicates that the cost of the imported route is the cost carried in the route. If the cost is not specified, the default cost set through the **default** command is used as the cost of the imported route.

     When OSPF imports external routes, you can set default values for some additional parameters, such as the metric of routes to be imported, route tag, and route type. The route tag is used to identify the protocol-related information. For example, it can be used to differentiate AS numbers when OSPF receives BGP routes.

     By default, the default metric of the external routes imported by OSPF is 1; the type of the imported external routes is Type 2; the default tag value is 1.

     NOTE

     You can run one of the following commands to set the cost of the imported route. The following commands are listed in descending order of priority:

     - Run the **apply cost** command in a route-policy to set the cost of the imported route.
     - Run the **import-route** command for OSPF to set the cost of the imported route.
     - Run the **default** command to set the default cost of the imported route.
  4. Run:
     ```
     commit
     ```
     The configuration is committed.

----End

10.7.3 Configuring IS-IS to Import External Routes

By configuring IS-IS to import routes, you can enable IS-IS to learn routing information of other protocols or other IS-IS processes.

Context

IS-IS regards the routes discovered by other routing protocols or other IS-IS processes as external routes. When routes of other protocols are being imported, you can specify their default costs.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
isis [ process-id ]
```

The IS-IS view is displayed.

Step 3 Configure IS-IS to import external routes.

Before setting costs for the imported routes, you can run the **import-route** *protocol* [*process-id*] [**cost-type** { **external** | **internal** } | **cost** *cost* | **tag** *tag* | **route-policy** *route-policy-name* | [**level-1** | **level-2** | **level-1-2**]] * command to import external routes.

Before retaining the original costs of the imported routes, you can run the **import-route** { { **rip** | **isis** | **ospf** } [*process-id*] | **bgp** } **inherit-cost** [**tag** *tag* | **route-policy** *route-policy-name* | [**level-1** | **level-2** | **level-1-2**]] * command to import external routes.

If you do not specify a level for the imported routes, the level of the imported routes is Level-2.

NOTE

When the cost of a route to be imported is modified by using the routing policy, the following situations occur:

- If both route-policy route-policy-name and inherit-cost are configured, the cost modified by using the routing policy does not take effect.
- If both route-policy route-policy-name and cost cost are configured, the cost modified by using the routing policy is preferred.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.7.4 Checking the Configuration

After filters are applied to the imported routes, you can check information about the routing table of each protocol.

Prerequisite

Applying filters to the imported routes is complete.

Procedure

- Run the **display rip** *process-id* **route** command to check information about the RIP routing table.
- Run the **display ospf** [*process-id*] **routing** command to check information about the OSPF routing table.
- Run the **display isis** [*process-id*] **route** command to check information about the IS-IS routing table.

- Run the **display ip routing-table** command to check information about the IP routing table.

  Run the **display ip routing-table** command on the local router, and you can view that the routes that meet the matching rules set on the neighbor are filtered or the actions defined by **apply** clauses are performed on these routes.

----End

10.8 Setting the Delay for Applying a Routing Policy

To ensure network stability, you need to set the delay for applying a routing policy when modifying the routing policy.

Applicable Environment

In practice, when the configurations of multiple routing policies change, each routing protocol applies the new routing policy immediately after the configuration of a routing policy is complete. In this case, an incomplete routing policy will cause route flapping, waste time processing intermediate results, and result in network instability. Therefore, you need to set the delay for applying a routing policy.

The NE5000E provides the following rules for processing changes of a routing policy:

- By default, if a routing policy changes, routing protocols apply a new routing policy immediately.
- If the delay for applying a routing policy is set, when the commands used to configure the routing policy change, routing protocols do not process the changes immediately. Instead, routing protocols wait for the specified period and then apply a new routing policy.
- If the configuration of a routing policy changes again within the delay, the system resets the timer.

You can run related commands to set the delay as required.

Pre-configuration Tasks

Before setting the delay for applying a routing policy, complete the following task:

- Applying the routing policy

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
route-policy-change notify-delay delay-time
```

The delay for applying a routing policy is set.

The delay ranges from 1 to 180, in seconds.

By default, after a routing policy changes, routing protocols apply a new routing policy immediately.

Step 3 (Optional) Run:

```
refresh bgp all
```

BGP is configured to apply the new routing policy immediately.

After the delay for applying the routing policy is set, to view the filtering effect of the new routing policy immediately, you can run this command to configure BGP to immediately apply the new policy.

The filters affected by the set delay are: ACLs, IP prefix lists, AS_Path filters, community filters, extended community filters, RD filters, and route-policies.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configuration

After the preceding configurations are complete, you can do as follows to check the configurations:

- Run the **display current-configuration** command to check the delay for applying a routing policy.

You can view the currently set delay for applying a routing policy. For example:

```
<HUAWEI> display current-configuration
route-policy-change notify-delay 10
```

10.9 Configuration Examples

Routing policy configuration examples explain networking requirements, networking diagram, configuration notes, configuration roadmap, and configuration procedure.

10.9.1 Example for Filtering the Routes to Be Received or Advertised

Filters can be applied to the received and advertised routes according to networking requirements.

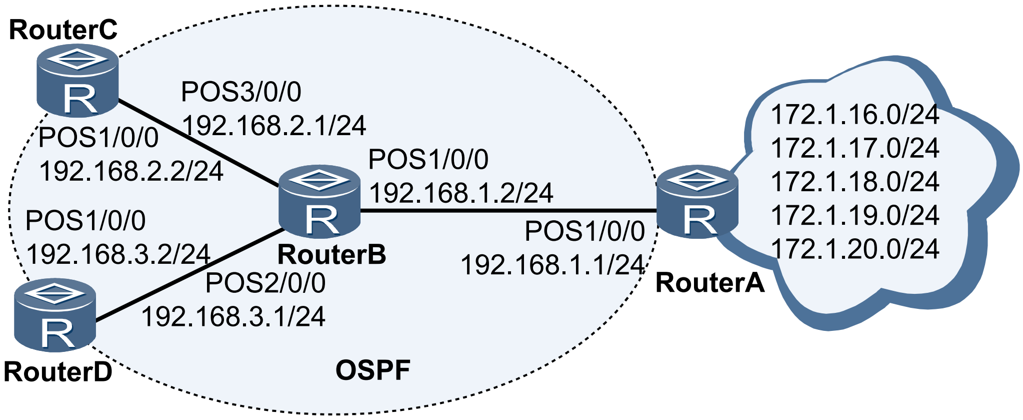
Networking Requirements

CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 10-2**, on the network running OSPF, Router A receives routes from the Internet and provides some Internet routes for Router B. It is required that Router A provide only 172.1.17.0/24, 172.1.18.0/24, and 172.1.19.0/24 for Router B; Router C receive only 172.1.18.0/24; Router D receive all the routes provided by Router B.

Figure 10-2 Networking diagram for filtering the received and advertised routes



Configuration Notes

When filtering the routes to be received or advertised, note the following:

- When configuring an IP prefix list, specify the IP prefix range as required.
- When applying an IP prefix list, ensure that the IP prefix list name is case sensitive.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic OSPF functions on Router A, Router B, Router C, and Router D.
2. Configure static routes on Router A and import these routes into OSPF.
3. Configure a policy for advertising routes on Router A and then check the filtering result on Router B.
4. Configure a policy for receiving routes on Router C and then check the filtering result on Router C.

Data Preparation

To complete the configuration, you need the following data:

- Five static routes imported by Router A
- Area 0 (backbone area), in which Router A, Router B, Router C, and Router D are located
- Name of the IP prefix list and routes to be filtered

Procedure

Step 1 Configure IP addresses for interfaces. The configuration details are not mentioned here.

Step 2 Configure OSPF.

# Configure Router A.

```
[~RouterA] ospf
[~RouterA-ospf-1] area 0
```

```
[~RouterA-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] commit
[~RouterA-ospf-1-area-0.0.0.0] quit
[~RouterA-ospf-1] quit
```

# Configure Router B.

```
[~RouterB] ospf
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.3.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] commit
[~RouterB-ospf-1-area-0.0.0.0] quit
```

# Configure Router C.

```
[~RouterC] ospf
[~RouterC-ospf-1] area 0
[~RouterC-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[~RouterC-ospf-1-area-0.0.0.0] commit
[~RouterC-ospf-1-area-0.0.0.0] quit
[~RouterC-ospf-1] quit
```

# Configure Router D.

```
[~RouterD] ospf
[~RouterD-ospf-1] area 0
[~RouterD-ospf-1-area-0.0.0.0] network 192.168.3.0 0.0.0.255
[~RouterD-ospf-1-area-0.0.0.0] commit
[~RouterD-ospf-1-area-0.0.0.0] quit
```

Step 3 Configure five static routes on Router A, and import these routes into OSPF.

```
[~RouterA] ip route-static 172.1.16.0 24 NULL0
[~RouterA] ip route-static 172.1.17.0 24 NULL0
[~RouterA] ip route-static 172.1.18.0 24 NULL0
[~RouterA] ip route-static 172.1.19.0 24 NULL0
[~RouterA] ip route-static 172.1.20.0 24 NULL0
[~RouterA] ospf
[~RouterA-ospf-1] import-route static
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

# Check the IP routing table on Router B. You can view that the five static routes are imported into OSPF.

```
[~RouterB] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : Public
         Destinations : 17       Routes : 17

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  Direct 0    0             D  127.0.0.1       LoopBack1
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
     172.1.16.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.17.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.18.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.19.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.20.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
    192.168.1.0/24  Direct 0    0             D  192.168.1.2     Pos1/0/0
    192.168.1.2/32  Direct 0    0             D  127.0.0.1       Pos1/0/0
  192.168.1.255/32  Direct 0    0             D  127.0.0.1       Pos1/0/0
    192.168.2.0/24  Direct 0    0             D  192.168.2.1     Pos3/0/0
    192.168.2.1/32  Direct 0    0             D  127.0.0.1       Pos3/0/0
    192.168.2.2/32  Direct 0    0             D  192.168.2.2     Pos3/0/0
    192.168.3.0/24  Direct 0    0             D  192.168.3.1     Pos2/0/0
    192.168.3.1/32  Direct 0    0             D  127.0.0.1       Pos2/0/0
  192.168.3.255/32  Direct 0    0             D  127.0.0.1       Pos2/0/0
```

Step 4 Configure a policy for advertising routes.

# Configure the IP prefix list named **a2b** on Router A.

```
[~RouterA] ip ip-prefix a2b index 10 permit 172.1.17.0 24
[~RouterA] ip ip-prefix a2b index 20 permit 172.1.18.0 24
[~RouterA] ip ip-prefix a2b index 30 permit 172.1.19.0 24
[~RouterA] commit
```

# Configure a policy for advertising routes on Router A, and use the IP prefix list named **a2b** to filter routes.

```
[~RouterA] ospf
[~RouterA-ospf-1] filter-policy ip-prefix a2b export static
[~RouterA-ospf-1] commit
[~RouterA-ospf-1] quit
```

# Check the IP routing table on Router B. You can view that Router B receives only three routes specified in the IP prefix list named **a2b**.

```
[~RouterB] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : Public
         Destinations : 15       Routes : 15

Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  Direct 0    0             D  127.0.0.1       LoopBack1
      127.0.0.0/8   Direct 0    0             D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0             D  127.0.0.1       InLoopBack0
     172.1.17.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.18.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
     172.1.19.0/24  O_ASE  150  1             D  192.168.1.1     Pos1/0/0
    192.168.1.0/24  Direct 0    0             D  192.168.1.2     Pos1/0/0
    192.168.1.2/32  Direct 0    0             D  127.0.0.1       Pos1/0/0
  192.168.1.255/32  Direct 0    0             D  127.0.0.1       Pos1/0/0
    192.168.2.0/24  Direct 0    0             D  192.168.2.1     Pos3/0/0
    192.168.2.1/32  Direct 0    0             D  127.0.0.1       Pos3/0/0
    192.168.2.2/32  Direct 0    0             D  192.168.2.2     Pos3/0/0
    192.168.3.0/24  Direct 0    0             D  192.168.3.1     Pos2/0/0
    192.168.3.1/32  Direct 0    0             D  127.0.0.1       Pos2/0/0
  192.168.3.255/32  Direct 0    0             D  127.0.0.1       Pos2/0/0
```

Step 5 Configure a policy for receiving routes.

# Configure the IP prefix list named **in** on Router C.

```
[~RouterC] ip ip-prefix in index 10 permit 172.1.18.0 24
[~RouterC] commit
```

# Configure a policy for receiving routes on Router C, and use the IP prefix list named **in** to filter routes.

```
[~RouterC] ospf
[~RouterC-ospf-1] filter-policy ip-prefix in import
[~RouterC-ospf-1] commit
```

# Check the IP routing table on Router C. In the local core routing table, you can view that Router C receives only one route specified in the IP prefix list named **in**.

```
[~RouterC] display ip routing-table
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Table : Public
         Destinations : 11       Routes : 11
```

```
Destination/Mask    Proto  Pre  Cost        Flags NextHop         Interface

        1.1.1.1/32  O_ASE   10   1             D  192.168.2.1     Pos1/0/0
      127.0.0.0/8   Direct 0    0            D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0            D  127.0.0.1       InLoopBack0
   172.1.18.0/24  O_ASE  150  1            D  192.168.2.1     Pos1/0/0
    192.168.1.0/24  O_ASE   10   2             D  192.168.2.1     Pos1/0/0
    192.168.2.0/24  Direct 0    0            D  192.168.2.2     Pos1/0/0
    192.168.2.1/32  Direct 0    0            D  192.168.2.1     Pos1/0/0
    192.168.2.2/32  Direct 0    0            D  127.0.0.1       Pos1/0/0
    192.168.3.0/24  O_ASE   10   2             D  192.168.2.1     Pos1/0/0
```

# Check the OSPF routing table on Router C. You can view that there are three routes specified in the IP prefix list named **a2b** in the OSPF routing table. In the link-state protocol, you can run the **filter-policy import** command to filter the routes to be installed into the local core routing table from the protocol routing table.

```
[~RouterC] display ospf routing

          OSPF Process 1 with Router ID 192.168.2.2
                   Routing Tables

 Routing for Network
 Destination        Cost       Type       NextHop         AdvRouter       Area

 1.1.1.1/32         1          Stub       192.168.2.1     1.1.1.1         0.0.0.
0
 192.168.1.0/24     2          Transit    192.168.2.1     192.168.1.1     0.0.0.
0
 192.168.3.0/24     2          Stub       192.168.2.1     1.1.1.1         0.0.0.
0

 Routing for ASEs
 Destination        Cost       Type       Tag        NextHop         AdvRouter

 172.1.17.0/24    1          Type2      1          192.168.2.1     192.168.1.1

 172.1.18.0/24    1          Type2      1          192.168.2.1     192.168.1.1

 172.1.19.0/24    1          Type2      1          192.168.2.1     192.168.1.1


 Total Nets: 6
 Intra Area: 3  Inter Area: 0  ASE: 3  NSSA: 0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
#
ospf 1
 filter-policy ip-prefix a2b export static
 import-route static
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
ip ip-prefix a2b index 10 permit 172.1.17.0 24
ip ip-prefix a2b index 20 permit 172.1.18.0 24
ip ip-prefix a2b index 30 permit 172.1.19.0 24
#
ip route-static 172.1.16.0 255.255.255.0 NULL0
ip route-static 172.1.17.0 255.255.255.0 NULL0
ip route-static 172.1.18.0 255.255.255.0 NULL0
ip route-static 172.1.19.0 255.255.255.0 NULL0
ip route-static 172.1.20.0 255.255.255.0 NULL0
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.3.1 255.255.255.0
#
interface Pos3/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
  network 192.168.3.0 0.0.0.255
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.2 255.255.255.0
#
ospf 1
 filter-policy ip-prefix in import
 area 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
ip ip-prefix in index 10 permit 172.1.18.0 24
#
return
```

- Configuration file of Router D

```
#
sysname RouterD
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.3.2 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.3.0 0.0.0.255
#
return
```

Related Tasks

10.5 Applying Filters to the Received Routes

10.6 Applying Filters to the Advertised Routes

10.9.2 Example for Applying a Routing Policy When Routes Are Imported

By applying routing policies, you can control the import of routes and set attributes for imported routes.

Networking Requirements

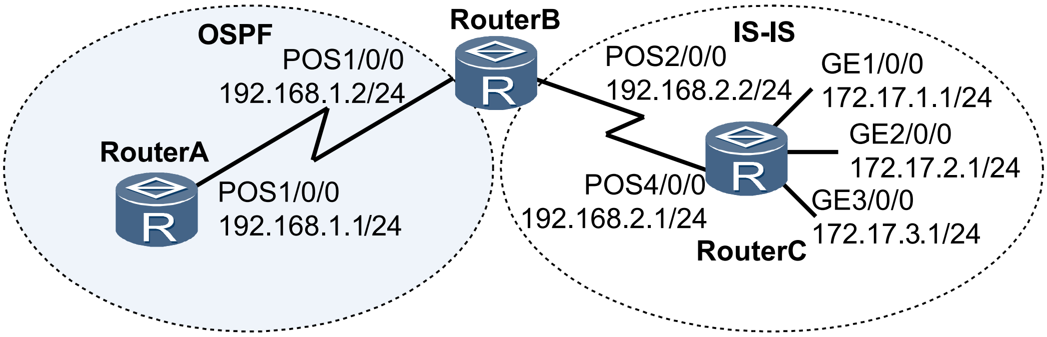
CAUTION

For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

As shown in **Figure 10-3**, Router A and Router B exchange routing information through OSPF; Router B and Router C exchange routing information through IS-IS.

Router B is required to import IS-IS routes into OSPF and use the routing policy to set route attributes. The cost of the route to 172.17.1.0/24 is set to 100, and the tag of the route to 172.17.2.0/24 is set to 20.

Figure 10-3 Networking diagram for applying a routing policy when routes are imported



Configuration Notes

When applying a routing policy when routes are imported, note the following:

- When configuring an IP prefix list, specify the IP prefix range as required.
- When applying a routing policy, ensure that the routing policy name is case sensitive.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic IS-IS functions on Router B and Router C.
2. Configure OSPF on Router A and Router B and import IS-IS routes.
3. Configure a routing policy on Router B, apply the routing policy when OSPF imports IS-IS routes, and check the routes.

Data Preparation

To complete the configuration, you need the following data:

- Area IDs, IS-IS levels, and system IDs of Router B and Router C
- Area 0 (backbone area), in which Router A and Router B are located
- ACL number, name of the IP prefix list, cost of the route to 172.17.1.0/24, and tag of the route to 172.17.2.0/24

Procedure

Step 1 Configure IP addresses for interfaces. The configuration details are not mentioned here.

Step 2 Configure IS-IS.

# Configure Router C.

```
[~RouterC] isis
[~RouterC-isis-1] is-level level-2
[~RouterC-isis-1] network-entity 10.0000.0000.0001.00
[~RouterC-isis-1] quit
[~RouterC] interface pos 4/0/0
[~RouterC-Pos4/0/0] isis enable
[~RouterC-Pos4/0/0] quit
[~RouterC] interface GigabitEthernet 1/0/0
[~RouterC-GigabitEthernet1/0/0] isis enable
[~RouterC-GigabitEthernet1/0/0] quit
[~RouterC] interface GigabitEthernet 2/0/0
[~RouterC-GigabitEthernet2/0/0] isis enable
[~RouterC-GigabitEthernet2/0/0] quit
[~RouterC] interface GigabitEthernet 3/0/0
[~RouterC-GigabitEthernet3/0/0] isis enable
[~RouterC-GigabitEthernet3/0/0] commit
[~RouterC-GigabitEthernet3/0/0] quit
```

# Configure Router B.

```
[~RouterB] isis
[~RouterB-isis-1] is-level level-2
[~RouterB-isis-1] network-entity 10.0000.0000.0002.00
[~RouterB-isis-1] quit
[~RouterB] interface pos 2/0/0
[~RouterB-Pos2/0/0] isis enable
[~RouterB-Pos2/0/0] commit
[~RouterB-Pos2/0/0] quit
```

Step 3 Configure OSPF and import routes.

# Configure Router A and enable OSPF.

```
[~RouterA] ospf
[~RouterA-ospf-1] area 0
[~RouterA-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterA-ospf-1-area-0.0.0.0] commit
[~RouterA-ospf-1-area-0.0.0.0] quit
[~RouterA-ospf-1] quit
```

# Configure Router B, enable OSPF, and import IS-IS routes.

```
[~RouterB] ospf
[~RouterB-ospf-1] area 0
[~RouterB-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[~RouterB-ospf-1-area-0.0.0.0] quit
[~RouterB-ospf-1] import-route isis 1
[~RouterB-ospf-1] commit
[~RouterB-ospf-1] quit
```

# Check the IP routing table on Router A. You can view the imported routes.

```
[~RouterA] display ospf routing
          OSPF Process 1 with Router ID 192.168.1.1
                   Routing Tables
 Routing for Network
 Destination        Cost  Type       NextHop         AdvRouter       Area
 192.168.1.0/24     1  Stub       192.168.1.1     192.168.1.1     0.0.0.0
 Routing for ASEs
 Destination        Cost      Type       Tag         NextHop         AdvRouter
 172.17.1.0/24      1        Type2      1           192.168.1.2     192.168.1.2
 172.17.2.0/24      1        Type2      1           192.168.1.2     192.168.1.2
 172.17.3.0/24      1        Type2      1           192.168.1.2     192.168.1.2
 192.168.2.0/24     1        Type2      1           192.168.1.2     192.168.1.2
 Routing for NSSAs
 Destination        Cost      Type       Tag         NextHop         AdvRouter
 Total Nets: 5
 Intra Area: 1  Inter Area: 0  ASE: 4  NSSA: 0
```

Step 4 Configure a filtering list.

# Configure an ACL numbered **2002** to allow the route to 172.17.2.0/24 to pass the filtering.

```
[~RouterB] acl number 2002
[~RouterB-acl-basic-2002] rule permit source 172.17.2.0 0.0.0.255
[~RouterB-acl-basic-2002] commit
[~RouterB-acl-basic-2002] quit
```

# Configure an IP prefix list named **prefix-a** to allow the route to 172.17.1.0/24 to pass the filtering.

```
[~RouterB] ip ip-prefix prefix-a index 10 permit 172.17.1.0 24
[~RouterB] commit
```

Step 5 Configure a route-policy.

```
[~RouterB] route-policy isis2ospf permit node 10
[~RouterB-route-policy] if-match ip-prefix prefix-a
[~RouterB-route-policy] apply cost 100
[~RouterB-route-policy] quit
[~RouterB] route-policy isis2ospf permit node 20
[~RouterB-route-policy] if-match acl 2002
[~RouterB-route-policy] apply tag 20
[~RouterB-route-policy] quit
[~RouterB] route-policy isis2ospf permit node 30
[~RouterB] commit
[~RouterB-route-policy] quit
```

Step 6 Apply the route-policy when routes are imported.

# Configure Router B and apply the route-policy when routes are imported.

```
[~RouterB] ospf
[~RouterB-ospf-1] import-route isis 1 route-policy isis2ospf
[~RouterB-ospf-1] commit
[~RouterB-ospf-1] quit
```

# Check the OSPF routing table on Router A. You can view that the cost of the route to 172.17.1.0/24 is 100; the tag of the route to 172.17.2.0/24 is 20; other route attributes do not change.

```
[~RouterA] display ospf routing
```

```
         OSPF Process 1 with Router ID 192.168.1.1
                  Routing Tables
 Routing for Network
 Destination        Cost  Type       NextHop         AdvRouter       Area
 192.168.1.0/24     1  Stub       192.168.1.1     192.168.1.1     0.0.0.0
 Routing for ASEs
 Destination        Cost      Type       Tag         NextHop         AdvRouter
 172.17.1.0/24      100       Type2      1           192.168.1.2     192.168.1.2
 172.17.2.0/24      1          Type2      20          192.168.1.2     192.168.1.2
 172.17.3.0/24      1          Type2      1           192.168.1.2     192.168.1.2
 192.168.2.0/24     1          Type2      1           192.168.1.2     192.168.1.2
 Routing for NSSAs
 Destination        Cost      Type       Tag         NextHop         AdvRouter
 Total Nets: 5
 Intra Area: 1  Inter Area: 0  ASE: 4  NSSA: 0
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.1 255.255.255.0
#
ospf 1
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Router B

```
#
sysname RouterB
#
acl number 2002
 rule 5 permit source 172.17.2.0 0.0.0.255
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0002.00
#
interface Pos1/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.1.2 255.255.255.0
#
interface Pos2/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.2 255.255.255.0
 isis enable 1
#
ospf 1
 import-route isis 1 route-policy isis2ospf
 area 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
route-policy isis2ospf permit node 10
 if-match ip-prefix prefix-a
 apply cost 100
#
route-policy isis2ospf permit node 20
 if-match acl 2002
 apply tag 20
#
route-policy isis2ospf permit node 30
#
ip ip-prefix prefix-a index 10 permit 172.17.1.0 24
#
return
```

- Configuration file of Router C

```
#
sysname RouterC
#
isis 1
 is-level level-2
 network-entity 10.0000.0000.0001.00
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 172.17.1.1 255.255.255.0
 isis enable 1
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 172.17.2.1 255.255.255.0
 isis enable 1
#
interface GigabitEthernet3/0/0
 undo shutdown
 ip address 172.17.3.1 255.255.255.0
 isis enable 1
#
interface Pos4/0/0
 undo shutdown
 link-protocol ppp
 ip address 192.168.2.1 255.255.255.0
 isis enable 1
#
return
```

Related Tasks

10.7 Applying Filters to the Imported Routes

10.9.3 Example for Applying the Routing Policy

By configuring routing policies, you can flexibly control the traffic on a complex network.

Networking Requirements

CAUTION

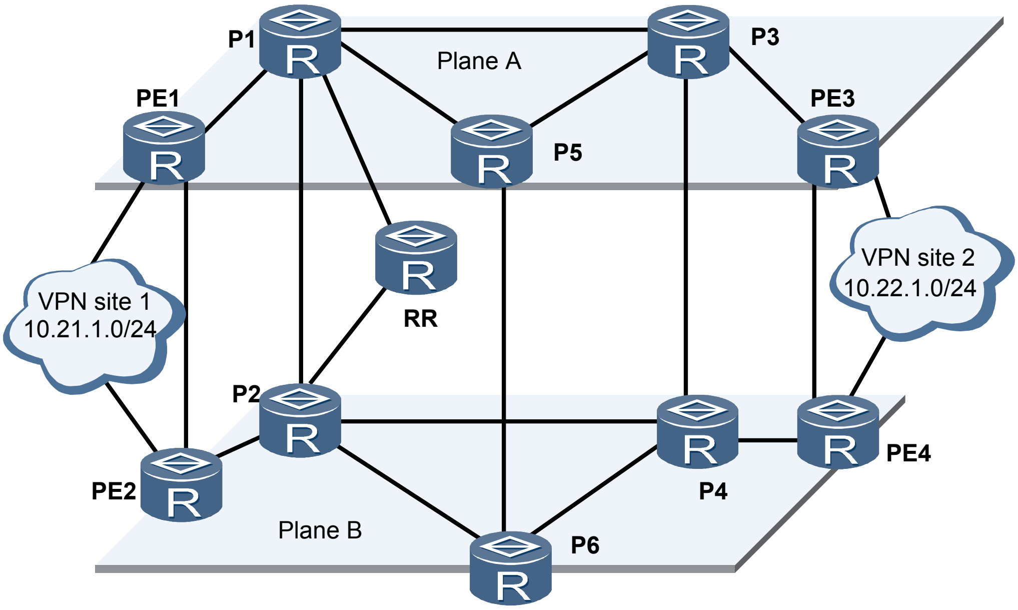
For the NE5000E, the interface is numbered as slot number/card number/interface number. For the NE5000E cluster, the interface is numbered as chassis ID/slot number/card number/interface number. The slot number is chassis ID/slot ID.

Figure 10-4 shows a simplified diagram of the MPLS network that carries multiple types of L3VPN services, such as multimedia, signaling, and accounting. In **Figure 10-4**, two sites, each of which has two PEs accessing the core layer, are taken as an example. The core layer is divided into two planes, each of which has three full-meshed P nodes. Nodes in different planes are connected to provide backup paths. MP-BGP is used to advertise inner tags and VPNv4 routes between the PEs. Each PE needs to establish the MP-IBGP peer relationship with a route reflector (RR).

NOTE

Figure 10-4 is a simplified networking diagram, in which there are two sites, one RR, and two planes with six P nodes. On the actual network, there are 14 sites with 28 PEs, and each plane has four P nodes and two RRs. Therefore, each RR needs to establish MP-IBGP connections with 28 PEs.

Figure 10-4 Networking diagram for applying the routing policy



In Figure 10-4, each PE sends BGP Update messages to an RR, and the other PEs receive BGP Update messages from different planes. Therefore, routing policies need to be applied to ensure that a VPN traffic flow is transmitted through only one plane.

Configuration Notes

When applying routing policies, note the following:

- Two PEs in the same site must be configured with different route distinguishers (RDs).
- The routes advertised by PEs in different planes need to be assigned different community attributes.
- The **undo policy vpn-target** command needs to be run in the BGP-VPNv4 address family view to ensure that VPN-target-based filtering is not performed on VPNv4 routes.
- When applying a routing policy, ensure that the routing policy name is case sensitive.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure different RDs for two PEs in the same site to ensure that each PE can receive two routes from different BGP next hops in the remote site. When two PEs in a site advertise

the routes to the same destination, configuring different RDs for the two PEs can ensure that BGP peers consider the advertised routes as two different routes. This is because BGP-VPNv4 uses the VPNv4 addresses that consist of IPv4 addresses and RDs.

2. Assign different community attributes to the routes advertised by the PE in plane A and the routes advertised by the PE in plane B.
3. Set different local preferences for routes based on the community attributes of the routes. In this manner, the PE in plane A always prefers the routes advertised by the remote PE in plane A, and the PE in plane B always prefers the routes advertised by the remote PE in plane B.

Data Preparation

To complete the configuration, you need the following data.

Table 10-2 IP addresses of physical interfaces

Local Device	Local Interface and Its IP Address	Remote Interface and Its IP Address	Remote Device
P1	GE1/0/0 10.1.1.1/30	GE1/0/0 10.1.1.2/30	P3
P1	GE2/0/0 10.1.2.1/30	GE1/0/0 10.1.2.2/30	P5
P1	GE3/0/0 10.1.3.1/30	GE1/0/0 10.1.3.2/30	RR
P1	GE4/0/0 10.1.4.1/30	GE1/0/0 10.1.4.2/30	P2
P1	GE5/0/0 10.1.5.1/30	GE1/0/0 10.1.5.2/30	PE1
P2	GE4/0/0 10.1.6.1/30	GE1/0/0 10.1.6.2/30	P6
P2	GE3/0/0 10.1.7.1/30	GE1/0/0 10.1.7.2/30	P4
P2	GE2/0/0 10.1.8.1/30	GE2/0/0 10.1.8.2/30	RR
P2	GE5/0/0 10.1.9.1/30	GE1/0/0 10.1.9.2/30	PE2
P3	GE2/0/0 10.1.10.1/30	GE2/0/0 10.1.10.2/30	P5
P3	GE3/0/0 10.1.11.1/30	GE2/0/0 10.1.11.2/30	P4

Local Device	Local Interface and Its IP Address	Remote Interface and Its IP Address	Remote Device
P3	GE4/0/0 10.1.12.1/30	GE1/0/0 10.1.12.2/30	PE3
P4	GE3/0/0 10.1.13.1/30	GE3/0/0 10.1.13.2/30	P6
P4	GE4/0/0 10.1.14.1/30	GE1/0/0 10.1.14.2/30	PE4
P5	GE3/0/0 10.1.15.1/30	GE2/0/0 10.1.15.2/30	P6
PE1	GE2/0/0 10.1.16.1/30	GE2/0/0 10.1.16.2/30	PE2
PE3	GE2/0/0 10.1.17.1/30	GE2/0/0 10.1.17.2/30	PE4

Table 10-3 IP addresses of loopback interfaces

Local Device	IP Address of Local Loopback 0 Interface	Remote Device	IP Address of Remote Loopback 0 Interface
P1	10.1.1.9/32	P2	10.2.2.9/32
P3	10.3.3.9/32	P4	10.4.4.9/32
P5	10.5.5.9/32	P6	10.6.6.9/32
PE1	10.7.7.9/32	PE2	10.8.8.9/32
PE3	10.9.9.9/32	PE4	10.10.10.9/32
RR	10.11.11.9/32		

Table 10-4 BGP parameter values

BGP Parameter	Value
AS number	65000
Router ID	Same as the address of Loopback 0 interface
BGP community attribute	Plane A: 65000:100 Plane B: 65000:200

BGP Parameter	Value
BGP local preference	Plane A: The local preference of community attribute 65000:100 is set to 200. Plane B: The local preference of community attribute 65000:200 is set to 200. **NOTE** By default, the BGP local preference is 100. The greater the value, the higher the preference.
Routing policy name	Policy for importing routes: **local_pre** Policy for advertising routes: **comm**
Community filter name	1
BGP peer group name	Client

Procedure

Step 1 Configure names for devices and IP addresses for interfaces.

For detailed configurations, see the configuration files in this example.

Step 2 Configure an IGP.

In this example, IS-IS is used. For detailed configurations, see the configuration files in this example.

After the configuration, run the **display ip routing-table** command. You can view that PEs, P nodes and PEs, and P nodes have learned the addresses of Loopback 0 interfaces from each other.

Step 3 Establish MP-IBGP connections between PEs and RRs.

# Take the configuration of PE1 as an example. Configurations of other PEs are the same as that of PE1 and are not mentioned here.

```
[~PE1] bgp 65000
[~PE1-bgp] peer 10.11.11.9 as-number 65000
[~PE1-bgp] peer 10.11.11.9 connect-interface LoopBack0
[~PE1-bgp] ipv4-family unicast
[~PE1-bgp-af-ipv4] undo peer 10.11.11.9 enable
[~PE1-bgp] ipv4-family vpnv4
[~PE1-bgp-af-vpnv4] peer 10.11.11.9 enable
[~PE1-bgp-af-vpnv4] commit
```

# Configure the RR.

```
[~RR] bgp 65000
[~RR-bgp] group client internal
[~RR-bgp] peer client connect-interface LoopBack0
[~RR-bgp] ipv4-family unicast
[~RR-bgp-af-ipv4] undo peer client enable
[~RR-bgp-af-ipv4] quit
[~RR-bgp] ipv4-family vpnv4
[~RR-bgp-af-vpnv4] undo policy vpn-target
[~RR-bgp-af-vpnv4] peer client enable
[~RR-bgp-af-vpnv4] peer 10.7.7.9 group client
[~RR-bgp-af-vpnv4] peer 10.8.8.9 group client
[~RR-bgp-af-vpnv4] peer 10.9.9.9 group client
```

```
[~RR-bgp-af-vpnv4] peer 10.10.10.9 group client
[~RR-bgp-af-vpnv4] peer client reflect-client
[~RR-bgp-af-vpnv4] commit
[~RR-bgp-af-vpnv4] quit
```

NOTE

The **undo policy vpn-target** command needs to be run in the BGP-VPNv4 address family view of the RR to ensure that VPN-target-based filtering is not performed on VPNv4 routes. By default, an RR performs VPN-target-based filtering on the received VPNv4 routes. The matching routes are installed into the VPN routing table, and other routes are discarded. In this example, no VPN instances are configured on the RR. In this case, if VPN-target-based filtering is enabled, all the received VPNv4 routes will be discarded.

After the configuration, run the **display bgp vpnv4 all peer** command on the RR. You can view that the RR establishes MP-IBGP connections with all PEs.

```
<RR> display bgp vpnv4 all peer
 BGP local router ID : 10.11.11.9
 Local AS number : 65000
 Total number of peers : 4                 Peers in established state : 4
  Peer          V    AS     MsgRcvd  MsgSent  OutQ  Up/Down       State
PrefRcv
  10.7.7.9      4    65000  79       82        0     00:01:31     Established
0
  10.8.8.9      4    65000  42       66        0     00:01:16     Established
0
  10.9.9.9      4    65000  21       34        0     00:00:50     Established
0
  10.10.10.9    4    65000  2         4        0     00:00:21     Established
0
```

Step 4 Configure routing policies.

NOTE

Take the configurations of PE1, PE2, and the RR as an example. The configurations of PE3 and PE4 are the same as the configurations of PE1 and PE2 respectively, and are not mentioned here.

# Configure a routing policy on PE1 so that the route advertised by the PE in plane A to the RR can carry community attribute 65000:100.

```
[~PE1] route-policy comm permit node 10
[~PE1] apply community 65000:100
[~PE1] commit
```

# Configure a routing policy on PE2 so that the route advertised by the PE in plane B to the RR can carry community attribute 65000:200.

```
[~PE2] route-policy comm permit node 10
[~PE2] apply community 65000:200
[~PE2] commit
```

# On PE1, apply the routing policy to the advertised BGP VPNv4 route so that the community attribute can be advertised to the RR.

```
[~PE1] bgp 65000
[~PE1-bgp] ipv4-family vpnv4
[~PE1-bgp-af-vpnv4] peer 10.11.11.9 route-policy comm export
[~PE1-bgp-af-vpnv4] peer 10.11.11.9 advertise-community
[~PE1-bgp-af-vpnv4] commit
```

# On PE2, apply the routing policy to the advertised BGP VPNv4 route so that the community attribute can be advertised to the RR.

```
[~PE2] bgp 65000
[~PE2-bgp] ipv4-family vpnv4
[~PE2-bgp-af-vpnv4] peer 10.11.11.9 route-policy comm export
[~PE2-bgp-af-vpnv4] peer 10.11.11.9 advertise-community
[~PE2-bgp-af-vpnv4] commit
```

# Configure the RR to advertise the community attributes to the PEs.

```
[~RR] bgp 65000
[~RR-bgp] ipv4-family vpnv4
[~RR-bgp-af-vpnv4] peer client advertise-community
[~RR-bgp-af-vpnv4] commit
```

# Configure a community filter on PE1.

```
[~PE1] ip community-filter 1 permit 65000:100
[~PE1] commit
```

# Configure a community filter on PE2.

```
[~PE2] ip community-filter 1 permit 65000:200
[~PE2] commit
```

# On PE1, configure a routing policy and set the local preference of the route with community attribute 65000:100 to 200.

```
[~PE1] route-policy local_pre permit node 10
[~PE1-route-policy] if-match community-filter 1
[~PE1-route-policy] apply local-preference 200
[~PE1-route-policy] commit
[~PE1-route-policy] quit
```

# On PE2, configure a routing policy and set the local preference of the route with community attribute 65000:200 to 200.

```
[~PE2] route-policy local_pre permit node 10
[~PE2-route-policy] if-match community-filter 1
[~PE2-route-policy] apply local-preference 200
[~PE2-route-policy] commit
[~PE2-route-policy] quit
```

# On PE1, apply the routing policy to the imported BGP VPNv4 route so that the PE in plane A prefers the route advertised by the remote PE in plane A.

```
[~PE1] bgp 65000
[~PE1-bgp] ipv4-family vpnv4
[~PE1-bgp-af-vpnv4] peer 10.11.11.9 route-policy local_pre import
[~PE1-bgp-af-vpnv4] commit
```

# On PE2, apply the routing policy to the imported BGP VPNv4 route so that the PE in plane B prefers the route advertised by the remote PE in plane B.

```
[~PE2] bgp 65000
[~PE2-bgp] ipv4-family vpnv4
[~PE2-bgp-af-vpnv4] peer 10.11.11.9 route-policy local_pre import
[~PE2-bgp-af-vpnv4] commit
```

NOTE

After this configuration, you also need to configure MPLS, establish tunnels, configure MPLS L3VPN functions, and connect the PEs to CEs. For detailed configurations, see the configuration files in this example.

Step 5 Verify the configuration.

# Run the **display bgp vpnv4 all routing-table community** command on a PE. You can view information about the VPNv4 routes with community attributes. Take the display on PE1 and PE2 as an example.

```
[~PE1] display bgp vpnv4 all routing-table community
Total Number of Routes from all PE: 2
BGP Local router ID is 10.7.7.9
Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Route Distinguisher: 65000:10001012
```

```
       Network        NextHop        MED        LocPrf    PrefVal  Community
*>   10.22.1.0/24     10.9.9.9         0            200             65000:100
*                     10.10.10.9       0            100             65000:200

 Total routes of vpn-instance NGN_Media: 2
      Network         NextHop        MED        LocPrf    PrefVal  Community
 *>i  10.22.1.0/24    10.9.9.9         0            200       0     65000:100
*                     10.10.10.9       0            100       0     65000:200

[~PE2] display bgp vpnv4 all routing-table community
Total Number of Routes from all PE: 2
BGP Local router ID is 10.8.8.9
Status codes: * - valid, > - best, d - damped,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
 Route Distinguisher: 65000:10001011
       Network        NextHop        MED        LocPrf    PrefVal  Community
*>   10.22.1.0/24     10.10.10.9       0           200              65000:200
*                     10.9.9.9         0           100              65000:100

 Total routes of vpn-instance NGN_Media: 2
      Network         NextHop        MED        LocPrf    PrefVal  Community
 *>i  10.22.1.0/24    10.10.10.9       0          200        0      65000:200
*                     10.9.9.9         0          100        0      65000:100
```

# Run the **display ip routing-table vpn-instance NGN_Media 10.22.1.0 24** command on PE1. You can view that the next hop of the route to 10.22.1.0/24 is PE3. That is, PE1 prefers the route advertised by PE3.

```
[~PE1] display ip routing-table vpn-instance NGN_Media 10.22.1.0 24
Route Flags: R - relay, D - download for forwarding
------------------------------------------------------------------------------
Routing Tables: NGN_Media
Destination/Mask  Proto  Pre  Cost  Flags  NextHop   Interface
   10.22.1.0/24   BGP    255  0       RD   10.9.9.9  GigabitEthernet1/0/0
```

----End

Configuration File

- Configuration file of P1

```
#
sysname P1
#
mpls lsr-id 10.1.1.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0100.1009.00
#
interface GigabitEthernet1/0/0
 description toP3GE1/0/0
 undo shutdown
 ip address 10.1.1.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP5GE1/0/0
 undo shutdown
 ip address 10.1.2.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
```

```
#
interface GigabitEthernet3/0/0
 description toRRGE1/0/0
 undo shutdown
 ip address 10.1.3.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet4/0/0
 description toP2GE1/0/0
 undo shutdown
 ip address 10.1.4.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet5/0/0
 description toP2GE1/0/0
 undo shutdown
 ip address 10.1.5.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.1.1.9 255.255.255.255
 isis enable 64
#
return
```

- Configuration file of P2

```
#
sysname P2
#
mpls lsr-id 10.2.2.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0200.2009.00
#
interface GigabitEthernet1/0/0
 description toP1GE4/0/0
 undo shutdown
 ip address 10.1.4.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toRRGE2/0/0
 undo shutdown
 ip address 10.1.8.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 description toP4GE1/0/0
 undo shutdown
 ip address 10.1.7.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet4/0/0
 description toP6GE1/0/0
 undo shutdown
 ip address 10.1.6.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet5/0/0
 description toPE2GE1/0/0
 undo shutdown
 ip address 10.1.9.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.2.2.9 255.255.255.255
 isis enable 64
#
return
```

- Configuration file of P3

```
#
sysname P3
#
mpls lsr-id 10.3.3.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0300.3009.00
#
interface GigabitEthernet1/0/0
 description toP1GE1/0/0
 undo shutdown
 ip address 10.1.1.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP5GE2/0/0
 undo shutdown
 ip address 10.1.10.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 description toP4GE2/0/0
 undo shutdown
 ip address 10.1.11.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet4/0/0
 description toPE3GE1/0/0
 undo shutdown
 ip address 10.1.12.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.3.3.9 255.255.255.255
 isis enable 64
```

```
#
return
```

- Configuration file of P4

```
#
sysname P4
#
mpls lsr-id 10.4.4.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0400.4009.00
#
interface GigabitEthernet1/0/0
 description toP2GE3/0/0
 undo shutdown
 ip address 10.1.7.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP3GE3/0/0
 undo shutdown
 ip address 10.1.11.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 description toP6GE3/0/0
 undo shutdown
 ip address 10.1.13.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet4/0/0
 description toPE4GE1/0/0
 undo shutdown
 ip address 10.1.14.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.4.4.9 255.255.255.255
 isis enable 64
#
return
```

- Configuration file of P5

```
#
sysname P5
#
mpls lsr-id 10.5.5.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0500.5009.00
#
interface GigabitEthernet1/0/0
 description toP1GE2/0/0
 undo shutdown
 ip address 10.1.2.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP3GE2/0/0
 undo shutdown
 ip address 10.1.10.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 description toP6GE2/0/0
 undo shutdown
 ip address 10.1.15.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.5.5.9 255.255.255.255
 isis enable 64
#
return
```

- Configuration file of P6

```
#
sysname P6
#
mpls lsr-id 10.6.6.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0600.6009.00
#
interface GigabitEthernet1/0/0
 description toP2GE4/0/0
 undo shutdown
 ip address 10.1.6.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP5GE3/0/0
 undo shutdown
 ip address 10.1.15.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 description toP4GE3/0/0
 undo shutdown
 ip address 10.1.13.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface LoopBack0
 ip address 10.6.6.9 255.255.255.255
 isis enable 64
#
return
```

- Configuration file of PE1

```
#
sysname PE1
#
ip vpn-instance NGN_Media
 route-distinguisher 65000:10001012
 apply-label per-instance
 vpn-target 65000:100 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Other
 route-distinguisher 65000:30001012
 apply-label per-instance
 vpn-target 65000:300 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Signaling
 route-distinguisher 65000:20001012
 apply-label per-instance
 vpn-target 65000:200 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
#
mpls lsr-id 10.7.7.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0700.7009.00
#
interface GigabitEthernet1/0/0
 description toP1GE5/0/0
 undo shutdown
 ip address 10.1.5.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toPE2GE2/0/0
 undo shutdown
 ip address 10.1.16.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 undo shutdown
#
interface GigabitEthernet3/0/0.10
 ip binding vpn-instance NGN_Media
 vlan-type dot1q 10
 ip address 10.21.1.73 255.255.255.252
#
interface GigabitEthernet3/0/0.11
 ip binding vpn-instance NGN_Signaling
 vlan-type dot1q 11
 ip address 10.21.1.77 255.255.255.252
#
interface GigabitEthernet3/0/0.12
 ip binding vpn-instance NGN_Other
 vlan-type dot1q 12
 ip address 10.21.1.81 255.255.255.252
#
```

```
interface LoopBack0
 ip address 10.7.7.9 255.255.255.255
 isis enable 64
#
bgp 65000
 peer 10.11.11.9 as-number 65000
 peer 10.11.11.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  undo peer 10.11.11.9 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 10.11.11.9 enable
  peer 10.11.11.9 route-policy local_pre import
  peer 10.11.11.9 route-policy comm export
  peer 10.11.11.9 advertise-community
 #
 ipv4-family vpn-instance NGN_Media
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Other
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Signaling
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
#
route-policy comm permit node 10
 apply community 65000:100
#
route-policy local_pre permit node 10
 if-match community-filter 1
 apply local-preference 200
#
ip community-filter 1 permit 65000:100
#
return
```

- Configuration file of PE2

```
#
sysname PE2
#
ip vpn-instance NGN_Media
 route-distinguisher 65000:10001011
 apply-label per-instance
 vpn-target 65000:100 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Other
 route-distinguisher 65000:30001011
 apply-label per-instance
 vpn-target 65000:300 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Signaling
 route-distinguisher 65000:20001011
 apply-label per-instance
 vpn-target 65000:200 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
#
mpls lsr-id 10.8.8.9
#
```

```
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0800.8009.00
#
interface GigabitEthernet1/0/0
 description toP2GE5/0/0
 undo shutdown
 ip address 10.1.9.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toPE1GE2/0/0
 undo shutdown
 ip address 10.1.16.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 undo shutdown
#
interface GigabitEthernet3/0/0.10
 ip binding vpn-instance NGN_Media
 vlan-type dot1q 10
 ip address 10.21.1.13 255.255.255.252
#
interface GigabitEthernet3/0/0.11
 ip binding vpn-instance NGN_Signaling
 vlan-type dot1q 11
 ip address 10.21.1.17 255.255.255.252
#
interface GigabitEthernet3/0/0.12
 ip binding vpn-instance NGN_Other
 vlan-type dot1q 12
 ip address 10.21.1.21 255.255.255.252
#
interface LoopBack0
 ip address 10.8.8.9 255.255.255.255
 isis enable 64
#
bgp 65000
 peer 10.11.11.9 as-number 65000
 peer 10.11.11.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  undo peer 10.11.11.9 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 10.11.11.9 enable
  peer 10.11.11.9 route-policy local_pre import
  peer 10.11.11.9 route-policy comm export
  peer 10.11.11.9 advertise-community
 #
 ipv4-family vpn-instance NGN_Media
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Other
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Signaling
```

```
  aggregate 10.21.1.0 255.255.255.0 detail-suppressed
  import-route direct
#
route-policy comm permit node 10
 apply community 65000:200
#
route-policy local_pre permit node 10
 if-match community-filter 1
 apply local-preference 200
#
ip community-filter 1 permit 65000:200
#
return
```

- Configuration file of PE3

```
#
sysname PE3
#
ip vpn-instance NGN_Media
 route-distinguisher 65000:10000811
 apply-label per-instance
 vpn-target 65000:100 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Other
 route-distinguisher 65000:30000811
 apply-label per-instance
 vpn-target 65000:300 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Signaling
 route-distinguisher 65000:20000811
 apply-label per-instance
 vpn-target 65000:200 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
#
mpls lsr-id 10.9.9.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.0900.9009.00
#
interface GigabitEthernet1/0/0
 description toP3GE4/0/0
 undo shutdown
 ip address 10.1.12.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toPE4GE2/0/0
 undo shutdown
 ip address 10.1.17.1 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 undo shutdown
#
interface GigabitEthernet3/0/0.10
 ip binding vpn-instance NGN_Media
 vlan-type dot1q 10
 ip address 10.22.1.73 255.255.255.252
#
interface GigabitEthernet3/0/0.11
 ip binding vpn-instance NGN_Signaling
 vlan-type dot1q 11
 ip address 10.22.1.77 255.255.255.252
#
interface GigabitEthernet3/0/0.12
 ip binding vpn-instance NGN_Other
 vlan-type dot1q 12
 ip address 10.22.1.81 255.255.255.252
#
interface LoopBack0
 ip address 10.9.9.9 255.255.255.255
 isis enable 64
#
bgp 65000
 peer 10.11.11.9 as-number 65000
 peer 10.11.11.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  undo peer 10.11.11.9 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 10.11.11.9 enable
  peer 10.11.11.9 route-policy local_pre import
  peer 10.11.11.9 route-policy comm export
  peer 10.11.11.9 advertise-community
 #
 ipv4-family vpn-instance NGN_Media
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Other
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Signaling
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
#
route-policy comm permit node 10
 apply community 65000:100
#
route-policy local_pre permit node 10
 if-match community-filter 1
 apply local-preference 200
#
route-policy local_pre permit node 20
#
ip community-filter 1 permit 65000:100
#
return
```

- Configuration file of PE4

```
#
sysname PE4
#
ip vpn-instance NGN_Media
 route-distinguisher 65000:10000712
 apply-label per-instance
 vpn-target 65000:100 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Other
 route-distinguisher 65000:30000712
 apply-label per-instance
 vpn-target 65000:300 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
ip vpn-instance NGN_Signaling
 route-distinguisher 65000:20000712
 apply-label per-instance
 vpn-target 65000:200 export-extcommunity
 vpn-target 65000:100 import-extcommunity
 vpn-target 65000:200 import-extcommunity
 vpn-target 65000:300 import-extcommunity
#
mpls lsr-id 10.10.10.9
#
mpls
#
mpls ldp
#
isis 64
 network-entity 49.0091.0100.1001.0009.00
#
interface GigabitEthernet1/0/0
 description toP4GE4/0/0
 undo shutdown
 ip address 10.1.14.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toPE3GE2/0/0
 undo shutdown
 ip address 10.1.17.2 255.255.255.252
 mpls
 mpls ldp
 isis enable 64
#
interface GigabitEthernet3/0/0
 undo shutdown
#
interface GigabitEthernet3/0/0.10
 ip binding vpn-instance NGN_Media
 vlan-type dot1q 10
 ip address 10.22.1.13 255.255.255.252
#
interface GigabitEthernet3/0/0.11
 ip binding vpn-instance NGN_Signaling
 vlan-type dot1q 11
 ip address 10.22.1.17 255.255.255.252
#
interface GigabitEthernet3/0/0.12
 ip binding vpn-instance NGN_Other
 vlan-type dot1q 12
 ip address 10.22.1.21 255.255.255.252
#
interface LoopBack0
 ip address 10.10.10.9 255.255.255.255
 isis enable 64
#
bgp 65000
 peer 10.11.11.9 as-number 65000
 peer 10.11.11.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  undo peer 10.11.11.9 enable
 #
 ipv4-family vpnv4
```

```
  policy vpn-target
  peer 10.11.11.9 enable
  peer 10.11.11.9 route-policy local_pre import
  peer 10.11.11.9 route-policy comm export
  peer 10.11.11.9 advertise-community
 #
 ipv4-family vpn-instance NGN_Media
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Other
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
 #
 ipv4-family vpn-instance NGN_Signaling
  aggregate 10.22.1.0 255.255.255.0 detail-suppressed
  import-route direct
#
route-policy comm permit node 10
 apply community 65000:200
#
route-policy local_pre permit node 10
 if-match community-filter 1
 apply local-preference 200
#
ip community-filter 1 permit 65000:200
#
return
```

- Configuration file of the RR

```
#
sysname RR
#
isis 64
 network-entity 49.0091.0100.1101.1009.00
#
interface GigabitEthernet1/0/0
 description toP1GE3/0/0
 undo shutdown
 ip address 10.1.3.2 255.255.255.252
 isis enable 64
#
interface GigabitEthernet2/0/0
 description toP2GE2/0/0
 undo shutdown
 ip address 10.1.8.2 255.255.255.252
 isis enable 64
#
interface LoopBack0
 ip address 10.11.11.9 255.255.255.255
 isis enable 64
#
bgp 65000
 group client internal
 peer client connect-interface LoopBack0
 peer 10.7.7.9 as-number 65000
 peer 10.8.8.9 as-number 65000
 peer 10.9.9.9 as-number 65000
 peer 10.10.10.9 as-number 65000
#
 ipv4-family unicast
  undo synchronization
  undo peer client enable
  undo peer 10.7.7.9 enable
  undo peer 10.8.8.9 enable
  undo peer 10.9.9.9 enable
  undo peer 10.10.10.9 enable
 #
 ipv4-family vpnv4
  undo policy vpn-target
  peer client enable
  peer client reflect-client
  peer client advertise-community
  peer 10.7.7.9 enable
  peer 10.7.7.9 group client
  peer 10.8.8.9 enable
  peer 10.8.8.9 group client
  peer 10.9.9.9 enable
  peer 10.9.9.9 group client
  peer 10.10.10.9 enable
  peer 10.10.10.9 group client
#
return
```

Related Tasks

10.4 Configuring a Route-Policy