9 BGP Configuration

About This Chapter

The Border Gateway Protocol (BGP) is used between Autonomous Systems (ASs) to transmit routing information. BGP applies to large and complex networks.

You can configure BGP Tracking, association between BGP and BFD, and BGP GR to speed up BGP network convergence and improve BGP reliability.

9.13 Configuring BGP Route Summarization
On IPv4 networks, BGP supports automatic route summarization and manual route summarization. Manual route summarization takes precedence over automatic route summarization. On IPv6 networks, BGP supports only manual route summarization.

9.14 Configuring On-demand Route Advertisement
If a BGP device only wants to received required routes but its peer cannot maintain different export policies for connected devices, you can configure prefix-based BGP outbound route filtering (ORF) to meet this requirement.

9.15 Configuring BGP to Advertise Default Routes to Peers
If a BGP device needs to send multiple routes to its peer, the BGP device can be configured to send only a default route with the local address as the next-hop address to its peer, regardless of whether there are default routes in the local routing table. This function reduces the number of network routes and saves memory and network resources.

9.16 Configuring Path MTU Auto Discovery
BGP path maximum transmission unit (MTU) auto discovery can discover the minimum MTU (path MTU) on the network path from the source to the destination so that TCP can transmit BGP messages based on the path MTU.

9.17 Configuring MP-BGP
Multiprotocol BGP (MP-BGP) enables BGP to support IPv4 unicast networks, IPv4 multicast networks, and IPv6 unicast networks.

9.18 Configuring the Dynamic BGP Peer Function

9.19 Maintaining BGP

9.20 Configuration Examples for BGP

9.21 FAQ About BGP

9.1 Overview of BGP

Definition

The Border Gateway Protocol (BGP) is a distance vector protocol that allows devices between Autonomous Systems (ASs) to communicate and selects optimal routes. BGP-1, BGP-2, and BGP-3 are three earlier versions of BGP. BGP-4 has been used since 1994. Since 2006, unicast IPv4 networks have been using BGP-4, and other networks (such as IPv6 networks) have been using **MP-BGP**.

MP-BGP is an extension of BGP-4 and applies to different networks; however, the original message exchange and routing mechanisms of BGP-4 are not changed. MP-BGP applications on IPv6 unicast and IPv4 multicast networks are called BGP4+ and Multicast BGP (MBGP) respectively.

Purpose

A network is divided into different ASs to facilitate the management over the network. In 1982, the Exterior Gateway Protocol (EGP) was used to dynamically exchange routing information between ASs. EGP advertises only reachable routes but not select optimal routes or prevent routing loops. Therefore, EGP cannot meet network management requirements.

BGP was designed to replace EGP. Different from EGP, BGP can select optimal routes, prevent routing loops, transmit routing information efficiently, and maintain a large number of routes.

Although BGP is used to transmit routing information between ASs, BGP is not the best choice in some scenarios. For example, on the egress connecting a data center to the Internet, static routing instead of BGP is used to prevent a huge number of Internet routes from affecting the internal network of the data center.

Benefits

BGP ensures high network security, flexibility, stability, reliability, and efficiency:

- BGP uses authentication and Generalized TTL Security Mechanism (GTSM) to ensure **network security**.
- BGP provides routing policies to allow for flexible **route selection** and **routing policy-based route advertisement**.
- BGP provides **9.2.8 Route Summarization** and **9.2.9 Route Dampening** to prevent route flapping and improve network stability.
- BGP uses the Transport Control Protocol (TCP) with port number 179 as the transport layer protocol and supports **9.2.10 BFD for BGP**, **9.2.11 BGP Tracking**, and **9.2.12 BGP GR and NSR** to improve network reliability.
- BGP uses the **9.2.14 Dynamic Update Peer-Groups** technology to send packets in groups when a large number of peers and routes exist and most peers share the same outbound policies, improving BGP forwarding performance.

9.2 Understanding BGP

9.2.1 Basic Concepts of BGP

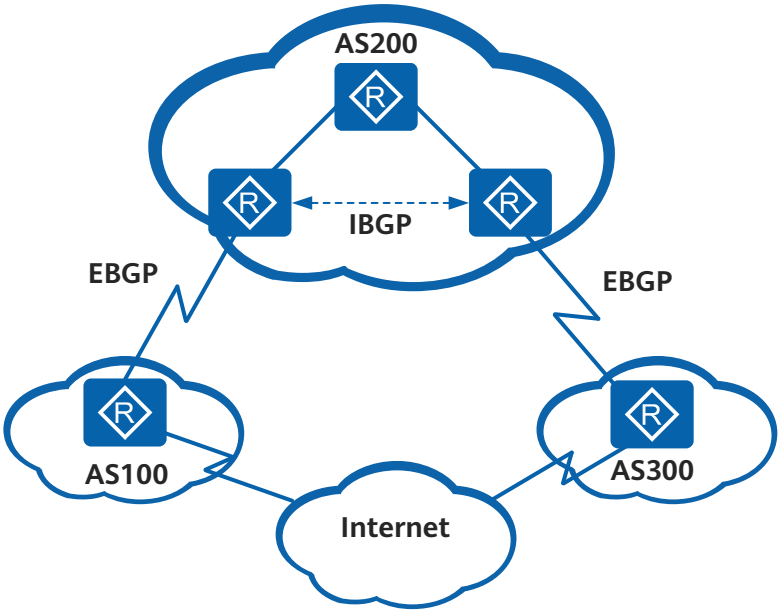
Autonomous System

An Autonomous System (AS) is a group of Internet Protocol (IP) networks that are controlled by one entity, typically an Internet service provider (ISP), and that have the same routing policy. Each AS is assigned a unique AS number, which identifies an AS on a BGP network. Two types of AS numbers are available: 2-byte AS numbers and 4-byte AS numbers. A 2-byte AS number ranges from 1 to 65535, and a 4-byte AS number ranges from 1 to 4294967295. Devices supporting 4-byte AS numbers are compatible with devices supporting 2-byte AS numbers.

BGP Neighbor Type

As shown in **Figure 9-1**, BGP neighbor type is classified into two types according to where it runs: External BGP (EBGP) and Internal BGP (IBGP).

Figure 9-1 BGP operating mode



- EBGP: runs between ASs. To prevent routing loops between ASs, a BGP device discards the routes with the local AS number when receiving the routes from EBGP peers.
- IBGP: runs within an AS. To prevent routing loops within an AS, a BGP device does not advertise the routes learned from an IBGP peer to the other IBGP peers and establishes full-mesh connections with all the IBGP peers. To address the problem of too many IBGP connections between IBGP peers, BGP uses **9.2.6 Route Reflector** and **9.2.7 BGP Confederation**.

  NOTE

  If a BGP device needs to advertise the route received from an EBGP peer outside an AS through another BGP device, IBGP is recommended.

Device Roles in BGP Message Exchange

There are two device roles in BGP message exchange:

- Speaker: The device that sends BGP messages is called a BGP speaker. The speaker receives and generates new routes, and advertises the routes to other BGP speakers.
- Peer: The speakers that exchange messages with each other are called BGP peers. A group of peers can form a peer group.

BGP Router ID

The BGP router ID is a 32-bit value that is often represented by an IPv4 address to identify a BGP device. It is carried in the Open message sent during the

establishment of a BGP session. When two BGP peers need to establish a BGP session, they each require a unique router ID. Otherwise, the two peers cannot establish a BGP session.

The BGP router ID of a device must be unique on a BGP network. It can be manually configured or selected from IPv4 addresses on the device. By default, an IPv4 address of a loopback interface on a device is used as the BGP router ID. If no loopback interface is configured on the device, the system selects the largest IPv4 address from all IPv4 addresses of interfaces as the BGP router ID. Once the BGP router ID is selected, the system retains this router ID even if a larger IPv4 address is configured on the device later. The system changes the BGP router ID only when the corresponding IPv4 address is deleted.

9.2.2 BGP Fundamentals

BGP peer establishment, update, and deletion involve five types of messages, six state machine states, and five route exchange rules.

BGP Messages

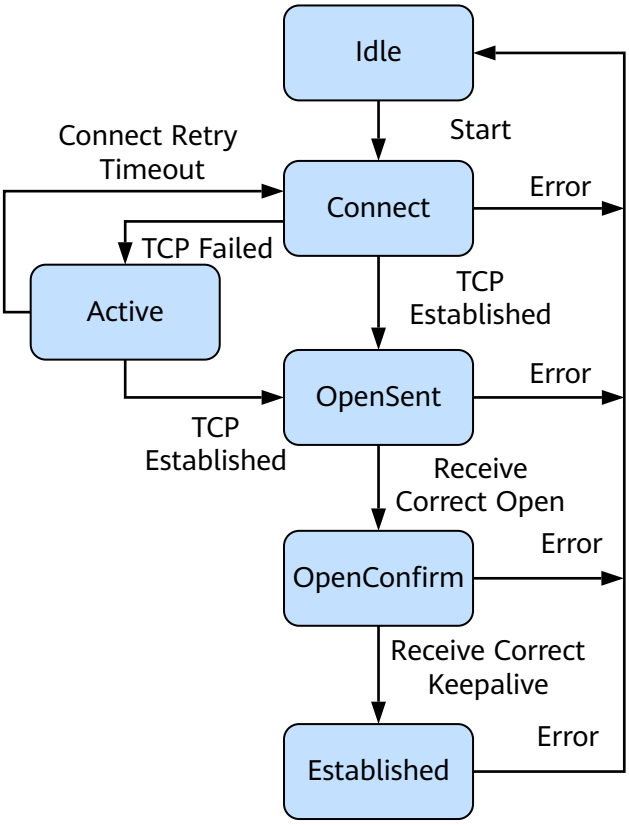
BGP peers exchange the following messages, among which Keepalive messages are periodically sent and other messages are triggered by events.

- Open message: is used to establish BGP peer relationships.
- Update message: is used to exchange routes between BGP peers.
- Notification message: is used to terminate BGP connections.
- Keepalive message: is used to maintain BGP connections.
- Route-refresh message: is used to request the peer to resend routes if routing policies are changed. Only the BGP devices supporting route-refresh can send and respond to Route-refresh messages.

BGP State Machine

As shown in **Figure 9-2**, a BGP device uses a finite state machine (FSM) to determine its operations with peers. The FSM has six states: Idle, Connect, Active, OpenSent, OpenConfirm, and Established. Three common states are involved in BGP peer establishment: Idle, Active, and Established.

Figure 9-2 BGP state machine



1. The Idle state is the initial BGP state. In Idle state, the BGP device refuses all connection requests from neighbors. The BGP device initiates a TCP connection with its BGP peer and changes its state to Connect only after receiving a Start event from the system.

   NOTE

   - The Start event occurs when an operator configures a BGP process or resets an existing BGP process, or when the router software resets a BGP process.
   - If an error occurs at any state of the FSM, for example, the BGP device receives a Notification message or TCP connection termination notification, the BGP device returns to the Idle state.

2. In Connect state, the BGP device starts the ConnectRetry timer and waits to establish a TCP connection.
   - If the TCP connection is established, the BGP device sends an Open message to the peer and changes to the OpenSent state.
   - If the TCP connection fails to be established, the BGP device moves to the Active state.
   - If the BGP device does not receive a response from the peer before the ConnectRetry timer expires, the BGP device attempts to establish a TCP connection with another peer and stays in Connect state.
3. In Active state, the BGP device keeps trying to establish a TCP connection with the peer.

- If the TCP connection is established, the BGP device sends an Open message to the peer, closes the ConnectRetry timer, and changes to the OpenSent state.
- If the TCP connection fails to be established, the BGP device stays in the Active state.
- If the BGP device does not receive a response from the peer before the ConnectRetry timer expires, the BGP device returns to the Connect state.

4. In OpenSent state, the BGP device waits an Open message from the peer and then checks the validity of the received Open message, including the AS number, version, and authentication password.
   - If the received Open message is valid, the BGP device sends a Keepalive message and changes to the OpenConfirm state.
   - If the received Open message is invalid, the BGP device sends a Notification message to the peer and returns to the Idle state.
5. In OpenConfirm state, the BGP device waits for a Keepalive or Notification message from the peer. If the BGP device receives a Keepalive message, it transitions to the Established state. If it receives a Notification message, it returns to the Idle state.
6. In Established state, the BGP device exchanges Update, Keepalive, Route-refresh, and Notification messages with the peer.
   - If the BGP device receives a valid Update or Keepalive message, it considers that the peer is working properly and maintains the BGP connection with the peer.
   - If the BGP device receives an invalid Update or Keepalive message, it sends a Notification message to the peer and returns to the Idle state.
   - If the BGP device receives a Route-refresh message, it does not change its status.
   - If the BGP device receives a Notification message, it returns to the Idle state.
   - If the BGP device receives a TCP connection termination notification, it terminates the TCP connection with the peer and returns to the Idle state.

Route Exchange Rules

A BGP device adds optimal routes to the BGP routing table to generate BGP routes. After establishing a BGP peer relationship with a neighbor, the BGP device follows the following rules to exchange routes with the peer:

- Advertises the BGP routes received from IBGP peers only to its EBGP peers.
- Advertises the BGP routes received from EBGP peers to its EBGP peers and IBGP peers.
- Advertises the optimal route to its peers when there are multiple valid routes to the same destination.
- Sends only updated BGP routes when BGP routes change.
- Accepts all the routes sent from its peers.

9.2.3 Interaction Between BGP and an IGP

BGP and IGPs use different routing tables. To enable different ASs to communicate, you need to configure interaction between BGP and IGPs so that BGP routes can be imported into IGP routing tables and IGP routes can also be imported to BGP routing tables.

Importing IGP Routes to BGP Routing Tables

BGP does not discover routes and so needs to import the routes discovered by IGPs to BGP routing tables so that different ASs can communicate. When an AS needs to advertise routes to another AS, an Autonomous System Boundary Router (ASBR) imports IGP routes to its BGP routing table. To better plan the network, you can use routing policies to filter routes and set route attributes when BGP imports IGP routes. Alternatively, you can set the multi-exit discriminator (MED) to help EBGP peers select the best path for traffic entering an AS.

BGP imports routes in either import or network mode:

- In import mode, BGP imports IGP routes, including RIP, OSPF, and IS-IS routes, into BGP routing tables based on protocol type. To ensure the validity of imported IGP routes, BGP can also import static routes and direct routes in import mode.
- In network mode, BGP imports the routes in the IP routing table one by one to BGP routing tables. The network mode is more accurate than the import mode.

Importing BGP Routes to IGP Routing Tables

When an AS needs to import routes from another AS, an ASBR imports BGP routes to its IGP routing table. To prevent a large number of BGP routes from affecting devices within the AS, IGPs can use routing policies to filter routes and set route attributes when importing BGP routes.

9.2.4 BGP Security

BGP uses authentication and Generalized TTL Security Mechanism (GTSM) to ensure exchange security between BGP peers.

BGP Authentication

BGP authentication includes Message Digest 5 (MD5) authentication and keychain authentication, which improves communication security between BGP peers. In MD5 authentication, you can only set the authentication password for a TCP connection. In keychain authentication, you can set the authentication password for a TCP connection and authenticate BGP messages.

BGP GTSM

BGP GTSM checks whether the time to live (TTL) value in the IP packet header is within a predefined range and permits or discards the packets of which the TTL values are out of the predefined range to protect services above the IP layer. BGP GTSM enhances system security.

Assume that the TTL value range of packets from BGP peers is set to 254-255. When an attacker forges valid BGP packets and keeps sending these packets to

attack a device, the TTL values of these packets are smaller than 254. If BGP GTSM is not enabled on the device, the device finds that these packets are destined for itself and sends the packets to the control plane for processing. Then the control layer needs to process a large number of such attack packets, causing high CPU usage. If BGP GTSM is enabled on the device, the system checks the TTL values in all BGP packets and discards the attack packets of which the TTL values are smaller than 254. This prevents network attack packets from consuming CPU resources.

9.2.5 BGP Route Selection Rules and Load Balancing

There may be multiple routes to the same destination in a BGP routing table. BGP will select one route as the optimal route and advertise it to peers. To select the optimal route among these routes, BGP compares the BGP attributes of the routes in sequence based on route selection rules.

BGP Attributes

Route attributes describe routes. BGP route attributes are classified into the following types. **Table 9-1** lists common BGP attributes.

- Well-known mandatory attribute

  All BGP devices can identify this type of attributes, which must be carried in Update messages. Without this type of attributes, errors occur in routing information.
- Well-known discretionary attribute

  All BGP devices can identify this type of attributes, which are optional in Update messages. Without this type of attributes, errors do not occur in routing information.
- Optional transitive attribute

  BGP devices may not identify this type of attributes but still accepts them and advertises them to peers.
- Optional non-transitive attribute

  BGP devices may not identify this type of attributes. If a BGP device does not identify this type of attributes, it ignores them and does not advertise them to peers.

Table 9-1 Common BGP attributes

Attribute	Type
Origin	Well-known mandatory
AS_Path	Well-known mandatory
Next_Hop	Well-known mandatory
Local_Pref	Well-known discretionary
Community	Optional transitive
MED	Optional non-transitive

Attribute	Type
Originator_ID	Optional non-transitive
Cluster_List	Optional non-transitive

The following describes common BGP route attributes:

- **Origin**

  The Origin attribute defines the origin of a route and marks the path of a BGP route. The Origin attribute is classified into three types:

  – IGP

    A route with IGP as the Origin attribute is of the highest priority. The Origin attribute of the routes imported into a BGP routing table using the **network** command is IGP.

  – EGP

    A route with EGP as the Origin attribute is of the secondary highest priority. The Origin attribute of the routes obtained through EGP is EGP.

  – Incomplete

    A route with Incomplete as the Origin attribute is of the lowest priority. The Origin attribute of the routes learned by other means is Incomplete. For example, the Origin attribute of the routes imported by BGP using the **import-route** command is Incomplete.

- **AS_Path**

  The AS_Path attribute records all the ASs that a route passes through from the source to the destination in the vector order. To prevent inter-AS routing loops, a BGP device does not receive the routes of which the AS_Path list contains the local AS number.

  When a BGP speaker advertises an imported route:

  – If the route is advertised to EBGP peers, the BGP speaker creates an AS_Path list containing the local AS number in an Update message.

  – If the route is advertised to IBGP peers, the BGP speaker creates an empty AS_Path list in an Update message.

  When a BGP speaker advertises a route learned in the Update message sent by another BGP speaker:

  – If the route is advertised to EBGP peers, the BGP speaker adds the local AS number to the leftmost of the AS_Path list. According to the AS_Path list, the BGP speaker that receives the route can learn about the ASs through which the route passes to reach the destination. The number of the AS that is nearest to the local AS is placed on the top of the AS_Path list. The other AS numbers are listed according to the sequence in which the route passes through ASs.

  – If the route is advertised to IBGP peers, the BGP speaker does not change the AS_Path attribute of the route.

- **Next_Hop**

  The Next_Hop attribute records the next hop that a route passes through. The Next_Hop attribute of BGP is different from that of an IGP because it may not

be the neighbor IP address. A BGP speaker processes the Next_Hop attribute based on the following rules:

- When advertising a route to an EBGP peer, a BGP speaker sets the Next_Hop attribute of the route to the address of the local interface through which the BGP peer relationship is established with the peer.
- When advertising a locally originated route to an IBGP peer, the BGP speaker sets the Next_Hop attribute of the route to the address of the local interface through which the BGP peer relationship is established with the peer.
- When advertising a route learned from an EBGP peer to an IBGP peer, the BGP speaker does not change the Next_Hop attribute of the route.

- **Local_Pref**

  The Local_Pref attribute indicates the BGP preference of a device and helps determine the optimal route when traffic 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 the route with the highest Local_Pref. The Local_Pref attribute is exchanged only between IBGP peers and is not advertised to other ASs. The Local_Pref attribute can be manually configured. If no Local_Pref attribute is configured for a route, the Local_Pref attribute of the route uses the default value 100.

- **MED**

  The multi-exit discriminator (MED) attribute helps determine the optimal route when traffic enters an AS. When a BGP device obtains multiple routes to the same destination address but with different next hops from EBGP peers, the BGP device selects the route with the smallest MED value as the optimal route.

  The MED attribute is exchanged only between two neighboring ASs. The AS that receives the MED attribute does not advertise it to any other ASs. The MED attribute can be manually configured. If no MED attribute is configured for a route, the MED attribute of the route uses the default value 0.

- **Community**

  The Community attribute identifies the BGP routes with the same characteristics, simplifies the applications of routing policies, and facilitates route maintenance and management.

  The Community attribute includes self-defined community attributes and well-known community attributes. **Table 9-2** lists well-known community attributes.

Table 9-2 Well-known community attributes

Community Attribute	Value	Description
Internet	0 (0x00000000)	A BGP device can advertise the received route with the Internet attribute to all peers.
No_Advertise	4294967042 (0xFFFFFF02)	A BGP device does not advertise the received route with the No_Advertise attribute to any peer.

Community Attribute	Value	Description
No_Export	4294967041 (0xFFFFFF01)	A BGP device does not advertise the received route with the No_Export attribute to devices outside the local AS.
No_Export_Subconfed	4294967043 (0xFFFFFF03)	A BGP device does not advertise the received route with the No_Export_Subconfed attribute to devices outside the local AS or to devices outside the local sub-AS.

- **Originator_ID and Cluster_List**

  The Originator_ID attribute and Cluster_List attribute help eliminate loops in route reflector scenarios. For details, see **9.2.6 Route Reflector**.

BGP Route Selection Policies

When there are multiple routes to the same destination, BGP compares the following attributes in sequence to select the optimal route:

1. Prefers the route with the largest PrefVal value.

   The PrefVal attribute is a Huawei proprietary attribute and is valid only on the device where it is configured.
2. Prefers the route with the highest Local_Pref.

   If a route does not have the Local_Pref attribute, the Local_Pref attribute of the route uses the default value 100.
3. Prefers the manually summarized route, automatically summarized route, route imported using the **network** command, route imported using the **import-route** command, and route learned from peers. These routes are in descending order of priority.
4. Prefers the route with the shortest AS_Path.
5. Prefers the route with the lowest origin type. IGP is lower than EGP, and EGP is lower than Incomplete.
6. Prefers the route with the lowest MED if routes are received from the same AS.
7. Prefers EBGP routes, IBGP routes, LocalCross routes, and RemoteCross routes, which are listed in descending order of priority.

   LocalCross allows a PE to add the VPNv4 route of a VPN instance to the routing table of the VPN instance if the export RT of the VPNv4 route matches the import RT of another VPN instance on the PE. RemoteCross allows a local PE to add the VPNv4 route learned from a remote PE to the routing table of a VPN instance on this local PE if the export RT of the VPNv4 route matches the import RT of the VPN instance.
8. Prefers the route with the lowest IGP metric to the BGP next hop.

NOTE

If there are multiple routes to the same destination, an IGP calculates the route metric using its routing algorithm.

9. Prefers the route with the shortest Cluster_List.
10. Prefers the route advertised by the device with the smallest router ID.

    If a route carries the Originator_ID attribute, BGP prefers the route with the smallest Originator_ID without comparing the router ID.
11. Prefers the route learned from the peer with the lowest IP address.

BGP Load Balancing

When there are multiple equal-cost routes to the same destination, you can perform load balancing among these routes to load balance traffic. Equal-cost BGP routes can be generated for traffic load balancing only when the first eight route attributes described in "BGP Route Selection Policies" are the same.

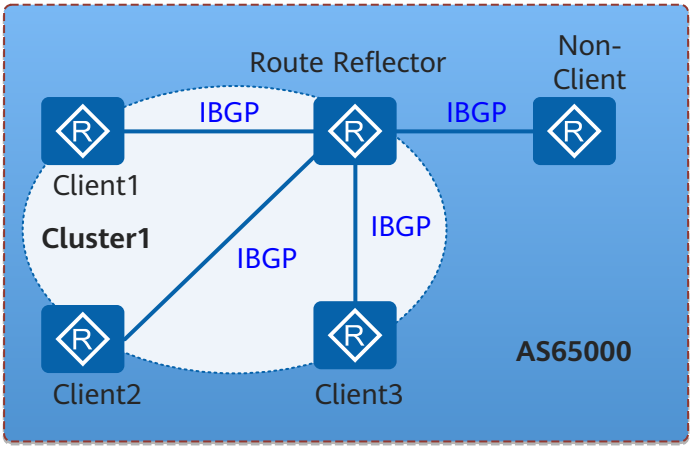
9.2.6 Route Reflector

To ensure connectivity between IBGP peers, you need to establish full-mesh connections between IBGP peers. If there are n devices in an AS, n(n-1)/2 IBGP connections need to be established. When there are a large number of devices, many network resources and CPU resources are consumed. A route reflector (RR) can be used between IBGP peers to solve this problem.

Roles in RR

As shown in **Figure 9-3**, the following roles are involved in RR scenarios in an AS.

Figure 9-3 Networking diagram of the RR



- Route reflector (RR): a BGP device that can reflect the routes learned from an IBGP peer to other IBGP peers. An RR is similar to a designated router (DR) on an OSPF network.
- Client: an IBGP device of which routes are reflected by the RR to other IBGP devices. In an AS, clients only need to directly connect to the RR.

- Non-client: an IBGP device that is neither an RR nor a client. In an AS, a non-client must establish full-mesh connections with the RR and all the other non-clients.
- Originator: is a device that originates routes in an AS. The Originator_ID attribute helps eliminate routing loops in a cluster.
- Cluster: is a set of the RR and clients. The Cluster_List attribute helps eliminate routing loops between clusters.

RR Principles

Clients in a cluster only need to exchange routing information with the RR in the same cluster. Therefore, clients only need to establish IBGP connections with the RR. This reduces the number of IBGP connections in the cluster. As shown in **Figure 9-3**, in AS 65000, Cluster1 is comprised of an RR and three clients. The number of IBGP connections in AS 65000 is then reduced from 10 to 4, which simplifies the device configuration and reduces the loads on the network and CPU.

The RR allows a BGP device to advertise the BGP routes learned from an IBGP peer to other IBGP peers, and uses the Cluster_List and Originator_ID attributes to eliminate routing loops. The RR advertises routes to IBGP peers based on the following rules:

- The RR advertises the routes learned from a non-client to all the clients.
- The RR advertises the routes learned from a client to all the other clients and all the non-clients.
- The RR advertises the routes learned from an EBGP peer to all the clients and non-clients.

Cluster_List Attribute

An RR and its clients form a cluster, which is identified by a unique cluster ID in an AS. To prevent routing loops between clusters, an RR uses the Cluster_List attribute to record the cluster IDs of all the clusters that a route passes through.

- When a route is reflected by an RR for the first time, the RR adds the local cluster ID to the top of the cluster list. If there is no cluster list, the RR creates a Cluster_List attribute.
- When receiving an updated route, the RR checks the cluster list of the route. If the cluster list contains the local cluster ID, the RR discards the route. If the cluster list does not contain the local cluster ID, the RR adds the local cluster ID to the cluster list and then reflects the route.

Originator_ID Attribute

The originator ID identifies the originator of a route and is generated by an RR to prevent routing loops in a cluster. Its value is the same as the router ID.

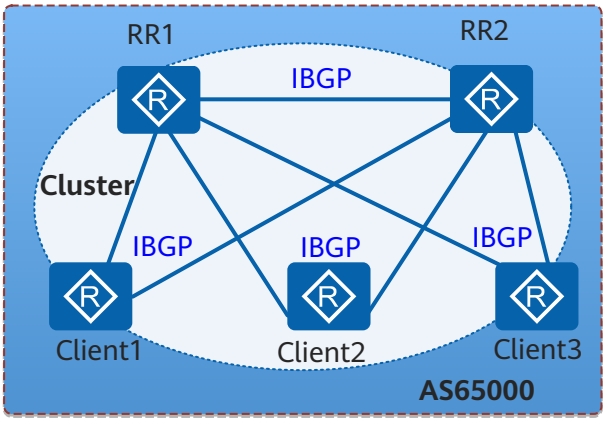
- When a route is reflected by an RR for the first time, the RR adds the Originator_ID attribute to this route. The Originator_ID attribute identifies the originator of the route. If the route contains the Originator_ID attribute, the RR retains this Originator_ID attribute.

- When a device receives a route, the device compares the originator ID of the route with the local router ID. If they are the same, the device discards the route.

Backup RR

To ensure network reliability and prevent single points of failures, redundant RRs are required in a cluster. An RR allows a BGP device to advertise the routes received from an IBGP peer to other IBGP peers. Therefore, routing loops may occur between RRs in the same cluster. To solve this problem, all the RRs in the cluster must use the same cluster ID.

Figure 9-4 Backup RR



As shown in **Figure 9-4**, RR1 and RR2 reside in the same cluster and have the same cluster ID configured.

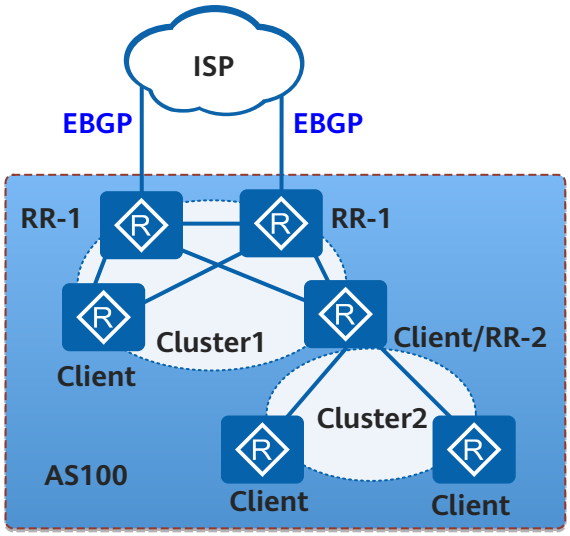
- When Client1 receives an updated route from an EBGP peer, Client1 advertises this route to RR1 and RR2 using IBGP.
- After RR1 and RR2 receive this route, they add the local cluster ID to the top of the cluster list of the route and then reflect the route to other clients (Client2 and Client3) and to each other.
- After RR1 and RR2 receive the reflected route from each other, they check the cluster list of the route, finding that the cluster list contains their local cluster IDs. RR1 and RR2 discard this route to prevent routing loops.

RRs of Multiple Clusters in an AS

There may be multiple clusters in an AS. RRs of the clusters establish IBGP peer relationships. When RRs reside at different network layers, an RR at the lower network layer can be configured as a client to implement hierarchical RR. When RRs reside at the same network layer, RRs of different clusters can establish full-mesh connections to implement flat RR.

Hierarchical RR

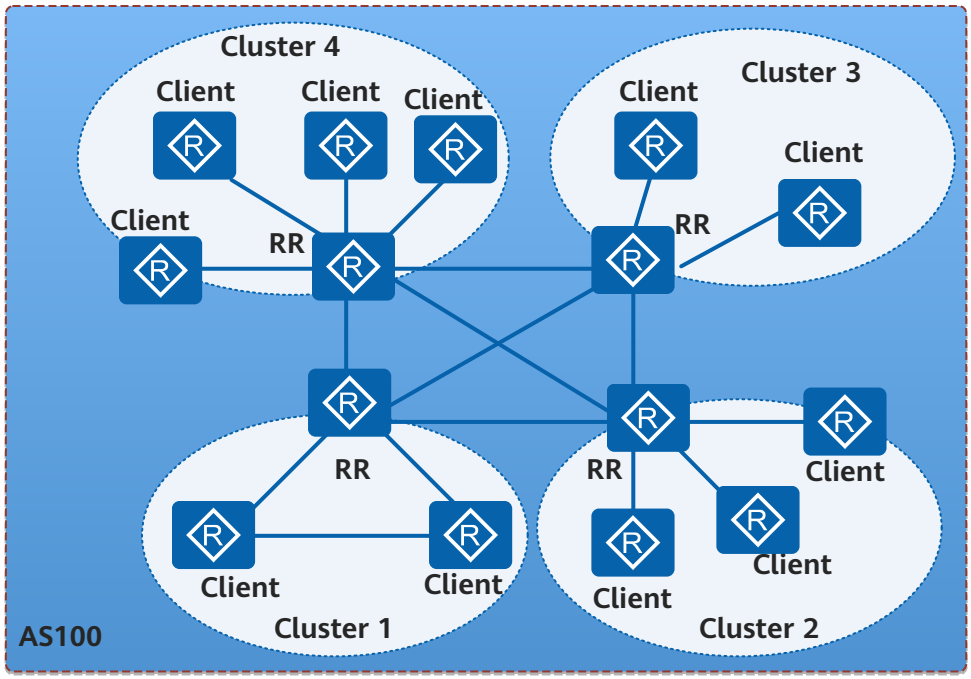
Figure 9-5 Hierarchical RR



In practice, hierarchical RR is often used. As shown in **Figure 9-5**, the ISP provides Internet routes to AS 100. AS 100 is divided into two clusters, Cluster1 and Cluster2. Four devices in Cluster1 are core routers and use a backup RR to ensure reliability.

Flat RR

Figure 9-6 Flat RR



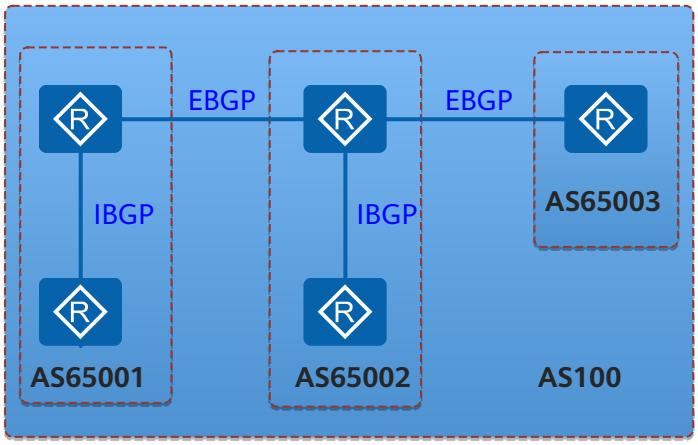
As shown in **Figure 9-6**, the backbone network is divided into multiple clusters. RRs of the clusters are non-clients and establish full-mesh connections with each

other. Although each client only establishes an IBGP connection with its RR, all the RRs and clients can receive all routing information.

9.2.7 BGP Confederation

In addition to a route reflector, the confederation is another method that reduces the number of IBGP connections in an AS. A confederation divides an AS into sub-ASs. Full-mesh IBGP connections are established in each sub-AS. EBGP connections are established between sub-ASs. ASs outside a confederation still consider the confederation as an AS. After a confederation divides an AS into sub-ASs, it assigns a confederation ID (the AS number) to each router within the AS. This brings two benefits. First, original IBGP attributes are retained, including the Local_Pref attribute, MED attribute, and Next_Hop attribute. Second, confederation-related attributes are automatically deleted when being advertised outside a confederation. Therefore, the administrator does not need to configure the rules for filtering information such as sub-AS numbers at the egress of a confederation.

Figure 9-7 Networking diagram of a confederation



As shown in **Figure 9-7**, AS 100 is divided into three sub-ASs after a confederation is configured: AS65001, AS65002, and AS65003. The AS number AS 100 is used as the confederation ID. The number of IBGP connections in AS 100 is then reduced from 10 to 4, which simplifies the device configuration and reduces the loads on the network and CPU. In addition, BGP devices outside AS 100 only know the existence of AS 100 but not the confederation within AS 100. Therefore, the confederation does not increase the CPU load.

Comparisons Between a Route Reflector and a Confederation

Table 9-3 compares a route reflector and a confederation in terms of the configuration, device connection, and applications.

Table 9-3 Comparisons between a route reflector and a confederation

Route Reflector	Confederation
Retains the existing network topology and ensures compatibility.	Requires the logical topology to be changed.
Requires only a route reflector to be configured because clients do not need to know that they are clients of a route reflector.	Requires all devices to be reconfigured.
Requires full-mesh connections between clusters.	Does not require full-mesh connections between sub-ASs of a confederation because the sub-ASs are special EBGP peers.
Applies to medium and large networks.	Applies to large networks.

9.2.8 Route Summarization

The BGP routing table of each device on a large network is large. This burdens devices, increases the route flapping probability, and affects network stability.

Route summarization is a mechanism that combines multiple routes into one route. This mechanism allows a BGP device to advertise only the summarized route but not all the specific routes to peers, therefore reducing the size of the BGP routing table. If the summarized route flaps, the network is not affected, so network stability is improved.

BGP supports automatic summarization and manual summarization on IPv4 networks, and supports only manual summarization on IPv6 networks.

- Automatic summarization: summarizes the routes imported by BGP. After automatic summarization is configured, BGP summarizes routes based on the natural network segment and advertises only the summarized route to peers. For example, BGP summarizes 10.1.1.1/24 and 10.2.1.1/24 (two Class A addresses with non-natural mask) into 10.0.0.0/8 (Class A address with natural mask).
- Manual summarization: summarizes routes in the local BGP routing table. Manual summarization can help control the attributes of the summarized route and determine whether to advertise specific routes.

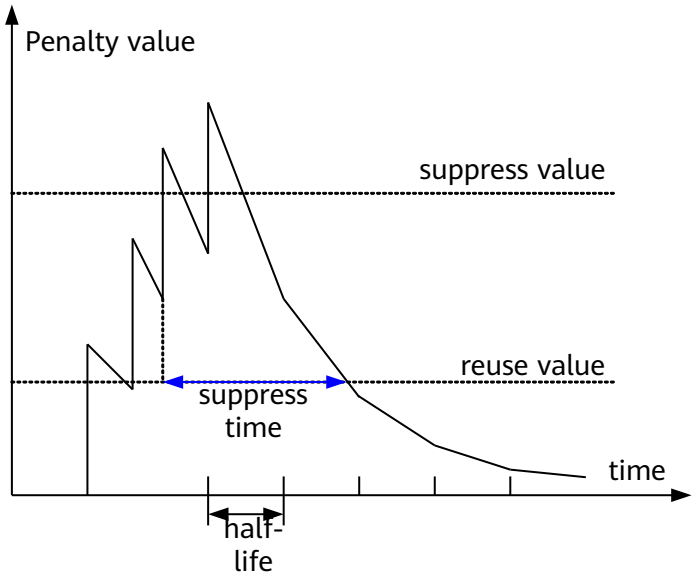
To prevent routing loops caused by route summarization, BGP uses the AS_Set attribute. The AS_Set attribute is an unordered set of all ASs that a route passes through. When the summarized route enters an AS in the AS_Set attribute again, BGP finds that the local AS number has been recorded in the AS_Set attribute of the route and discards this route to prevent a routing loop.

9.2.9 Route Dampening

When BGP is used on complex networks, route flapping occurs frequently. To prevent frequent route flapping, BGP uses route dampening to suppress unstable routes.

Route flapping is a process of adding a route to an IP routing table and then withdrawing this route. When route flapping occurs, a BGP device sends an Update message to its neighbors. The devices that receive the Update message need to recalculate routes and modify routing tables. Frequent route flapping consumes lots of bandwidths and CPU resources and even affects normal network operation.

Figure 9-8 Diagram of BGP route dampening



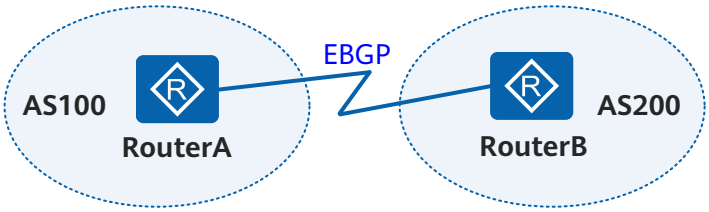
Route dampening measures the stability of a route using a penalty value. A larger penalty value indicates a less stable route. As shown in **Figure 9-8**, each time route flapping occurs, BGP increases the penalty of this route by a value of 1000. When the penalty value of a route exceeds the suppression threshold, BGP suppresses this route, and does not add it to the IP routing table or advertise any Update message to peers. After a route is suppressed for a period of time (half life), the penalty value is reduced by half. When the penalty value of a route decreases to the reuse threshold, the route is reusable and is added to the routing table. At the same time, BGP advertises an Update message to peers. The suppression time is the period from when a route is suppressed to when the route is reusable.

Route dampening applies only to EBGP routes but not IBGP routes. IBGP routes may include the routes of the local AS, and an IGP network requires that the routing tables of devices within an AS be the same. If IBGP routes were dampened, routing tables on devices are inconsistent when these devices have different dampening parameters. Therefore, route dampening does not apply to IBGP routes.

9.2.10 BFD for BGP

BGP periodically sends messages to peers to detect the status of the peers. It takes more than 1 second for this detection mechanism to detect a fault. When data is transmitted at gigabit rates, long-time fault detection will cause packet loss. This cannot meet high reliability requirements of networks. Bidirectional Forwarding Detection (BFD) provides the millisecond-level fault detection for BGP to improve network reliability.

Figure 9-9 Networking diagram of BFD for BGP



As shown in **Figure 9-9**, RouterA belongs to AS 100 and RouterB belongs to AS 200. RouterA and RouterB are directly connected and establish the EBGP peer relationship. Association between BGP and BFD is configured on RouterA and RouterB. When a fault occurs on the link between RouterA and RouterB, BFD can rapidly detect that the BFD session changes from Up to Down and notify this fault to RouterA and RouterB. RouterA and RouterB process the neighbor Down event and select routes again using BGP.

9.2.11 BGP Tracking

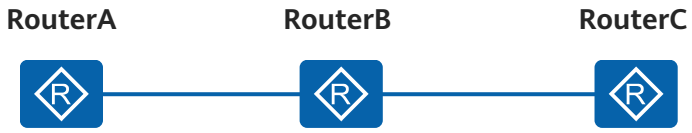
BGP tracking provides fast link fault detection to speed up network convergence. When a fault occurs on the link between BGP peers that have BGP tracking configured, BGP tracking can quickly detect peer unreachability and instruct the routing management module to notify BGP of the fault, implementing rapid network convergence.

Compared to BFD, BGP tracking is easy to configure because it needs to be configured only on the local device. BGP tracking is a fault detection mechanism at the routing layer, whereas BFD is a fault detection mechanism at the link layer. BGP route convergence on a network where BGP tracking is configured is slower than that on a network where BFD is configured. Therefore, BGP tracking cannot meet the requirements of voice services that require fast convergence.

Applications

As shown in **Figure 9-10**, RouterA and RouterB, and RouterB and RouterC establish IGP connections. RouterA and RouterC establish an IBGP peer relationship. BGP tracking is configured on RouterA. When a fault occurs on the link between RouterA and RouterB, IGP performs fast convergence. Subsequently, BGP tracking detects the unreachability of the route to RouterC and notifies the fault to BGP on RouterA, which then interrupts the BGP connection with RouterC.

Figure 9-10 Networking diagram of BGP tracking



NOTE

If establishing an IBGP peer relationship requires IGP routes, the interval between peer unreachability discovery and connection interruption needs to be configured, and this interval must be longer than the IGP route convergence time. Otherwise, the BGP peer relationship may have been interrupted before IGP route flapping caused by transient interruption is suppressed, causing unnecessary BGP convergence.

9.2.12 BGP GR and NSR

BGP graceful restart (GR) and non-stop routing (NSR) are high availability solutions that minimize the impact of device failures on user services.

BGP GR

BGP GR ensures that the forwarding plane continues to guide data forwarding during a device restart or active/standby switchover. The operations on the control plane, such as reestablishing peer relationships and performing route calculation, do not affect the forwarding plane. This mechanism prevents service interruptions caused by route flapping and improves network reliability.

GR concepts are as follows:

- GR restarter: is the device that is restarted by the administrator or triggered by failures to perform GR.
- GR helper: is the neighbor that helps the GR restarter to perform GR.
- GR time: is the time during which the GR helper retains forwarding information after detecting the restart or active/standby switchover of the GR restarter.

NOTE

In practical application, in order to realize that business forwarding is not affected by motherboard failure, it is usually possible to configure BGP GR in the hardware environment of dual motherboard to make sense.

All the models support the GR Helper, and only AR3200 series support the GR Restarter.

BGP GR process is as follows:

1. Using the BGP capability negotiation mechanism, the GR restarter and helper know each other's GR capability and establish a GR session.
2. When detecting the restart or active/standby switchover of the GR restarter, the GR helper does not delete the routing information and forwarding entries of the GR restarter or notify other neighbors of the restart or switchover, but waits to reestablish a BGP connection with the GR restarter.
3. The GR restarter reestablishes neighbor relationships with all GR helpers before the GR time expires.

BGP NSR

NSR is a reliability technique that prevents neighbors from detecting the control plane switchover. It applies to the devices that have the active and standby MPUs configured. Compared to GR, NSR does not require the help of neighbors and does not need to deal with interoperability issues. For details about NSR, see "NSR" in the *Feature Description - Reliability*.

NOTE

Only the AR3200 series support NSR.

Comparisons Between Active/Standby Switchovers with and Without GR and NSR

Table 9-4 Comparisons between active/standby switchovers with and without GR and NSR

Active/Standby Switchover Without GR and NSR	Active/Standby Switchover in GR Mode	Active/Standby Switchover in NSR Mode
The BGP peer relationship is reestablished.	The BGP peer relationship is reestablished.	The BGP peer relationship is reestablished.
Routes are recalculated.	Routes are recalculated.	Routes are recalculated.
The forwarding table changes.	The forwarding table remains unchanged.	The forwarding table remains unchanged.
Traffic is lost during forwarding, and services are interrupted.	No traffic is lost during forwarding, and services are not affected.	No traffic is lost during forwarding, and services are not affected.
The network detects route changes, and route flapping occurs for a short period of time.	Except the neighbors of the device where the active/standby switchover occurs, other routers do not detect route changes.	The network does not detect route changes.
-	The GR restarter requires neighbors to support the GR helper function. The GR helper function does not allow multiple neighbors to perform active/standby switchovers in GR mode simultaneously.	Neighbors do not need to support the NSR function.

9.2.13 BGP ORF

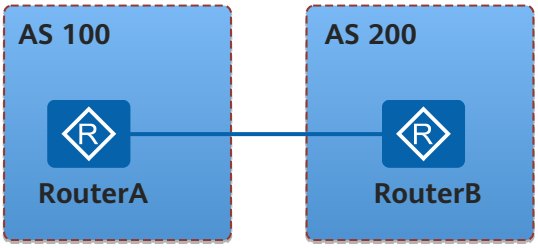
The prefix-based BGP outbound route filtering (ORF) capability to advertise required BGP routes. BGP ORF allows a device to send prefix-based import policies in a Route-refresh message to BGP peers. BGP peers construct export policies based on these import policies to filter routes before sending these routes, which has the following advantages:

- Prevents the local device from receiving a large number of unnecessary routes.
- Reduces CPU usage of the local device.
- Simplifies the configuration of BGP peers.
- Improves link bandwidth efficiency.

Applications

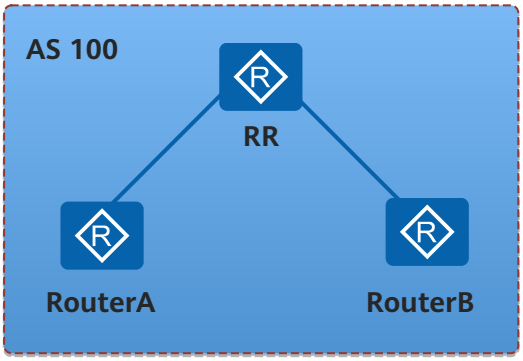
BGP ORF applies to the scenario when a device wants BGP peers to send only required routes, and BGP peers do not want to maintain different export policies for different devices.

Figure 9-11 Inter-AS EBGP peers



As shown in **Figure 9-11**, after negotiating the prefix-based ORF capability with RouterB, RouterA adds the local prefix-based import policies to a Route-refresh message and sends the message to RouterB. RouterB constructs export policies based on the received Route-refresh message and sends required routes to RouterA using a Route-refresh message. RouterA receives only required routes, and RouterB does not need to maintain routing policies. This reduces the configuration workload.

Figure 9-12 Intra-AS route reflector



As shown in **Figure 9-12**, there is a route reflector (RR) in AS 100. RouterA and RouterB are the clients of the RR. RouterA, RouterB, and the RR negotiate the prefix-based ORF capability. RouterA and RouterB then add the local prefix-based import policies to Route-refresh messages and send the messages to the RR. The RR constructs export policies based on the received import policies and reflects required routes in Route-refresh messages to RouterA and RouterB. RouterA and RouterB receive only required routes, and the RR does not need to maintain routing policies. This reduces the configuration workload.

9.2.14 Dynamic Update Peer-Groups

Currently, the rapid growth in the size of the routing table and the complexity of the network topology require BGP to support more peers. Especially in the case of a large number of peers and routes, high-performance grouping and forwarding are required when a router needs to send routes to a large number of BGP peers, most of which share the same outbound policies.

The dynamic update peer-groups feature treats all the BGP peers with the same outbound policies as an update-group. In this case, routes are grouped uniformly and then sent separately. That is, each route to be sent is grouped once and then sent to all peers in the update-group, improving grouping efficiency exponentially. For example, a route reflector (RR) has 100 clients and needs to reflect 100,000 routes to these clients. If the RR sends the routes grouped per peer to 100 clients, the total number of times that all routes are grouped is 10,000,000 (100,000 x 100). After the dynamic update peer-groups feature is used, the total number of grouping times changes to 100,000 (100,000 x 1), improving grouping performance by a factor of 100.

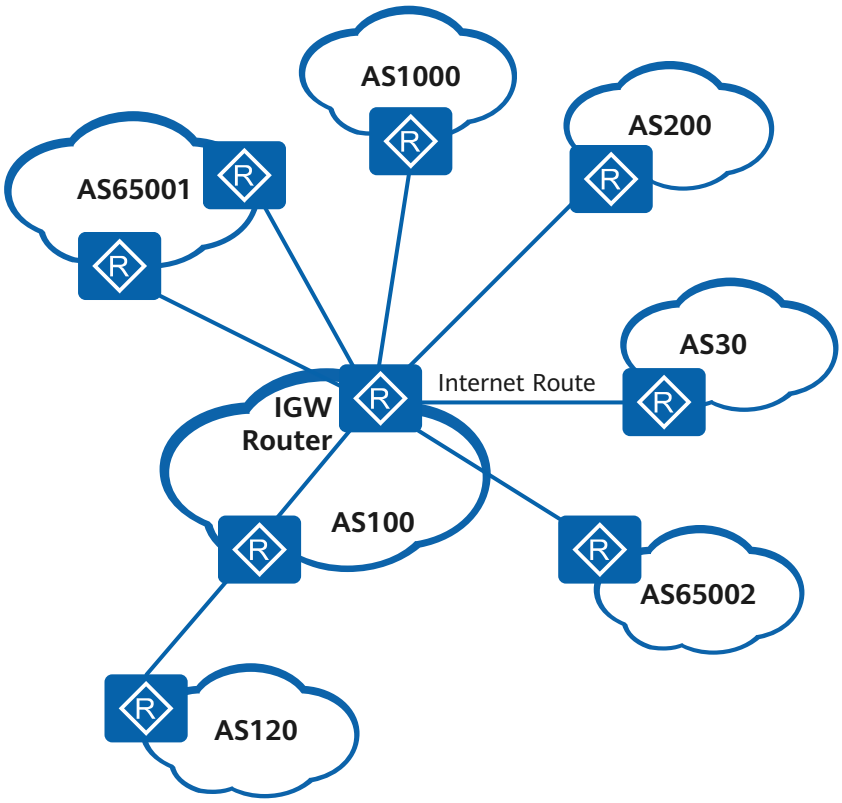
Applications

BGP uses the dynamic update peer-groups technology when a large number of peers and routes exist and most peers share the same outbound policies, improving BGP route grouping and forwarding performance. The dynamic update peer-groups feature applies to the following scenarios:

- International gateway

  As shown in **Figure 9-13**, the Internet gateway (IGW) router sends routes to all neighboring ASs. If the IGW router supports the dynamic update peer-groups feature, its BGP route forwarding performance will be greatly improved.

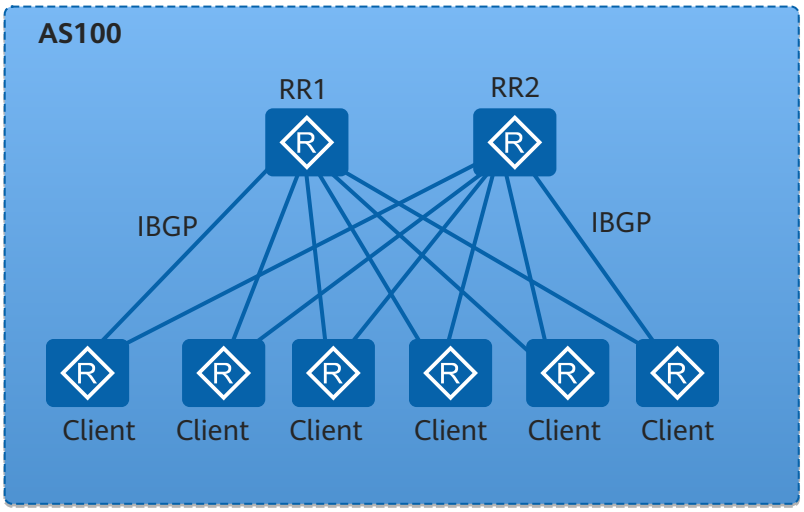
Figure 9-13 Networking diagram of the international gateway



- RR

  As shown in **Figure 9-14**, RRs send routes to all clients. If the RRs support the dynamic update peer-groups feature, their BGP route forwarding performance will be greatly improved.

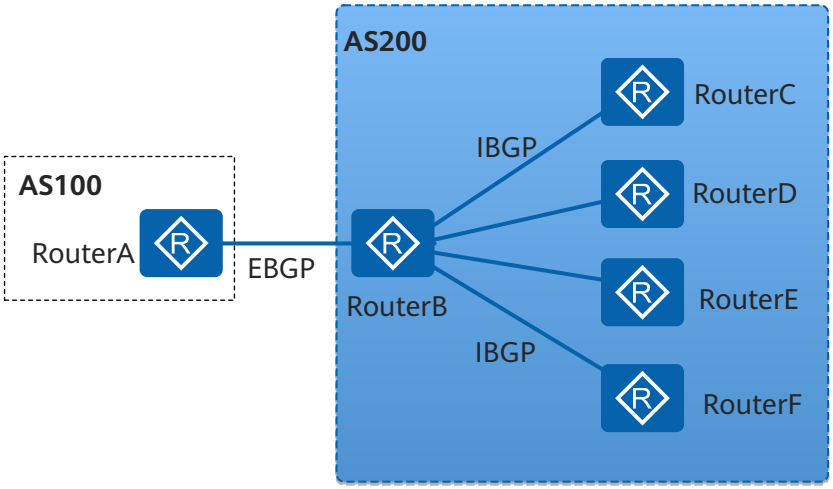
Figure 9-14 Networking diagram of RRs



- ASBR

  As shown in **Figure 9-15**, RouterB, as an Autonomous System Boundary Router (ASBR), sends all the routes received from an EBGP neighbor RouterA to all IBGP neighbors. If RouterB supports the dynamic update peer-groups feature, its BGP route forwarding performance will be greatly improved.

  Figure 9-15 Networking diagram of a PE connecting to multiple IBGP neighbors



9.2.15 MP-BGP

Traditional BGP-4 manages only IPv4 routing information. Inter-AS transmission of other network layer protocol packets (such as IPv6 and multicast packets) is limited. To support multiple network layer protocols, Multiprotocol BGP (MP-BGP) is designed in RFC 4760 as an extension to BGP-4. MP-BGP uses extended attributes and address families to support IPv6, multicast, and VPN, without changing the existing BGP packet forwarding and routing mechanism.

MP-BGP is called BGP4+ on IPv6 unicast networks or called multicast BGP (MBGP) on IPv4 multicast networks. MP-BGP establishes separate topologies for IPv6 unicast networks and IPv4 multicast networks, and stores IPv6 unicast and IPv4 multicast routing information in different routing tables. This ensures that routing information of IPv6 unicast networks and IPv4 multicast networks is separated from each other, and allows routes of different networks to be maintained using different routing policies.

Extended Attributes

In BGP, an Update message carries three IPv4-related attributes: NLRI, Next_Hop, and Aggregator.

To support multiple network layer protocols, BGP requires NLRI and Next_Hop attributes to carry information about network layer protocols. Therefore, MP-BGP uses the following new optional non-transitive attributes:

- MP_REACH_NLRI: indicates the multiprotocol reachable NLRI. It is used to advertise reachable routes and next hop information.
- MP_UNREACH_NLRI: indicates the multiprotocol unreachable NLRI. It is used to withdraw unreachable routes.

Address Families

MP-BGP uses address families to differentiate network layer protocols. Currently, devices support the following address family views:

- BGP-IPv4 unicast address family view
- BGP-IPv4 multicast address family view
- BGP-VPN instance IPv4 address family view
- BGP-VPNv4 address family view
- BGP-IPv6 unicast address family view
- BGP-VPN instance IPv6 address family view

9.2.16 BGP 6PE

Background

As IPv6 technology becomes more popular, an increasing number of separate IPv6 networks take shape. IPv6 provider edge (6PE), a technology designed to provide IPv6 services over IPv4 networks, allows service providers to provide IPv6 services without constructing IPv6 backbone networks. The 6PE solution connects separate IPv6 networks using multiprotocol label switching (MPLS) tunnels. The 6PE solution implements IPv4/IPv6 dual stack on the provider edge devices (PEs) of Internet service providers and uses the Multi-protocol Extensions for Border Gateway Protocol (MP-BGP) to assign labels to IPv6 routes. In this manner, the 6PE solution connects separate IPv6 networks over IPv4 tunnels between PEs.

Related Concepts

In practical application, different metropolitan area networks (MANs) of a service provider or collaborative backbone networks of different service providers often span multiple autonomous systems (ASs). The 6PE solution can be intra-AS 6PE or inter-AS 6PE, depending on whether separate IPv6 networks connect to the same AS. RFC defines three inter-AS 6PE modes: inter-AS 6PE OptionB with autonomous system boundary routers (ASBRs) as PEs, inter-AS 6PE OptionB, and inter-AS OptionC. This section describes the following 6PE modes:

- Intra-AS 6PE: Separate IPv6 networks are connected by the same AS. PEs in the AS exchange IPv6 routes by establishing MP-IBGP peer relationships.
- Inter-AS 6PE OptionB (with ASBRs as PEs): ASBRs in different ASs exchange IPv6 routes using MP-EBGP.
- Inter-AS 6PE OptionB: ASBRs in different ASs exchange labeled IPv6 routes by establishing MP-EBGP peer relationships.
- Inter-AS 6PE OptionC: PEs in different ASs exchange labeled IPv6 routes over multi-hop MP-EBGP peer sessions.

Intra-AS 6PE

Figure 9-16 shows intra-AS 6PE networking. 6PE runs on the edge of a service provider network. PEs that connect to IPv6 networks are IPv4/IPv6 dual-stack devices. PEs and customer edge devices (CEs) exchange IPv6 routes using the IPv6 Interior Gateway Protocol (IGP), or IPv6 External Border Gateway Protocol (EBGP). PEs exchange IPv4 routes with each other or with provider devices (Ps) using an IPv4 routing protocol. PEs must establish tunnels to transparently transmit IPv6 packets. PEs often use MPLS label switched paths (LSPs) and MPLS Local IFNET tunnels. By default, a PE uses an MPLS LSP to transmit IPv6 packets. If no MPLS LSP is available, a PE uses an MPLS Local IFNET tunnel to transmit IPv6 packets.

Figure 9-16 Intra-AS 6PE networking diagram

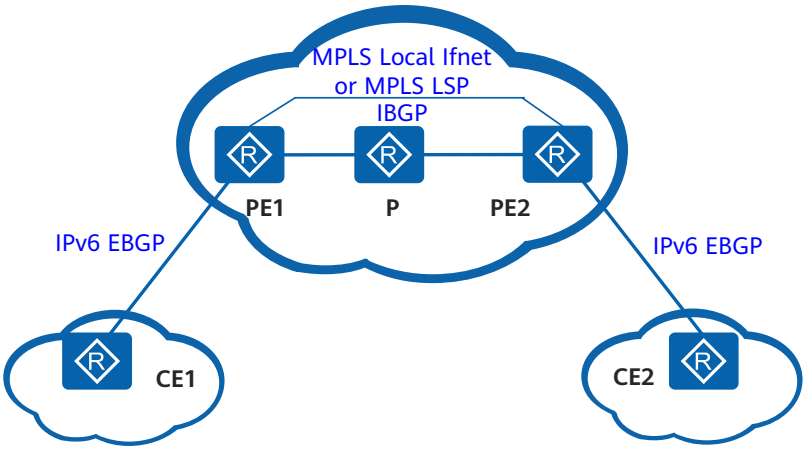


Figure 9-17 shows route and packet transmission in an intra-AS 6PE scenario. In this figure, CE2 sends routes to CE1, and CE1 sends packets to CE2. I-L indicates an inner label, and O-L indicates an outer label. The outer label is allocated by MPLS. The outer label directs the packet to the BGP next hop, and the inner label identifies the outbound interface or CE to which the packet should be forwarded.

The route transmission process is as follows:

1. CE2 sends an IPv6 route to PE2, its EBGP peer.
2. Upon receipt, PE2 changes the next hop of the IPv6 route to itself and assigns a label to the IPv6 route. Then, PE2 sends the labeled IPv6 route to PE1, its IBGP peer.
3. Upon receipt, PE1 relays the labeled IPv6 route to a tunnel and adds information about the route to the local forwarding table. Then, PE1 changes the next hop of the route to itself, removes the label of the route, and sends the route to CE1, its EBGP peer.

The IPv6 route transmission from CE2 to CE1 is complete.

The packet transmission process is as follows:

1. CE1 sends an ordinary IPv6 packet to PE1 over an IPv6 link on the public network.
2. Upon receipt, PE1 searches its local forwarding table for the forwarding entry based on the destination address of the packet and encapsulates the packet

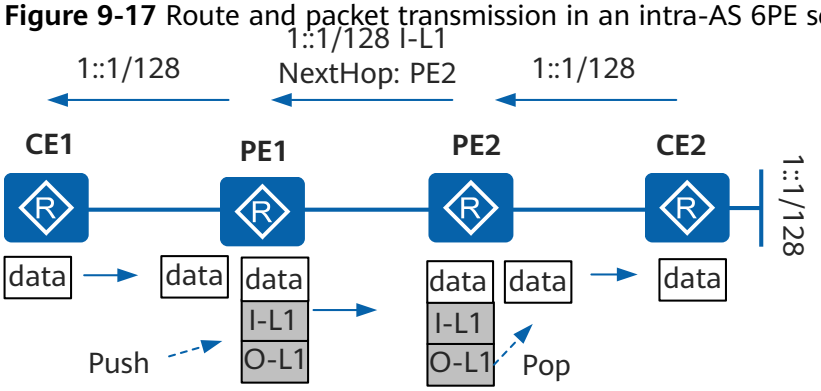
with inner and outer labels. Then, PE1 sends the IPv6 packet to PE2 over a public network tunnel.

3. Upon receipt, PE2 removes the inner and outer labels and forwards the IPv6 packet to CE2 over an IPv6 link.

As a result, the IPv6 packet is transmitted from CE1 to CE2.

The route and packet transmission processes show that whether the public network is an IPv4 or IPv6 network does not matter to the CEs.

Figure 9-17 Route and packet transmission in an intra-AS 6PE scenario

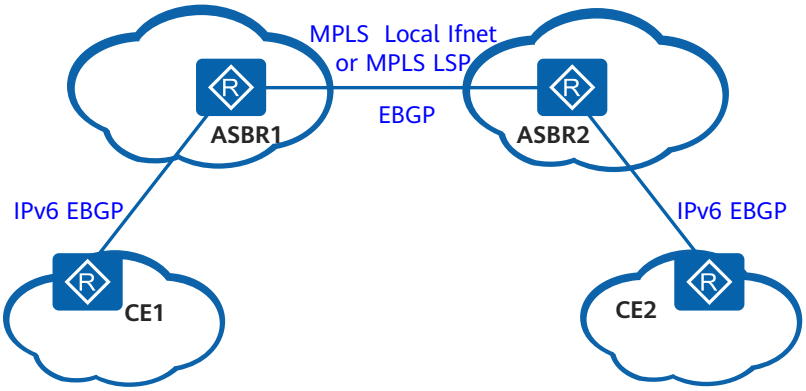


Inter-AS 6PE

- **Inter-AS 6PE OptionB (with ASBRs as PEs)**

  Figure 9-18 shows inter-AS 6PE OptionB (with ASBRs as PEs) networking. Inter-AS 6PE OptionB (with ASBRs as PEs) is similar to intra-AS 6PE. The only difference is that in an inter-AS 6PE OptionB scenario in which ASBRs also function as PEs, ASBRs establish EBGP peer relationships between each other. The route and packet transmission processes in an inter-AS 6PE OptionB scenario in which ASBRs also function as PEs are similar to those in an intra-AS 6PE scenario.

  Figure 9-18 Networking diagram for inter-AS 6PE OptionB (with ASBRs as PEs)



- **Inter-AS 6PE OptionB**

  Figure 9-19 shows inter-AS 6PE OptionB networking. ASBRs exchange labeled IPv6 routes with each other or with PEs using an IPv4 routing protocol. Tunnels must be established between ASBRs and between PEs and ASBRs to transparently transmit IPv6 packets. MPLS LSPs, MPLS Local IFNET tunnels, MPLS TE tunnels, and GRE tunnels are often used between ASBRs to transmit IPv6 packets. By default, an ASBR uses an MPLS LSP to transmit IPv6 packets. If no MPLS LSP is available, an ASBR uses an MPLS Local IFNET tunnel to transmit IPv6 packets. If you want an ASBR to transmit IPv6 packets over an MPLS TE or a Generic Routing Encapsulation (GRE) tunnel, configure a tunnel policy on the ASBR.

  Figure 9-19 Networking diagram for inter-AS 6PE OptionB

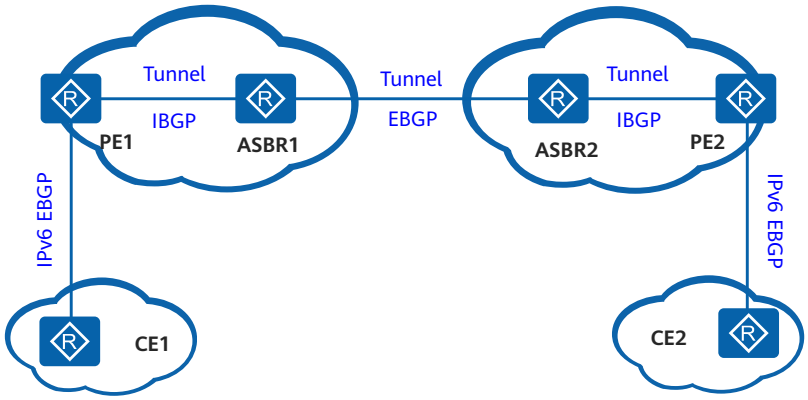
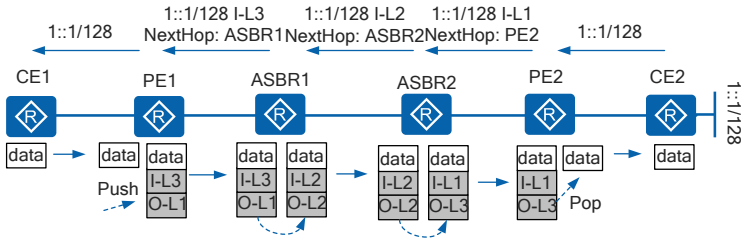


  Figure 9-20 shows route and packet transmission in an inter-AS 6PE OptionB scenario. In this figure, CE2 sends routes to CE1, and CE1 sends packets to CE2. I-L indicates an inner label, and O-L indicates an outer label.

  Figure 9-20 Route and packet transmission in an inter-AS 6PE OptionB scenario



  The route transmission process is as follows:

  a. CE2 sends an IPv6 route to PE2, its EBGP peer.

  b. Upon receipt, PE2 changes the next hop of the IPv6 route to itself and assigns a label to the IPv6 route. Then, PE2 sends the labeled IPv6 route to ASBR2 over an IBGP peer session.

c. Upon receipt, ASBR2 relays the route to a tunnel and adds information about the route to the local forwarding table. Then, ASBR2 changes the next hop of the route to itself, replaces the label of the route, and sends the route to ASBR1, its EBGP peer.

d. Upon receipt, ASBR1 relays the route to a tunnel and adds information about the route to the local forwarding table. Then, ASBR1 changes the next hop of the route to itself, replaces the label of the route, and sends the route to PE1, its IBGP peer.

e. Upon receipt, PE1 relays the route to a tunnel and adds information about the route to the local forwarding table. Then, PE1 changes the next hop of the route to itself, removes the label of the route, and sends the route to CE1, its EBGP peer.

As a result, the IPv6 route is transmitted from CE2 to CE1.

The packet transmission process is as follows:

a. CE1 sends an ordinary IPv6 packet to PE1 over an IPv6 link on the public network.

b. Upon receipt, PE1 looks up its local forwarding table based on the destination address of the packet and encapsulates the packet with inner and outer labels. Then, PE1 sends the IPv6 packet to ASBR1 over a public network tunnel.

c. Upon receipt, ASBR1 removes the inner and outer labels of the packet, looks up the local forwarding table based on the destination address of the packet, and encapsulates the packet with new inner and outer labels. Then, ASBR1 sends the IPv6 packet to ASBR2 over a public network tunnel.

d. Upon receipt, ASBR2 removes the inner and outer labels of the packet, looks up the local forwarding table based on the destination address of the packet, and encapsulates the packet with new inner and outer labels. Then, ASBR2 sends the IPv6 packet to PE2 over a public network LSP.

e. Upon receipt, PE2 removes the inner and outer labels and forwards the IPv6 packet to CE2 over an IPv6 link.

As a result, the IPv6 packet is transmitted from CE1 to CE2.

- **Inter-AS 6PE OptionC**

  Figure 9-21 shows inter-AS 6PE OptionC networking. In an inter-AS 6PE OptionC scenario, PEs establish multi-hop MP-EBGP peer relationships between each other and exchange labeled IPv6 routes using an IPv4 routing protocol. PEs exchange IPv6 packets over end-to-end BGP LSPs.

  NOTE

  Two inter-AS 6PE OptionC solutions are available, depending on the establishment methods of end-to-end LSPs. In an inter-AS 6PE OptionC scenario, PEs establish multi-hop MP-EBGP peer relationships to exchange labeled IPv6 routes and establish end-to-end BGP LSPs to transmit IPv6 packets. The way in which an end-to-end BGP LSP is established does not matter much to inter-AS 6PE OptionC and therefore is not described here.

Figure 9-21 Networking diagram for inter-AS 6PE OptionC

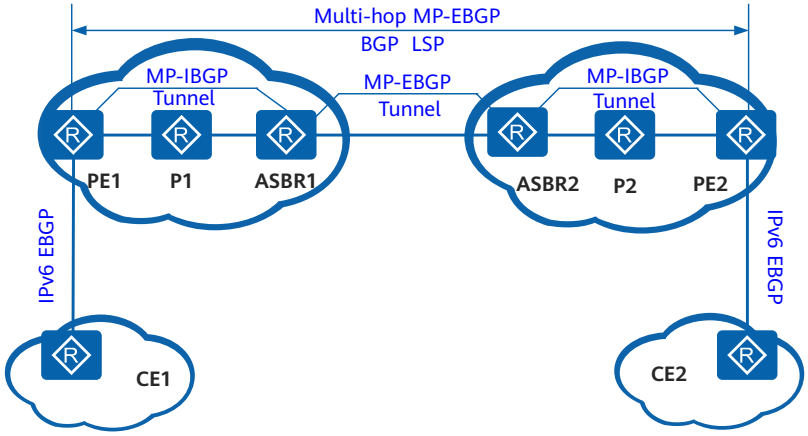


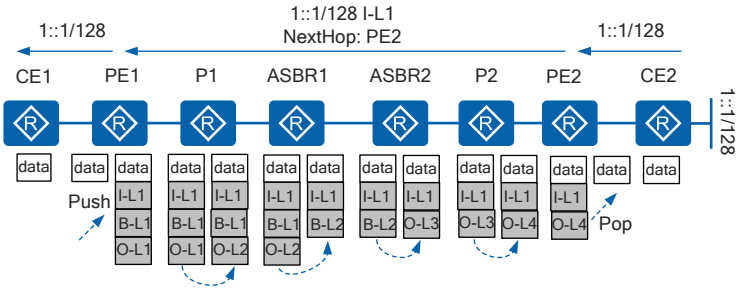
Figure 9-22 shows route and packet transmission in an inter-AS 6PE OptionC scenario. In this figure, CE2 sends routes to CE1, and CE1 sends packets to CE2. I-L indicates an inner label, B-L indicates a BGP LSP label, and O-L indicates an outer label.

NOTE

To simplify the description of the figure, it is assumed that:

- The two ASBRs are connected by an MPLS Local IFNET tunnel.
- MPLS does not use the penultimate hop popping (PHP) function.

Figure 9-22 Route and packet transmission in an inter-AS 6PE OptionC scenario



The route transmission process is as follows:

a. CE2 sends an IPv6 route to PE2, its EBGP peer.

b. Upon receipt, PE2 changes the next hop of the IPv6 route to itself and assigns a label to the IPv6 route. Then, PE2 sends the labeled IPv6 route to PE1, its MP-EBGP peer.

c. Upon receipt, PE1 relays the route to a tunnel and adds information about the route to the local forwarding table. Then, PE1 changes the next hop of the route to itself, removes the label of the route, and sends the route to CE1, its EBGP peer.

The IPv6 route transmission from CE2 to CE1 is complete. During this process, ASBRs transparently transmit information about the labeled IPv6 route.

The packet transmission process is as follows:

a. CE1 sends an ordinary IPv6 packet to PE1 over an IPv6 link on the public network.

b. Upon receipt, PE1 searches its local forwarding table for the forwarding entry based on the destination address of the packet, changes the next hop of the packet based on the search result, and encapsulates the packet with an inner label, a BGP LSP label, and an outer label. Then, PE1 sends the IPv6 packet to P1 over a public network tunnel.

c. Upon receipt, P1 replaces the outer label of the packet and forwards the packet to ASBR1 over a public network tunnel.

d. Upon receipt, ASBR1 removes the outer and BGP LSP labels and encapsulates the packet with a new BGP LSP label. Then, ASBR1 sends the IPv6 packet to ASBR2 over a public network tunnel.

e. Upon receipt, ASBR2 removes the BGP LSP label and encapsulates the packet with an outer label. Then, ASBR2 sends the IPv6 packet to PE2 over a public network tunnel.

f. Upon receipt, P2 replaces the outer label of the packet and forwards the packet to PE2 over a public network tunnel.

g. Upon receipt, PE2 removes the inner and outer labels and forwards the IPv6 packet to CE2 over an IPv6 link.

As a result, the IPv6 packet is transmitted from CE1 to CE2.

Usage Scenarios

Each 6PE mode has its advantages and usage scenarios. The intra-AS 6PE mode is best suited for scenarios in which separate IPv6 networks connect to the same AS. Inter-AS 6PE modes are best suited for scenarios in which separate IPv6 networks connect to different ASs. **Table 9-5** lists the usage scenarios for inter-AS 6PE modes.

Table 9-5 Usage scenarios for inter-AS 6PE modes

Mode	Characteristic	Usage Scenario
Inter-AS 6PE OptionB (with ASBRs as PEs)	Advantage: Configuration is similar to that for intra-AS 6PE and additional inter-AS configuration is not required. Disadvantage: Network expansibility is poor. ASBRs must have high performance to manage information about all labeled IPv6 routes.	A small network with ASBRs in different ASs to which separate IPv6 networks are connected. The smaller the number of ASs spanned, the more obvious the advantage of this solution is.

Mode	Characteristic	Usage Scenario
Inter-AS 6PE OptionB	Advantage: MPLS tunnels are established segment by segment and easy to manage. Disadvantage: Information about labeled IPv6 routes is stored on and advertised by ASBRs. If a large number of VPN routes exist, the overburdened ASBRs are likely to encounter bottlenecks.	An inter-AS OptionB public network with tunnels established between the PEs in different ASs and with separate IPv6 networks connected to the ASs.
Inter-AS 6PE OptionC	Advantage: • Labeled IPv6 routes are directly exchanged between the ingress and egress PEs. • Information about labeled IPv6 routes is managed by PEs only and ASBRs are no longer the bottlenecks. Disadvantage: Management costs of end-to-end connections between PEs are high.	An inter-AS OptionC public network with end-to-end tunnels established between the PEs in different ASs and with separate IPv6 networks connected to the ASs. The greater the number of ASs spanned, the more obvious the advantage of this solution is.

Benefits

6PE offers the following benefits:

- Easy maintenance: All configurations are performed on PEs and network maintenance is simple. IPv6 services are carried over IPv4 networks, but the users on IPv6 networks are unaware of IPv4 networks.
- Low network construction costs: Service providers can provide IPv6 services over existing MPLS networks without upgrading the networks. 6PE devices can provide multiple types of services, such as IPv6 VPN and IPv4 VPN.

9.2.17 6PE Routes Sharing the Explicit Null Label

On an IPv6 provider edge (6PE) networking, by default, each 6PE route is assigned. Therefore, each route advertised to other 6PE peers needs to apply for a label. The number of required labels is directly proportional to the number of 6PE routes. When there are many 6PE routes, a large number of labels are required.

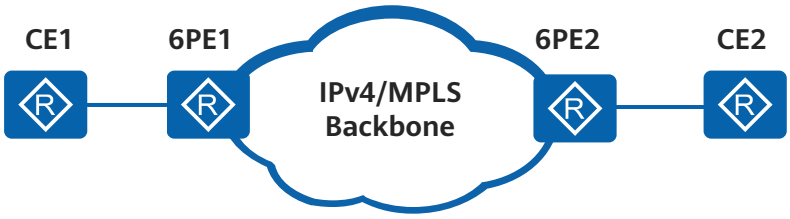
After 6PE routes sharing the explicit null label is enabled, all 6PE routes share the explicit null label, without applying for labels. In such a case, the number of

required labels is irrelevant to the number of 6PE routes, saving label resources on 6PE routers.

The explicit null label is a special label which needs to be popped out on the egress PE. The packets then must be forwarded on the basis of IPv6.

In the 6PE networking shown in **Figure 9-23**, 6PE routes sharing the explicit null label is enabled on 6PE1. 6PE1 then can advertise routes sharing the explicit null label to 6PE2 without applying for a label for each route. When 6PE2 sends data to 6PE1, the data packet carries two labels, the top label being the label assigned by LDP and the bottom label is the explicit null label assigned by MP-BGP. After the data packet reaches 6PE1, 6PE1 pops the explicit null label and forwards the IPv6 data packet to CE1.

Figure 9-23 6PE networking diagram



Note that when you enable or disable 6PE routes sharing the explicit null label after a 6PE peer relationship is set up, temporary packet loss occurs. Therefore, enable this function prior to setting up a 6PE peer relationship.

9.3 Summary of BGP Configuration Tasks

After basic BGP functions are configured, you can enable basic communication functions on BGP networks. If other BGP functions are required, configure them according to reference sections.

Table 9-6 describes the BGP configuration tasks.

NOTE

If BGP is configured on an IPv6 network, all the peer addresses specified in the **Peer** command must be IPv6 addresses.

Table 9-6 BGP configuration tasks

Scenario	Description	Task
Configuring basic BGP functions	The configuration of basic BGP functions is the foundation of the BGP network construction and the precondition for other BGP functions.	**9.6 Configuring Basic BGP Functions**

Scenario	Description	Task
Configuring BGP security	On BGP networks, unauthorized users can attack the BGP network by modifying data packets or forging authorized users. To ensure security of services carried on BGP networks, configure BGP MD5 authentication, BGP Keychain authentication, or Generalized TTL Security Mechanism (GTSM) function.	**9.7 Configuring BGP Security**
Simplifying IBGP network connection	Because routes received from the IBGP neighbors will not be sent to other IBGP neighbors, fully-meshed connections must be established on the IBGP network. However, when the number of devices is large, peer configuration is very complex on the fully-meshed IBGP network, and the consumption of network resources and device CPU resources will increase. To reduce the number of IBGP network connections and better plan the network, configure the route reflector and confederation.	**9.8 Simplifying IBGP Network Connections**

Scenario	Description	Task
Configuring BGP route selection and load balancing	In a BGP routing table, multiple routes to the same destination may exist. To guide route selection, BGP defines next-hop policies and route selection rules. The priority of next-hop policies is higher than that of BGP route selection rules. After the next-hop policies are performed, BGP selects routes according to the route selection rules. Usually there are multiple valid routes to the same destination on the network. If BGP only advertises the optimal route to its peer, unbalanced traffic on different routes will occur. The BGP load-balancing configuration can balance load on different routes and reduce network congestion.	**9.9 Configuring BGP Route Selection and Load Balancing**
Controlling advertising and receiving of BGP routes	With the expansion of the network scale, the sharp increase of routing tables leads to greater load on networks and increasing network security problems. To solve this problem, filter routes according to the routing policies and only send and receive required BGP routes. In addition, multiple routes to the same destination may exist. If these routes need to pass through different ASs, direct service traffic to specific ASs or filter the routes to be advertised.	**9.10 Controlling the Receiving and Advertisement of BGP Routes**

Scenario	Description	Task
Configuring and adjusting the BGP network convergence rate	To enable BGP to rapidly detect network changes, speed up the BGP network convergence. To minimize the effect on networks from route flapping and reduce load on the device, slow down the BGP network convergence.	**9.11 Adjusting the BGP Network Convergence Speed**
Configuring BGP reliability	To avoid long service interruption when faults occur on BGP networks, adopt the solution of standby link. However, the BGP mechanism requires more than one second to detect the faults and perform active/standby switchover. To ensure that users of delay-sensitive services such as the voice service do not detect the service interruption, associate BGP tracking, BGP, and BFD to implement fast fault detection, and meanwhile use BGP GR to perform fast switchover after the fault detection.	**9.12 Configuring BGP Reliability**

Scenario	Description	Task
Configuring BGP route aggregation	The BGP routing table on a medium or large BGP network contains a large number of routing entries. Storing the routing table consumes a large number of memory resources, and transmitting and processing the routing information consumes a large number of network resources. Route aggregation can reduce the size of a routing table, prevent specific routes from being advertised, and minimize the impact of route flapping on networks. Although BGP automatic route aggregation is easy to configure, it only aggregates routes according to the natural network segment. BGP manual route aggregation can be used with flexible routing policies to enable BGP to effectively transmit and control routes.	**9.13 Configuring BGP Route Summarization**

Scenario	Description	Task
Configuring BGP neighbors to advertise routes on demand	BGP Outbound Route Filters (ORF) is used to enable BGP neighbors to advertise routes on demand. If neighbors of the local BGP device have different route requirements, different export policies must be configured on the local BGP device. In this case, the configuration workload and maintenance costs of the local BGP device will increase. To solve this problem, configure BGP ORF on BGP neighbor devices, allowing BGP neighbor devices to maintain route policies on demand and send them to the local BGP device as export policies. This reduces the configuration workload and maintenance costs of the local BGP device.	**9.14 Configuring On-demand Route Advertisement**

Scenario	Description	Task
Configuring a local BGP device to send a default route to its peer	The BGP routing table on a medium or large BGP network contains a large number of routing entries. Storing the routing table consumes a large number of memory resources, and transmitting and processing the routing information consumes a large number of network resources. If multiple routes in a peer BGP routing table are sent only from a local device, configure the local device to send a default route to its peer. In this case, the local device will send a default route with the next hop address as the local address to its peer, regardless of whether there is a default route in the local routing table. After the local device is configured to send only the default route to its peer using the routing policies, the number of network routes is greatly reduced and the peer memory resources and network resources are largely saved.	**9.15 Configuring BGP to Advertise Default Routes to Peers**

Scenario	Description	Task
Configuring path MTU auto discovery	BGP path MTU auto discovery allows the discovery of the minimum MTU value (path MTU) on the network path from the source to the destination, which enables TCP to transmit the BGP messages according to the path MTU. This increases the BGP message transmission efficiency and improves the BGP performance.	**9.16 Configuring Path MTU Auto Discovery**
Configuring MP-BGP	Traditional BGP-4 only manages IPv4 unicast routing information and does not support route transmission between ASs of other networks such as IPv6 and multicast networks. To support multiple network layer protocols, the Internet Engineering Task Force (IETF) extends BGP-4 to Multiprotocol Extensions for BGP-4 (MP-BGP). Features supported by MP-BGP on IPv6 networks are called BGP4+ and multicast networks are called Multicast BGP (MBGP).	**9.17 Configuring MP-BGP**

9.4 Licensing Requirements and Limitations for BGP

Involved Network Elements

None

Licensing Requirements

BGP is a basic feature of a router and is not under license control.

Feature Limitations

The QPPB feature is just for beta test, and is not for commercial use. If the feature is required in the test, contact Huawei technical support personnel.

9.5 Default Settings for BGP

Table 9-7 describes the default settings for BGP.

Table 9-7 Default settings for BGP

Parameter	Default Setting
BGP	Disabled
Keepalive message interval	60s
Hold time	180s

9.6 Configuring Basic BGP Functions

Before building a BGP network, you need to configure basic BGP functions.

Pre-configuration Tasks

Before configuring basic BGP functions, complete the following task:

- Configuring IP addresses for interfaces to ensure network-layer communication between neighbor nodes

Configuration Procedure

Perform the following operations in sequence and as required.

9.6.1 Starting a BGP Process

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

BGP is started, the local AS number is specified, and the BGP view is displayed.

NOTICE

After BGP peers are configured, changing the router ID of a BGP peer resets BGP peer relationships.

Step 3 Run **router-id** *ipv4-address*

NOTE

By default, BGP automatically selects the router ID in the system view. If the IP address of a physical interface is used as the router ID, route flapping occurs when the IP address of the physical interface changes. To enhance network stability, configuring the address of a loopback interface as the router ID is recommended. For Router ID selection rules in the system view, see descriptions in Command Reference about the **router-id** command.

----End

9.6.2 Configuring BGP Peers

Context

During the configuration of BGP peers, if the AS number of the specified peer is the same as the local AS number, an IBGP peer is configured. If the AS number of the specified peer is different from the local AS number, an EBGP peer is configured. To enhance the stability of BGP connections, you are advised to use the reachable loopback interface addresses to establish BGP connections.

When loopback interface addresses are used to establish a BGP connection, run the **peer connect-interface** command on both ends of the BGP connection to ensure the correctness of interfaces and addresses on the TCP connection. If the command is run on only one end, the BGP connection may fail to be established.

When loopback interface addresses are used to establish an EBGP connection, the **peer ebgp-max-hop** command with *hop-count* greater than or equal to 2 must be run. Otherwise, the EBGP connection cannot be established.

To perform the same configuration on a large number of peers, configure a BGP peer group according to **9.6.3 (Optional) Configuring a BGP Peer Group** to reduce the configuration workload.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** { *ipv4-address* | *ipv6-address* } **as-number** { *as-number-plain* | *as-number-dot* }

The BGP peer is created.

By default, BGP does not create BGP peers.

Step 4 (Optional) Run **peer** *ipv4-address* **connect-interface** *interface-type interface-number* [*ipv4-source-address*]Or **peer** *ipv6-address* **connect-interface** *interface-type interface-number* [*ipv6-source-address*]

A source interface from which BGP packets are sent, and a source address used for initiating a connection.

By default, the outbound interface of a BGP packet serves as the source interface of a BGP packet.

Step 5 (Optional) Run **peer connected-check-ignore**

The device has been configured not to check the hop count when establishing a one-hop EBGP peer relationship using a loopback interface address.

NOTE

To allow one-hop EBGP peer relationships to be established using loopback interface addresses, run the command or the **peer ebgp-max-hop** command (in which *hop-count* is greater than or equal to 2).

Step 6 (Optional) Run **peer** { *ipv4-address* | *ipv6-address* } **ebgp-max-hop** [*hop-count*]

The maximum number of hops allowed for the establishment of an EBGP connection is set.

By default, the maximum number of hops allowed for an EBGP connection is 1. That is, an EBGP connection must be established on a directly connected physical link.

Step 7 (Optional) Run **peer** { *ipv4-address* | *ipv6-address* } **description** *description-text*

The description of the peer is configured.

NOTE

If a BGP peer group is configured on an IPv4 unicast network, steps 7 and 8 are not required. If a BGP peer group is configured on an IPv4 unicast network and an IPv6 unicast network, steps 7 and 8 are required.

Step 8 (Optional) Run the following commands as required.

- Run **ipv4-family multicast**

  The BGP-IPv4 multicast address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The BGP-IPv6 unicast address family view is displayed.

Step 9 (Optional) Run **peer** { *ipv4-address* | *ipv6-address* } **enable**

MP-BGP is enabled on the BGP peers to configure them as MP-BGP peers.

----End

9.6.3 (Optional) Configuring a BGP Peer Group

Context

A large BGP network has a large number of peers. It is difficult to configure and maintain these peers. You can add the BGP peers with the same configurations to a BGP peer group and then configure the BGP peers in batches. This simplifies peer management and improves route advertisement efficiency.

NOTE

- If a function is configured on a peer and its peer group, the function configured on the peer takes precedence over that configured on the peer group.
- When loopback interface or sub-interface addresses are used to establish a BGP connection, you are advised to perform step 6 on both ends of the BGP connection simultaneously to ensure the correct establishment of the connection. If step 6 is performed on only one end, the BGP connection may fail to be established.
- When loopback interfaces are used to establish an EBGP connection, step 7 is required and *hop-count* in the *peer ebgp-max-hop* command must be greater than or equal to 2. Otherwise, the EBGP connection cannot be established.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **group** *group-name* [**external** | **internal**]

A BGP peer group is created.

NOTE

The AS number of an IBGP peer group is the local AS number. Therefore, step 4 is not required.

Step 4 Run **peer** *group-name* **as-number** { *as-number-plain* | *as-number-dot* }

An AS number is configured for the EBGP peer group.

NOTE

To add an EBGP peer to a peer group, configure the EBGP peer according to **9.6.2 Configuring BGP Peers** and then perform step 5.

To add an IBGP peer to a peer group, perform step 5. The system creates an IBGP peer in the BGP view and sets its AS number as the AS number of the peer group.

Step 5 Run **peer** { *ipv4-address* | *ipv6-address* } **group** *group-name*

A peer is added to the peer group.

NOTE

You can repeat step 5 to add multiple peers to a peer group.

Step 6 (Optional) Run **peer** *group-name* **connect-interface** *interface-type interface-number* [*ipv4-source-address*]Or **peer** *group-name* **connect-interface** *interface-type interface-number* [*ipv6-source-address*]

A source interface and a source IP address are specified for the peer to establish a TCP connection.

By default, the outbound interface of a BGP packet serves as the source interface of a BGP packet.

NOTE

The configurations of GTSM and EBGP-MAX-HOP affect the TTL values of BGP packets, which may cause a conflict between TTL values. Therefore, you can configure only one of the two functions for a peer or peer group.

Step 7 (Optional) Run **peer** *group-name* **ebgp-max-hop** [*hop-count*]

The maximum number of hops allowed for the establishment of an EBGP connection is set.

By default, the maximum number of hops allowed for an EBGP connection is 1. That is, an EBGP connection must be established on a directly connected physical link.

Step 8 (Optional) Run **peer** *group-name* **description** *description-text*

The description is configured for the peer group.

NOTE

If a BGP peer group is configured on an IPv4 unicast network, steps 9 and 10 are not required. If a BGP peer group is configured on an IPv4 multicast network and an IPv6 unicast network, steps 9 and 10 are required.

Step 9 (Optional) Run the following commands as required.

- Run **ipv4-family multicast**

  The BGP-IPv4 multicast address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The BGP-IPv6 unicast address family view is displayed.

Step 10 Run **peer** *group-name* **enable**

MP-BGP is enabled on the BGP peers to configure them as MP-BGP peers.

----End

9.6.4 Configuring BGP to Import Routes

Context

BGP cannot discover routes and needs to import routes such as IGP routes into BGP routing tables so that the imported routes can be transmitted within an AS or between ASs. BGP imports routes in either import or network mode:

- In import mode, BGP imports IGP routes, including RIP, OSPF, and IS-IS routes, into BGP routing tables based on protocol type. To ensure the validity of imported IGP routes, BGP can also import static routes and direct routes in import mode.
- In network mode, BGP imports the routes in the IP routing table one by one to BGP routing tables. The network mode is more accurate than the import mode.

Procedure

- In import mode
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.
  d. Run **import-route** *protocol* [*process-id*] [**med** *med* | **route-policy** *route-policy-name*] *

     BGP is configured to import routes of other routing protocols.
  e. (Optional) Run **default-route imported**

     BGP is allowed to import default routes from the local IP routing table.

     To import default routes, you need to run both the **default-route imported** command and the **import-route (BGP)** command. If only the **import-route (BGP)** 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.

     By default, BGP does not add default routes to BGP routing tables.
- In network mode
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.
  d. Run **network** *ipv4-address* [*mask* | *mask-length*] [**route-policy** *route-policy-name*]Or **network** *ipv6-address prefix-length* [**route-policy** *route-policy-name*]

BGP is configured to import routes from the IPv4 or IPv6 routing table one by one.

----End

9.6.5 Verifying the Basic BGP Function Configuration

Procedure

- Run the **display bgp peer** [**verbose**] command to check information about all BGP peers.
- Run the **display bgp peer** *ipv4-address* { **log-info** | **verbose** } command to check information about the specified BGP peer.
- Run the **display bgp routing-table** [*ipv4-address* [{ *mask* | *mask-length* } [**longer-prefixes**]]] command to check BGP routing information.
- Run the **display bgp group** [*group-name*] command to check information about the specified BGP peer group.
- Run the **display bgp multicast peer** [[*peer-address*] **verbose**] command to check information about the specified MBGP peer.
- Run the **display bgp multicast group** [*group-name*] command to check information about an MBGP peer group.
- Run the **display bgp multicast network** command to check the routing information that MBGP advertises.
- Run the **display bgp multicast routing-table** [*ip-address* [*mask-length* [**longer-prefixes**] | *mask* [**longer-prefixes**]]] command to check the MBGP routing information of a specified network in the MBGP routing table.

----End

9.7 Configuring BGP Security

Configuring connection authentication and BGP GTSMfor BGP peers can improve BGP network security.

Pre-configuration Tasks

Before configuring BGP security, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedure

You can perform the following configuration tasks as required. The following configuration tasks (excluding the task of Verifying the BGP Security Configuration) can be performed in any sequence.

9.7.1 Configuring MD5 Authentication

Context

BGP uses TCP as the transmission protocol, and 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. However, most parameters in a packet may be easily obtained by attackers. To protect BGP from attacks, MD5 authentication or keychain authentication can be used between BGP peers to reduce the possibility of attacks. The MD5 algorithm is easy to configure, generates a single password that needs to be manually changed.

NOTICE

If **simple** is selected during the configuration of the MD5 authentication password, the password is saved in the configuration file in plain text. This brings security risks. It is recommended that you select **cipher** to save the password in cipher text. MD5 authentication has potential security risks.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **password** { **cipher** *cipher-password* | **simple** *simple-password* }

The MD5 authentication password is set.

NOTE

- To prevent the MD5 password set on BGP peers from being decrypted, update the MD5 password periodically.
- BGP MD5 authentication and BGP keychain authentication are mutually exclusive, and only one of them can be configured for a BGP peer.

----End

9.7.2 Configuring Keychain Authentication

Context

BGP uses TCP as the transmission protocol, and 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. However, most parameters in a packet may be easily obtained by attackers. To protect BGP from attacks, use MD5 authentication or keychain authentication between BGP peers to reduce the

possibility of attacks. The keychain algorithm is complex to configure and generates a set of passwords. Keychain authentication allows automatically changing a password based on the configuration. Therefore, keychain authentication applies to networks requiring high security.

NOTE

Before configuring BGP keychain authentication, configure a keychain corresponding to *keychain-name*. Otherwise, the TCP connection cannot be established. For details about configuring a keychain, see Keychain Configuration in the *Huawei AR Series Configuration Guide - Security Configuration*.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **keychain** *keychain-name*

Keychain authentication is configured.

NOTE

- You must configure keychain authentication on both BGP peers. Encryption algorithms and passwords configured on both peers must be the same; otherwise, the TCP connection cannot be established between BGP peers and BGP messages cannot be transmitted. SHA256 and HMAC-SHA256 encryption algorithm are recommended in keychain authentication.
- BGP MD5 authentication and BGP keychain authentication are mutually exclusive, and only one of them can be configured for a BGP peer.

----End

9.7.3 Configuring BGP GTSM

Context

To protect a device against the attacks of forged BGP packets, you can configure GTSM to check whether the TTL value in the IP packet header is within the specified range. GTSM allows or discards packets of which TTL values are not within the specified range according to networking requirements. When the default action to be taken on packets is set to drop in GTSM, set a proper TTL range according to the network topology. Then packets of which TTL values are not within the specified range are discarded. This prevents attackers from sending forged BGP packets to consume CPU resources.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

NOTE

The configurations of GTSM and **peer ebgp-max-hop** affect the TTL values of BGP packets, which may cause a conflict between TTL values. Therefore, you can configure only one of the two functions for a peer or peer group.

Step 3 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **valid-ttl-hops** [*hops*]

BGP GTSM is configured.

By default, GTSM is not configured on any BGP peer or peer group.

Step 4 (Optional) Run the following command in the system view:

gtsm default-action { **drop** | **pass** }

The default action to be taken on the packets that do not match a GTSM policy is set.

By default, the action to be taken on the packets that do not match the GTSM policy is pass.

Step 5 (Optional) Run the following command in the system view:

gtsm log drop-packet all

The log function is enabled on boards.

The log records information that GTSM drops packets, which helps locate faults.

----End

9.7.4 Verifying the BGP Security Configuration

Procedure

- Run the **display bgp peer verbose** command to check authentication detailed information about the specified BGP peer.

----End

9.8 Simplifying IBGP Network Connections

Configuring a route reflector and a confederation on an IBGP network can simplify IBGP network connections.

Pre-configuration Tasks

Before simplifying IBGP network connections, complete the following configuration task:

- **Configuring Basic BGP Functions**

Configuration Procedure

Perform the following configuration tasks in any sequence as required.

9.8.1 Configuring a BGP Route Reflector

Context

To ensure the connectivity between IBGP peers within an AS, you need to establish full-mesh connections between the IBGP peers. When there are many IBGP peers, it is costly to establish a fully-meshed network. A route reflector (RR) can solve this problem.

A cluster ID can help prevent routing loops between multiple RRs within a cluster and between clusters. When a cluster has multiple RRs, the same cluster ID must be configured for all the RRs within the cluster.

If full-mesh IBGP connections are established between clients of multiple RRs, route reflection between clients is not required and wastes bandwidth resources. In this case, prohibit route reflection between clients to reduce the network burden.

Within an AS, an RR transmits routing information and forwards traffic. When an RR connects to a large number of clients and non-clients, many CPU resources are consumed if the RR transmits routing information and forwards traffic simultaneously. This also reduces route transmission efficiency. To improve route transmission efficiency, prohibit BGP from adding preferred routes to IP routing tables on the RR to enable the RR only to transmit routing information.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

BGP is enabled and the BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family unicast**

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **reflect-client**

An RR and its client are configured.

By default, the route reflector and its client are not configured.

Step 5 (Optional) Run **reflector cluster-id** *cluster-id*

A cluster ID is configured for the RR.

By default, each RR uses its router ID as the cluster ID.

Step 6 (Optional) Run **undo reflect between-clients**

Route reflection is prohibited between clients.

By default, route reflection is allowed between clients.

Step 7 (Optional) Run **routing-table rib-only** [**route-policy** *route-policy-name*]

BGP is prohibited from adding preferred routes to IP routing tables.

By default, BGP adds preferred routes to IP routing tables.

----End

Verifying the Configuration

- Run the **display bgp group** [*group-name*] command to check information about the specified BGP peer group.
- Run the **display bgp routing-table** [*ipv4-address* [{ *mask* | *mask-length* } [**longer-prefixes**]]] command to check routing information in a BGP routing table.
- Run the **display bgp multicast routing-table** [*ip-address* [*mask-length* [**longer-prefixes**] | *mask* [**longer-prefixes**]]] command to check the MBGP routing table.

9.8.2 Configuring a BGP Confederation

Context

A confederation divides an AS into sub-ASs. Within each sub-AS, IBGP peers establish full-mesh connections or have an RR configured. Sub-ASs establish EBGP connections. On a large BGP network, configuring a confederation can reduce the number of IBGP connections, simplify routing policy management, and improve route advertisement efficiency.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

BGP is enabled and the BGP view is displayed.

Step 3 Run **confederation id** { *as-number-plain* | *as-number-dot* }

A confederation ID is configured.

By default, no BGP confederation is configured.

NOTICE

An old speaker that has a 2-byte AS number cannot be in the same confederation with a new speaker that has a 4-byte AS number. Otherwise, a routing loop may occur. This is because the AS4_Path attribute does not support confederations.

Step 4 Run **confederation peer-as** { *as-number-plain* | *as-number-dot* } &<1-32>

A sub-AS number is configured for a confederation.

By default, no sub-AS number of the confederation is configured.

Step 5 (Optional) Run **confederation nonstandard**

Confederation compatibility is configured.

By default, confederations comply with RFC 3065.

----End

Verifying the Configuration

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to check detailed information about BGP peers.
- Run the **display bgp routing-table** [*ipv4-address* [{ *mask* | *mask-length* } [**longer-prefixes**]]] command to check routing information in a BGP routing table.

9.9 Configuring BGP Route Selection and Load Balancing

BGP has many route attributes. These attributes can be configured to change the route selection result.

Pre-configuration Tasks

Before configuring BGP route attributes, complete the following task:

- **Configure Basic BGP Functions**.

Configuration Procedure

Perform the following configuration tasks as required. The following configuration tasks (excluding the task of Verifying the BGP Route Selection and Load Balancing Configuration) can be performed in any sequence. For detailed route selection rules, see **9.2.5 BGP Route Selection Rules and Load Balancing**.

9.9.1 Configuring the BGP Priority

Context

The routing protocols may share and select routing information because routers may run multiple dynamic routing protocols at the same time. The system sets a

default priority for each routing protocol. When multiple routing protocols are used to select routes, the route selected by the routing protocol with a higher priority takes effect.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }
  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]
  The IPv6 address family view is displayed.

Step 4 Run **preference** { *external internal local* | **route-policy** *route-policy-name* } or **preference** *external internal local* **route-policy** *route-policy-name*

The BGP priority is set.

The default BGP priority is 255.

The smaller the preference value, the higher the preference.

BGP has the following types of routes:

- EBGP routes learned from peers in other ASs
- IBGP routes learned from peers in the same AS
- Locally originated routes (A locally originated route is a route summarized by using the **summary automatic** command or the **aggregate** command.)

Different preference values can be set for these three types of routes.

In addition, a routing policy can also be used to set the preferences for the routes that match the policy. The routes that do not match the policy use the default preference.

If both *external internal local* and **route-policy** *route-policy-name* are specified in the command, the priority of the routes that match the route-policy is set based on the route-policy, and the priorities of other routes are set based on the *external internal local* configuration.

----End

9.9.2 Configuring the Next_Hop Attribute

Context

When an Autonomous System Boundary Router (ASBR) forwards the route learned from an EBGP peer to an IBGP peer, the ASBR does not change the next

hop of the route by default. When the IBGP peer receives this route, it finds the next hop unreachable, sets the route to inactive, and does not use this route to guide traffic forwarding. To enable the IBGP peer to use this route to guide traffic forwarding, configure the ASBR to set its IP address as the next hop of the route when the ASBR forwards this route to the IBGP peer. After the IBGP peer receives the route from the ASBR, it finds the next hop of the route reachable, sets the route to active, and uses this route to guide traffic forwarding.

When a BGP route changes, BGP needs to iterate the indirect next hop of the route again. If no restriction is imposed on the iterated route, BGP may iterate the next hop to an incorrect forwarding path, causing traffic loss. To prevent traffic loss, configure routing policy-based route iteration.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Perform either of the following operations as required:

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **next-hop-local**

  A BGP device is configured to set its IP address as the next hop when the device advertises routes to an IBGP peer or an IBGP peer group.

  By default, a BGP device does not modify the next-hop address when advertising routes to its IBGP peers.
- Run **nexthop recursive-lookup route-policy** *route-policy-name*

  Routing-policy-based next hop iteration is configured.

  By default, routing-policy-based next hop iteration is not configured.
- Run the following command in the IPv4 unicast address family view:

  peer { *ipv4-address* | *group-name* } **next-hop-invariable**

  The device is prevented from changing the next-hop address of a route imported from an IGP before advertising the route to an IBGP peer.

  By default, a device changes the next-hop address of a route imported from an IGP to the address of the interface connecting the device to its peer when advertising the route to an IBGP peer.

NOTE

The **nexthop recursive-lookup route-policy** *route-policy-name* command does not take effect for the routes received from direct connected EBGP peers.

----End

9.9.3 Configuring the PrefVal Attribute

Context

The PrefVal attribute is a Huawei proprietary attribute and is valid only on the device where it is configured. When a BGP routing table contains multiple routes to the same destination, BGP prefers the route with the highest PrefVal.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }
  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]
  The IPv6 address family view is displayed.

Step 4 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **preferred-value** *value*

The PrefVal attribute is configured for all the routes learned from a specified peer.

By default, the PrefVal of a route learned from a peer is 0.

----End

9.9.4 Configuring the Default Local_Pref Attribute

Context

The Local_Pref attribute is used to determine the optimal route for outgoing traffic of 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 the route with the highest Local_Pref.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Run **default local-preference** *local-preference*

The default Local_Pref attribute is configured.

By default, the Local_Pref attribute is 100.

----End

9.9.5 Configuring the AS_Path Attribute

Context

The AS_Path attribute records all the ASs that a route passes through from the source to the destination in the vector order. You can configure the AS_Path attribute to implement flexible route selection.

- Generally, BGP compares the AS_Path lists of routes and prefers the route with the shortest AS_Path list. When the AS_Path attribute is not required in route selection, configure BGP not to compare the AS_Path lists of routes during route selection.
- In most cases, BGP detects routing loops based on AS number. However, to ensure correct route transmission on a hub-and-spoke network, you need to configure all the BGP peers that VPN routes advertised from a hub CE to a spoke CE pass through to accept the routes with a repeated AS number.
- Public AS numbers can be used on the Internet, but private AS numbers cannot because they may cause routing loops. To prevent routing loops, configure the AS_Path attribute to carry only public AS numbers in EBGP Update messages.
- When the AS_Path attribute is reconstructed or summarized routes are generated, you can set the maximum number of AS numbers in the AS_Path attribute. Then a BGP device checks whether the number of AS numbers in the AS_Path attribute of a route exceeds the maximum value. If so, the BGP device discards the route.
- A device usually supports only one BGP process. This indicates that a device supports only one AS number. In some cases, for example, when network migration changes an AS number, you can set a fake AS number to ensure successful network migration.
- BGP checks the first AS number in the AS_Path list that is carried in the Update message sent by an EBGP peer. If the first AS number specifies the AS where the EBGP peer resides, BGP accepts the Update message. Otherwise, BGP rejects the Update message and interrupts the EBGP connection. If you do not want BGP to check the first AS number, disable BGP from checking the first AS number.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **route-policy** *route-policy-name* { **deny** | **permit** } **node** *node*

A node is configured for a route-policy, and the view of the route-policy is displayed.

Step 3 (Optional) Configure matching rules for the route-policy to change only the community attributes of the routes that meet the matching rules.

By default, all routes meet matching rules. For details, see **10.7.2 (Optional) Configuring an if-match Clause**.

Step 4 Run **apply as-path** { *as-number-plain* | *as-number-dot* } &<1-10> { **additive** | **overwrite** }

The AS_Path attribute is set for BGP routes.

Step 5 Run **quit**

Return to the system view.

Step 6 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 7 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 8 Add the AS_Path attribute to routes.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **export**

  The AS_Path attribute is added to the routes advertised to BGP peers or peer groups.
- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **import**

  The AS_Path attribute is added to the routes received from BGP peers or peer groups.
- Run **import-route** *protocol* [*process-id*] **route-policy** *route-policy-name*

  The AS_Path attribute is added to the routes imported by BGP in import mode.
- Run **network** { *ipv4-address* [*mask* | *mask-length*] | *ipv6-address prefix-length* } **route-policy** *route-policy-name*

  The AS_Path attribute is added to the routes imported by BGP in network mode.

Step 9 (Optional) Run one of the following commands to configure the AS_Path attribute as required.

- Run **bestroute as-path-ignore**

  BGP is configured not to compare the AS_Path attributes of routes during route selection.

  By default, BGP compares the AS_Path attributes of routes during route selection.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **allow-as-loop** [*number*]

  Repeated local AS numbers are allowed in routes.

  By default, repeated local AS number is not allowed.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **public-as-only**

  BGP is configured to carry only public AS numbers in the AS_Path attribute in an EBGP Update message.

  By default, the AS_Path attribute can carry both public and private AS numbers in an EBGP Update message.

- Return to the BGP view to configure the AS_Path attribute.

  a. Run **quit**

  Return to the BGP view.

  b. (Optional) Run one of the following commands to configure the AS_Path attribute as required.

  - 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.

  - Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **fake-as** { *as-number-plain* | *as-number-dot* } [**prepend-global-as**]

    A fake AS number is configured for an EBGP peer group.

    The **peer fake-as** command can be used to hide the actual AS number of a BGP device. EBGP peers in other ASs will use the fake AS number of this BGP device to set up EBGP peer relationships with this device.

    By default, EBGP peers establish a connection using a real AS number.

    NOTICE

    Running the **undo check-first-as** command increases the probability of routing loops. Therefore, exercise caution when using this command.

  - Run **undo check-first-as**

    BGP is configured not to check the first AS number in the AS_Path list that is carried in the Update message sent by an EBGP peer.

By default, BGP checks the first AS number in the AS_Path list that is carried in the Update message sent by an EBGP peer.

NOTE

When BGP is disabled from checking the first AS number, run the **refresh bgp** command in the user view if you want BGP to check the first AS number of received routes.

----End

9.9.6 Configuring the MED Attribute

Context

The multi-exit discriminator (MED) helps determine the optimal route for incoming traffic of an AS. It is similar to the metric used in IGP. When a BGP device obtains multiple routes to the same destination address but with different next hops from EBGP peers, the BGP device selects the route with the smallest MED value as the optimal route.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Perform one of the following operations as required:

- Run **default med** *med*

  The default MED value is set.

  By default, the MED is 0.
- Run **bestroute med-none-as-maximum**

  BGP defines the MED value as the maximum value if a route does not have the MED attribute.

  By default, BGP uses the default MED value when a route does not have the MED attribute.
- Run **compare-different-as-med**

  BGP is allowed to compare the MED values of routes received from EBGP peers in any AS.

  By default, BGP compares only the MEDs of the routes received from EBGP peers within the same AS.

- Run **deterministic-med**

  The deterministic-MED function is enabled.

  By default, the BGP deterministic-MED function is disabled.
- Run **bestroute med-confederation**

  The MED values of routes in a confederation are compared.

  By default, BGP compares only the MEDs of the routes from the same AS.

----End

9.9.7 Configuring BGP to Ignore the Metric Value of the Next-Hop IGP Route When Selecting the Optimal Route

Context

On a BGP network, the BGP device always receives multiple routes with the same prefix but to different paths from neighbors. BGP must select the optimal route to the specified prefix to guide packet forwarding. By default, BGP compares the next-hop IGP route metric values of these routes and selects the route with the smallest metric value as the optimal route. For detailed BGP route selection rules, see **9.2.5 BGP Route Selection Rules and Load Balancing**.

To customize route selection policies, you can run the **bestroute igp-metric-ignore** command to configure BGP to ignore the metric value of the next-hop IGP route when selecting the optimal route.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter an address family view based on the network type, and configure the BGP device on the network.

- Run the **ipv4-family** { **unicast** | **multicast** } command to enter the IPv4 address family view.
- Run the **ipv6-family** [**unicast**] command to enter the IPv6 address family view.

Step 4 Run **bestroute igp-metric-ignore**

BGP is configured to ignore the metric value of the next-hop IGP route when selecting the optimal route.

----End

9.9.8 Configuring the BGP Community Attribute

Context

The Community attribute is a private BGP route attribute. It is transmitted between BGP peers and is not restricted within an AS. The Community attribute allows a group of BGP devices in multiple ASs to share the same routing policies, which simplifies routing policy applications and facilitates routing policy management and maintenance. A BGP device can add or change the community attributes of routes to be advertised.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **route-policy** *route-policy-name* { **deny** | **permit** } **node** *node*

A node is configured for a route-policy, and the view of the route-policy is displayed.

Step 3 (Optional) Configure matching rules for the route-policy to change only the community attributes of the routes that meet the matching rules.

By default, all routes meet matching rules. For details, see **10.7.2 (Optional) Configuring an if-match Clause**.

Step 4 Run either of the following commands to configure the Community attribute.

- Run **apply community** { *community-number* | *aa:nn* | **internet** | **no-advertise** | **no-export** | **no-export-subconfed** } &<1-32> [**additive**]

  Common community attributes are configured for BGP routes.

  NOTE

  This command allows you to configure a maximum of 32 community attributes.

- Run **apply extcommunity** { **rt** { *as-number:nn* | *ipv4-address:nn* } } &<1-16> [**additive**]

  An extended community attribute (route-target) is configured.

  Extended community attributes are extensions to community attributes in services. Currently, only the route-target attribute is supported in VPN.

Step 5 Run **quit**

Return to the system view.

Step 6 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 7 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.

- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 8 Add the Community attribute to routes.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **export**

  The Community attribute is added to the routes advertised to BGP peers or peer groups.
- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **import**

  The Community attribute is added to the routes received from BGP peers or peer groups.
- Run **import-route** *protocol* [*process-id*] **route-policy** *route-policy-name*

  The Community attribute is added to the routes imported by BGP in import mode.
- Run **network** { *ipv4-address* [*mask* | *mask-length*] | *ipv6-address prefix-length* } **route-policy** *route-policy-name*

  The Community attribute is added to the routes imported by BGP in network mode.

NOTE

Step 9 is required only when the Community attribute needs to be added to the routes advertised to BGP peers or peer groups.

Step 9 (Optional) Allow BGP to advertise community attributes when BGP adds community attributes to the routes advertised to BGP peers or peer groups.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **advertise-community**

  BGP is allowed to advertise community attributes to BGP peers or peer groups.

  By default, BGP does not advertise community attributes to any peer or peer group.
- To advertise an extended community attribute to a specified peer or peer group, perform the following steps:
  a. Run the **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **advertise-ext-community** command to advertise an extended community attribute to a specified peer or peer group.
  b. Run the **ext-community-change enable** command to enable the device to change extended community attributes using a routing policy.

     By default, BGP peers cannot change extended community attributes using a route-policy; specifically, BGP peers advertise only the extended community attributes carried in routes to a specified peer or peer group, and the **peer route-policy** command cannot be used to modify the extended community attributes.

----End

9.9.9 (Optional) Configuring a QPPB Policy

Context

QoS Policy Propagation Through the Border Gateway Protocol (QPPB) allows the BGP route sender to classify BGP routes by setting BGP route attributes, and

allows the BGP route receiver to enforce different local QoS policies on the BGP routes based on the attributes set by the BGP route sender.

The following describes QPPB implementation:

- The BGP route sender sets different BGP route attributes for different BGP routes before sending the routes to the BGP route receiver. The attributes include the AS_Path, community, and extended community attributes.
- The BGP route receiver performs the following operations after receiving the BGP routes:
  a. Sets the associated QoS local IDs for the BGP routes that match routing policies based on the attributes in the BGP routes, including the AS_Path, community, and extended community attributes.
  b. Enforces different traffic behaviors based on the associated QoS local IDs during packet forwarding.
  c. Creates a local QPPB policy to configure the associated QoS policy for the BGP routes.
  d. Applies the local QPPB policy to an interface to implement the policy on all the packets that match rules.

NOTE

- The device currently allows a QPPB policy to be applied only to the inbound interface of the destination route.
- Currently, the device supports only the EBGP QPPB policies of the public and private networks and does not support IBGP QPPB policies.
- QPPB policies support the following traffic behaviors: packet filtering, traffic policing, remarking, traffic statistics collection, and congestion management.
- Currently, only IPv4 route forwarding supports QPPB policies, and IPv6 route forwarding does not support QPPB policies.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **qppb local-policy** *policy-name*

A QPPB policy is created.

By default, no QPPB policy is created.

Step 3 Run **qos-local-id** *qos-local-id* **behavior** *behavior*

The QoS local ID carried in a local route is bound to the specified traffic behavior.

By default, the QoS local ID carried in a local route is not bound to any traffic behavior.

NOTE

Before running this command, ensure that the following conditions have been met:

- The **apply qos-local-id***qos-local-id* command has been executed in the route-policy view to configure a QoS local ID for the route requiring QoS control.
- The **traffic behavior** *behavior-name* command has been executed in the system view to create a traffic behavior.

Step 4 Run **quit**

Return to the system view.

Step 5 Run **interface** *interface-type interface-number*

The interface view is displayed.

Step 6 Run **qppb-policy** *policy-name* **enable**

The QPPB policy has been enabled.

By default, no QPPB policy is enabled.

----End

9.9.10 Configuring BGP Load Balancing

Context

On large networks, there may be multiple valid routes to the same destination. BGP, however, advertises only the optimal route to its peers. This may result in unbalanced traffic on different routes. Configuring BGP load balancing enables traffic to be load balanced and network congestion to be reduced.

Equal-cost BGP routes can only be generated for traffic load balancing when the first eight route attributes described in "BGP Route Selection Policies" are the same. Change load balancing rules by adjusting some configurations, for example, ignoring the comparison of the AS_Path attribute. When adjusting these configurations, ensure that these configurations do not result in routing loops.

Local cross routes and routes imported between public network and VPN instances do not support load balancing.

NOTE

If BGP load balancing is configured, the local device changes the next-hop address of routes to its address when advertising routes to IBGP peer groups, regardless of whether the **peer next-hop-local** command is used.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Run **maximum load-balancing** [**ebgp** | **ibgp**] *number* [**ecmp-nexthop-changed**]

The maximum number of BGP routes to be used for load balancing is set.

By default, the maximum number of BGP routes to be used for load balancing is 1, indicating that load balancing is not implemented.

NOTE

- On a public network, if the routes to the same destination implement load balancing, the system will determine the optimal route type. If the optimal routes are IBGP routes, only IBGP routes carry out load balancing. If the optimal routes are EBGP routes, only EBGP routes carry out load balancing. This means that load balancing cannot be implemented among IBGP and EBGP routes with the same destination address.

NOTICE

Configuring BGP not to compare the AS_Path attributes of the routes to be used for load balancing may cause routing loops.

Step 5 (Optional) Run **load-balancing as-path-ignore**

BGP is configured not to compare the AS_Path attributes of the routes to be used for load balancing.

By default, BGP compares the AS_Path attributes of the routes to be used for load balancing.

----End

9.9.11 Verifying the BGP Route Selection and Load Balancing Configuration

Procedure

- Run the **display bgp paths** [*as-regular-expression*] command to check BGP AS_Path information.
- Run the **display bgp routing-table different-origin-as** command to check the routes with the same destination address but different origin ASs.
- Run the **display bgp routing-table regular-expression** *as-regular-expression* command to check information about routes that match the AS regular expression.
- Run the **display bgp routing-table** [*ipv4-address* [{ *mask* | *mask-length* } [**longer-prefixes**]]] command to check routing information in a BGP routing table.
- Run the **display bgp routing-table community** [*community-number* | *aa:nn*] &<1-29> [**internet** | **no-advertise** | **no-export** | **no-export-subconfed**] * [**whole-match**] command to check routing information with the specified BGP community.
- Run the **display bgp routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-*

community-filter-number } command to check information about routes matching a specified BGP community filter.

- Run the **display bgp multicast routing-table** [*ip-address* [*mask-length* [**longer-prefixes**] | *mask* [**longer-prefixes**]]] command to check the MBGP routing table.
- Run the **display bgp multicast routing-table statistics** command to check statistics about the MBGP routing table.

----End

9.10 Controlling the Receiving and Advertisement of BGP Routes

Controlling the receiving and advertisement of BGP routes can reduce the routing table size and improve network security.

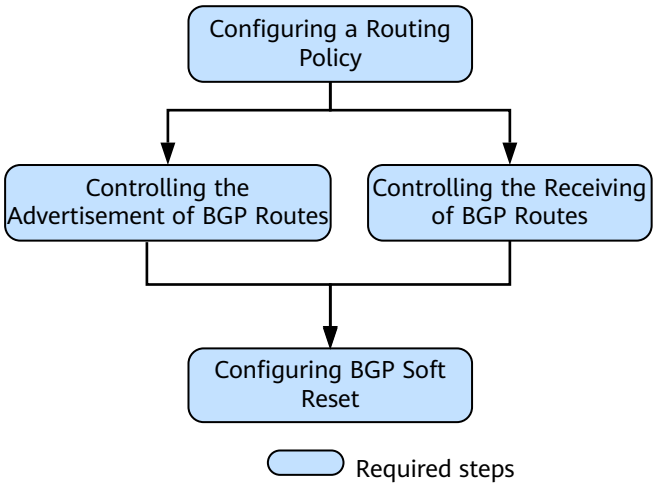
Pre-configuration Tasks

Before controlling the receiving and advertisement of BGP routes, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedure

Figure 9-24 Flowchart of controlling the receiving and advertisement of BGP routes



9.10.1 Configuring a Routing Policy

Context

Before controlling the receiving and advertisement of BGP routes, configure routing policies or filters of routing policies for route selection. For details, see "**10 Routing Policy Configuration**" in the *Huawei AR Series Access Routers Configuration Guide - IP Routing*.

9.10.2 Controlling the Advertisement of BGP Routes

Context

There are usually a large number of routes in a BGP routing table. Transmitting a great deal of routing information brings a heavy load to devices. Routes to be advertised need to be controlled to address this problem. You can configure devices to advertise only routes that these devices want to advertise or routes that their peers require. Multiple routes to the same destination may exist and traverse different ASs. Routes to be advertised need to be filtered in order to direct routes to specific ASs.

Procedure

- Configure a BGP device to advertise routes to all peers or peer groups.

  You can configure a BGP device to filter routes to be advertised.

  a. Run **system-view**

     The system view is displayed.

  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.

  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.

     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.

     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.

  d. Perform either of the following operations to configure the BGP device to advertise routes to all peers or peer groups:

     - To filter routes based on an ACL, run the **filter-policy** { *acl-number* | **acl-name** *acl-name* } **export** [*protocol* [*process-id*]] or the **filter-policy** { *acl6-number* | **acl6-name** *acl6-name* } **export** [*protocol* [*process-id*]] command.

     - To filter routes based on an IP prefix list, run the **filter-policy ip-prefix** *ip-prefix-name* **export** [*protocol* [*process-id*]] or the **filter-policy ipv6-prefix** *ipv6-prefix-name* **export** [*protocol* [*process-id*]] command.

NOTE

If an ACL has been referenced in the **filter-policy** command but no VPN instance is specified in the ACL rule, BGP will filter routes including public and private network routes in all address families. If a VPN instance is specified in the ACL rule, only the data traffic from the VPN instance will be filtered, and no route of this VPN instance will be filtered.

- Configure a BGP device to advertise routes to a specific peer or peer group.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.
  d. Perform any of the following operations to configure the BGP device to advertise routes to a specific peer or peer group:
     - To filter routes based on an ACL, run the **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **filter-policy** { *acl-number* | **acl-name** *acl-name* | *acl6-number* | **acl6-name** *acl6-name* } **export** command.
     - To filter routes based on an IP prefix list, run the **peer** { *ipv4-address* | *group-name* } **ip-prefix** *ip-prefix-name* **export** or the **peer** { *group-name* | *ipv6-address* } **ipv6-prefix** *ipv6-prefix-name* **export** command.
     - To filter routes based on an AS_Path filter, run the **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **as-path-filter** { *as-path-filter-number* | *as-path-filter-name* } **export** command.
     - To filter routes based on a route-policy, run the **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **export** command.

     NOTE

     The routing policy applied in the **peer route-policy export** command does not support a specific interface as one matching rule. That is, the routing policy does not support the **if-match interface** command.

----End

9.10.3 Controlling the Receiving of BGP Routes

Context

When a BGP device is attacked or network configuration errors occur, the BGP device will receive a large number of routes from its neighbor. As a result, many

device resources are consumed. Therefore, the administrator must limit the resources used by the device based on network planning and device capacity. BGP provides peer-based route control to limit the number of routes to be sent by a neighbor. This addresses the preceding problem.

Procedure

- Configure a BGP device to receive routes from all its peers or peer groups.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.
  d. Perform either of the following operations to configure the BGP device to filter the routes received from all its peers or peer groups:
     - To filter routes based on an ACL, run the **filter-policy** { *acl-number* | **acl-name** *acl-name* } **import** or the **filter-policy** { *acl6-number* | **acl6-name** *acl6-name* } **import** command.
     - To filter routes based on an IP prefix list, run the **filter-policy ip-prefix** *ip-prefix-name* **import** or the **filter-policy ipv6-prefix** *ipv6-prefix-name* **import** command.

     NOTE

     If an ACL has been referenced in the **filter-policy** command but no VPN instance is specified in the ACL rule, BGP will filter routes including public and private network routes in all address families. If a VPN instance is specified in the ACL rule, only the data traffic from the VPN instance will be filtered, and no route of this VPN instance will be filtered.
- Configure a BGP device to receive routes from a specific peer or peer group.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

The IPv6 address family view is displayed.

d. Perform any of the following operations to configure the BGP device to filter the routes received from a specific peer or peer group:

- To filter routes based on an ACL, run the **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **filter-policy** { *acl-number* | **acl-name** *acl-name* | *acl6-number* | **acl6-name** *acl6-name* } **import** command.
- To filter routes based on an IP prefix list, run the **peer** { *ipv4-address* | *group-name* } **ip-prefix** *ip-prefix-name* **import** or the **peer** { *group-name* | *ipv6-address* } **ipv6-prefix** *ipv6-prefix-name* **import** command.
- To filter routes based on an AS_Path filter, run the **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **as-path-filter** { *as-path-filter-number* | *as-path-filter-name* } **import** command.
- To filter routes based on a route-policy, run the **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-policy** *route-policy-name* **import** command.

NOTE

The routing policy applied in the **peer route-policy import** command does not support a specific interface as one matching rule. That is, the routing policy does not support the **if-match interface** command.

NOTICE

If the number of routes received by the local device exceeds the upper limit and the **peer route-limit** command is used for the first time, the local device and its peer reestablish the peer relationship, regardless of whether **alert-only** is set.

e. (Optional) Run **peer** { *group-name* | *ipv4-address* } **route-limit** *limit* [*percentage*] [**alert-only** | **idle-forever** | **idle-timeout** *times*]

The maximum number of routes that can be received from the peer or peer group is set.

----End

9.10.4 Configuring BGP Soft Reset

Context

After changing a BGP import policy, you must reset BGP connections for the new import policy to take effect. This, however, interrupts these BGP connections temporarily. BGP route-refresh allows the system to softly reset BGP connections to refresh a BGP routing table without tearing down any BGP connection. If a device's peer does not support route-refresh, configure the device to remain all routing updates received from the peer so that the device can refresh its routing table without tearing down the BGP connection with the peer.

Procedure

- If a device's peer supports route-refresh, configure the device to softly reset the BGP connection with the peer and update the BGP routing table.
  a. Run **system-view**
     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }
     The BGP view is displayed.
  c. (Optional) Run **peer** { *ipv4-address* | *group-name* } **capability-advertise route-refreshpeer** *ipv6-address* **capability-advertise** { **4-byte-as** | **route-refresh** }or run:
     Route-refresh is enabled.
     By default, route-refresh is enabled.
  d. Run **quit**
     Return to the system view.
  e. Run **quit**
     Return to the user view.
  f. Run **refresh bgp** [**vpn-instance** *vpn-instance-name* **ipv4-family** | **vpnv4**] { **all** | *ipv4-address* | **group** *group-name* | **external** | **internal** } { **export** | **import** }
     or **refresh bgp ipv6** { **all** | **group** *group-name* | *ipv6-address* | **external** | **internal** } { **export** | **import** }
     BGP soft reset is configured.
- If a device's peer does not support route-refresh, configure the device to remain all routing updates received from the peer so that the device can refresh its routing table without tearing down the BGP connection with the peer.
  a. Run **system-view**
     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }
     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }
       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]
       The IPv6 address family view is displayed.

NOTICE

If the **peer keep-all-routes** command is used on the device for the first time, the sessions between the device and its peers are reestablished.

The **refresh bgp** command takes effect when the **peer keep-all-routes** command is used on the device supporting route-refresh.

d. Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **keep-all-routes**

The device is configured to store all the routing updates received from its peers or peer groups.

By default, the device stores only the routing updates that are received from peers or peer groups and match a configured import policy.

----End

9.10.5 Verifying the BGP Route Receiving and Advertisement Control Configuration

Procedure

- Run the **display ip as-path-filter** [*as-path-filter-number* | *as-path-filter-name*] command to check information about a configured AS_Path filter.
- Run the **display ip community-filter** [*basic-comm-filter-num* | *adv-comm-filter-num* | *comm-filter-name*] command to check information about a configured community filter.
- Run the **display ip extcommunity-filter** [*basic-extcomm-filter-num* | *advanced-extcomm-filter-num* | *extcomm-filter-name*] command to check information about a configured extcommunity filter.
- Run the **display bgp routing-table as-path-filter** { *as-path-filter-number* | *as-path-filter-name* } command to check information about 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 information about routes matching a specified BGP community filter.
- Run the **display bgp routing-table peer** *ipv4-address* **received-routes** [**active**] [**statistics**] command to check information about routes received by a BGP device from its peers.
- Run the **display bgp multicast routing-table different-origin-as** command to check information about MBGP routes with different origin ASs.
- Run the **display bgp multicast routing-table regular-expression** *as-regular-expression* to check information about MBGP routes matching the AS regular expression.
- Run the **display bgp multicast paths** [*as-regular-expression*] command to check information about AS paths.
- Run the **display bgp multicast routing-table as-path-filter** { *as-path-filter-number* | *as-path-filter-name* } command to check information about MBGP routes matching the AS_Path filter.
- Run the **display bgp multicast routing-table community-filter** { { *community-filter-name* | *basic-community-filter-number* } [**whole-match**] | *advanced-community-filter-number* } command to check information about routes matching a specified MBGP community filter.
- Run the **display bgp multicast routing-table peer** *peer-address* { **advertised-routes** [*network* [{ *mask* | *mask-length* } [**longer-prefixes**]]] | **received-routes** [**active**] | **accepted-routes** } command to

check information about routes that are sent by and received from the specified MBGP peer.

- Run the **display bgp multicast network** command to check the routing information that MBGP advertises.

----End

9.11 Adjusting the BGP Network Convergence Speed

You can configure BGP timers, disable rapid EBGP connection reset, and configure BGP route dampening to speed up BGP network convergence and improve BGP security.

Pre-configuration Tasks

Before configuring adjusting the BGP network convergence speed, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedure

You can perform the following configuration tasks as required. The following configuration tasks (excluding the task of Verifying the BGP Network Convergence Speed Adjustment Configuration) can be performed in any sequence.

9.11.1 Configuring a BGP ConnectRetry Timer

Context

After BGP initiates a TCP connection, the ConnectRetry timer will be stopped if the TCP connection is established successfully. If the first attempt to establish a TCP connection fails, BGP tries again to establish the TCP connection after the ConnectRetry timer expires.

- Setting a short ConnectRetry interval reduces the period BGP waits between attempts to establish a TCP connection. This speeds up the establishment of the TCP connection.
- Setting a long ConnectRetry interval suppresses routing flapping caused by peer relationship flapping.

A ConnectRetry timer can be configured either for all peers or peer groups, or for a specific peer or peer group. A ConnectRetry timer configured for a specific peer takes precedence over that configured for the peer group of this peer. In addition, a ConnectRetry timer configured for a specific peer or peer group takes precedence over that configured for all peers or peer groups.

Procedure

- Configure a BGP ConnectRetry timer for all peers or peer groups.
  a. Run **system-view**

     The system view is displayed.

b. Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

c. Run **timer connect-retry** *connect-retry-time*

A BGP ConnectRetry timer is configured for all peers or peer groups.

By default, the ConnectRetry timer value is 32s.

- Configure a ConnectRetry timer for a specific peer or peer group.
  a. Run **system-view**

  The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

  The BGP view is displayed.
  c. Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **timer connect-retry** *connect-retry-time*

  A ConnectRetry timer is configured for a specific peer or peer group.

  By default, the ConnectRetry timer value is 32s.

----End

9.11.2 Configuring BGP Keepalive and Hold Timers

Context

Keepalive messages are used by BGP to maintain peer relationships.

- If short Keepalive time and holdtime are set, BGP can detect a link fault quickly. This speeds up BGP network convergence, but increases the number of Keepalive messages on the network and loads of devices, and consumes more network bandwidth resources.
- If long Keepalive time and holdtime are set, the number of Keepalive messages on the network is reduced, loads of devices are reduced, and fewer network bandwidths are consumed. If the Keepalive time is too long, BGP is unable to detect link status changes in a timely manner. This is unhelpful for implementing rapid BGP network convergence and may cause many packets to be lost.

Keepalive and hold timers can be configured either for all peers or peer groups, or for a specific peer or peer group. Keepalive and hold timers configured for a specific peer take precedence over those configured for the peer group of this peer. In addition, Keepalive and hold timers configured for a specific peer or peer group take precedence over those configured for all peers or peer groups.

NOTICE

Changing timer values using the **timer** command or the **peer timer** command interrupts BGP peer relationships between routers.

Setting the Keepalive time to 20s is recommended. If the Keepalive time is smaller than 20s, sessions between peers may be closed.

Procedure

- Configure BGP timers for all peers or peer groups.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Run **timer keepalive** *keepalive-time* **hold** *hold-time* [**min-holdtime** *min-holdtime*]

     BGP timers are configured.

     The proper maximum interval at which Keepalive messages are sent is one third the holdtime. By default, the Keepalive time is 60s and the holdtime is 180s.
- Configure BGP timers for a specific peer or peer group.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **timer keepalive** *keepalive-time* **hold** *hold-time* [**min-holdtime** *min-holdtime*]

     The Keepalive and hold timers are configured for a specific peer or peer group.

     The proper maximum interval at which Keepalive messages are sent is one third the holdtime. By default, the Keepalive time is 60s and the holdtime is 180s.

----End

9.11.3 Configuring an Update Message Timer

Context

BGP does not periodically update a routing table. When BGP routes change, BGP updates the changed BGP routes in the BGP routing table by sending Update messages.

- If a short Update message interval is set, BGP can fast detect route changes. This speeds up BGP network convergence, but increases the number of Update messages on the network and loads of devices, and consumes more network bandwidth resources.
- If a long Update message interval is set, the number of Update messages on the network is reduced, loads of devices are reduced, and fewer network bandwidths are consumed. This avoids network flapping. If the Update message interval is too long, BGP is unable to detect route changes in a timely manner. This is unhelpful for implementing rapid BGP network convergence and may cause many packets to be lost.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **route-update-interval** *interval*

An Update message timer is configured.

By default, the interval at which Update messages are sent to IBGP peers is 15s, and the interval at which Update messages are sent to EBGP peers is 30s.

Step 5 (Optional) Configure a delay in sending Update messages.

- Run **peer** { *ipv4-address* | *group-name* | *ipv6-address* } **out-delay** *delay-value*

  A delay in sending Update messages is set.

  The default delay is 0, indicating that Update packets are sent without a delay.
- Run **out-delay** *delay-value*

  A global delay in sending Update messages is set.

  The default delay value is 0, indicating that the intermediate device on the primary path sends Update packets without a delay.

----End

9.11.4 Disabling Rapid EBGP Connection Reset

Context

Rapid EBGP connection reset is enabled by default. This allows BGP to immediately respond to a fault on an interface and delete the direct EBGP sessions on the interface without waiting for the hold timer to expire and implements rapid BGP network convergence.

If the status of an interface used to establish an EBGP connection changes frequently, the EBGP session will be deleted and reestablished repeatedly, causing network flapping. Rapid EBGP connection reset can be disabled in such a situation. BGP will delete direct EBGP sessions on the interface until the hold timer expires. This suppresses BGP network flapping, helps implement rapid BGP network convergence, and reduces network bandwidth consumption.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **undo ebgp-interface-sensitive**

Rapid EBGP connection reset is disabled.

By default, rapid EBGP connection reset is enabled.

NOTE

Rapid EBGP connection reset enables BGP to quickly respond to interface faults but does not enable BGP to quickly respond to interface recovery. After the interface recovers, BGP uses its state machine to restore relevant sessions.

Rapid EBGP connection reset is disabled in a situation where the status of an interface used to establish an EBGP connection changes frequently. If the status of the interface becomes stable, run the **ebgp-interface-sensitive** command to enable rapid EBGP connection reset to implement rapid BGP network convergence.

----End

9.11.5 Configuring the BGP Next Hop Delayed Response

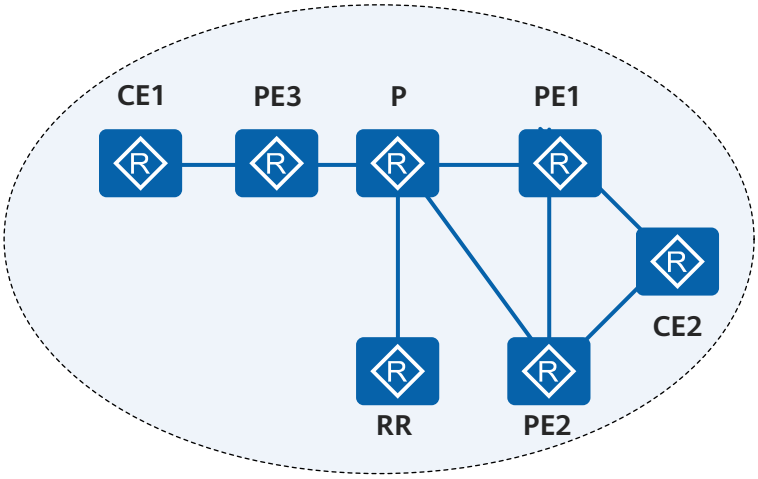
Context

Configuring the BGP next hop delayed response can speed up BGP route convergence and minimize traffic loss.

As shown in **Figure 9-25**, PE1, PE2, and PE3 are the clients of the RR. CE2 is dual-homed to PE1 and PE2. PE1 and PE2 advertise their routes to CE2 to the RR. The RR advertises the route from PE1 to PE3. PE3 has a route to CE2 only and advertises this route to CE1. After the route exchange, CE1 and CE2 can communicate. If PE1 fails, PE3 detects that the next hop is unreachable and instructs CE1 to delete the route to CE2. Traffic is interrupted. After BGP route convergence is complete, the RR selects the route advertised by PE2 and sends a route update message to PE3. PE3 then advertises this route to CE1, and traffic forwarding is restored to the normal state. A high volume of traffic will be lost during traffic interruption because BGP route convergence is rather slow.

If the BGP next hop delayed response is enabled on PE3, PE3 does not reselect a route or instruct CE1 to delete the route to CE2 immediately after detecting that the route to PE1 is unreachable. After BGP convergence is complete, the RR selects the route advertised by PE2 and sends the route to PE3. PE3 then reselects a route and sends a route update message to CE1. Traffic forwarding is restored to the normal state. After the BGP next hop delayed response is enabled on PE3, PE3 does not need to delete the route or instruct CE1 to delete the route. This delayed response speeds up BGP route convergence and minimizes traffic loss.

Figure 9-25 Networking diagram for configuring the BGP next hop delayed response



The BGP next hop delayed response applies to a scenario where the next hop has multiple links to reach the same destination. If there is only one link between the next hop and the destination, configuring the BGP next hop delayed response may cause heavier traffic loss when the link fails because link switching is impossible.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **nexthop recursive-lookup delay** [*delay-time*]

A delay in responding to a next hop change is set.

By default, the delay in responding to changes of the next hop is not configured.

NOTE

BGP route convergence depends on IGP route convergence. If IGP route convergence is quick, the default delay time does not need to be changed. If IGP route convergence is slow, setting a delay time longer than IGP route convergence time is recommended.

----End

9.11.6 Configuring BGP Route Dampening

Context

A route is considered to be flapping when it repeatedly appears and then disappears in the routing table. BGP generally applies to complex networks where routes change frequently. Frequent route flapping consumes lots of bandwidths

and CPU resources and even affects normal network operation. BGP route dampening prevents frequent route flapping.

BGP can differentiate routes based on policies and use different route dampening parameters to suppress different routes. For example, on a network, you can set a long suppression time for routes with a long mask and set a short suppression time for routes with a short mask (such as 8-bit mask).

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** | **vpnv4** [**unicast**] | **vpn-instance** *vpn-instance-name* }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast** | **vpn-instance** *vpn-instance-name*]

  The IPv6 address family view is displayed.

Step 4 Run **dampening** [**ibgp**] [*half-life-reach reuse suppress ceiling* | **route-policy** *route-policy-name*] *

BGP route dampening parameters are configured.

NOTE

The **dampening** command is valid only for EBGP routes.

The **dampening ibgp** command is valid only for BGP VPNv4 routes.

----End

9.11.7 Verifying the BGP Network Convergence Speed Adjustment Configuration

Procedure

- Run the **display bgp peer** [**verbose**] command to check information about all BGP peers.
- Run the **display bgp group** [*group-name*] command to check information about the specified BGP peer group.
- Run the **display bgp routing-table dampened** command to check dampened BGP routes.
- Run the **display bgp routing-table dampening parameter** command to check configured BGP route dampening parameters.
- 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 route flapping statistics.

- Run the **display bgp multicast routing-table dampened** command to check dampened MBGP routes.
- Run the **display bgp multicast routing-table dampening parameter** command to check MBGP route dampening parameters.
- Run the following commands to check statistics about flapping MBGP routes.
  - **display bgp multicast routing-table flap-info** [*ip-address* [*mask* [**longer-match**] | *mask-length* [**longer-match**]] | **as-path-filter** { *as-path-filter-number* | *as-path-filter-name* }]
  - **display bgp multicast routing-table flap-info regular-expression** *as-regular-expression*

----End

9.12 Configuring BGP Reliability

You can configure BGP Tracking, association between BGP and BFD, and BGP GR to speed up BGP network convergence and improve BGP reliability.

Pre-configuration Tasks

Before configuring BGP reliability, complete the following task:

- **Configuring Basic BGP Functions**

Configuration Procedure

You can perform the following configuration tasks as required. The following configuration tasks can be performed in any sequence.

9.12.1 Enabling BGP Tracking

Context

BFD can be configured to detect peer relationship status changes in order to implement rapid BGP convergence. BFD, however, needs to be configured on the entire network, and has poor extensibility. If BFD cannot be deployed on a device to detect BGP peer relationship status, BGP peer tracking can be enabled on the device to quickly detect link or peer unreachability, implementing rapid network convergence.

BGP tracking can be used to adjust the interval between peer unreachability discovery and connection interruption. This suppresses BGP peer relationship flapping caused by route flapping and improves BGP network stability.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **tracking** [**delay** *delay-time*]

BGP peer tracking is enabled on the device to detect the status of a specified peer.

By default, BGP peer tracking is disabled.

----End

9.12.2 Configuring BFD for BGP

Context

BGP periodically sends Keepalive messages to its peers to detect the status of its peers. It takes more than 1 second for this detection mechanism to detect a fault. When data is transmitted at gigabit rates, long-time fault detection will cause packet loss. This cannot meet high reliability requirements of carrier-class networks. Association between BGP and BFD can solve this problem. BFD is a millisecond-level fault detection mechanism. It can detect faults on the link between BGP peers within 50 ms. Therefore, BFD can speed up BGP route convergence, ensures fast link switching, and reduces traffic loss.

When a peer joins a peer group on which BFD is enabled, BFD also takes effect on the peer and a BFD session is created on the peer. To prevent BFD from taking effect on the peer, run the **peer bfd block** command.

By default, Huawei devices establish multi-hop IBGP sessions with each other. When a Huawei device communicates with a non-Huawei device that establishes a single-hop IBGP session by default, you are advised to configure only association between IGP and BFD or association between IBGP and BFD.

NOTE

BFD for routing protocols can only be configured on GRE tunnel interfaces.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bfd**

Global BFD is enabled on the local device.

Step 3 Run **quit**

Return to the system view.

Step 4 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 5 (Optional) The BGP-VPN instance address family view is displayed.

- Run **ipv4-family vpn-instance** *vpn-instance-name*

  The BGP-VPN instance IPv4 address family view is displayed.
- Run **ipv6-family vpn-instance** *vpn-instance-name*

  The BGP-VPN instance IPv6 address family view is displayed.

NOTE

BFD for BGP can be configured for the VPN in this view. To configure BFD for BGP for the public network, skip this step.

Step 6 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **bfd enable** [**single-hop-prefer**]

BFD is configured for the peer or peer group, and default BFD parameters are used to establish BFD sessions.

If BFD is configured for a peer group, BFD sessions are created for the peers on which the **peer bfd block** command is not used.

Step 7 Run **peer** { *group-name* | *ipv4-address* | *ipv6-address* } **bfd** { **min-tx-interval** *min-tx-interval* | **min-rx-interval** *min-rx-interval* | **detect-multiplier** *multiplier* | **wtr** *wtr-value* } *

BFD session parameters are configured.

Step 8 (Optional) Run **peer** { *ipv4-address* | *ipv6-address* } **bfd block**

The peer is disabled from inheriting the BFD function of the peer group to which the peer belongs.

NOTE

- BFD sessions are established when they are in Established state.
- If BFD parameters are configured on a peer, BFD sessions are established using these parameters.
- The **peer** { *ipv4-address* | *ipv6-address* } **bfd block** and **peer** { *ipv4-address* | *ipv6-address* } **bfd enable** commands are mutually exclusive.

----End

Verifying the Configuration

- Run the **display bgp bfd session** { [**vpnv4 vpn-instance** *vpn-instance-name*] **peer** *ipv4-address* | **all** } command to check information about the BFD sessions established between BGP peers.
- Run the **display bgp** [**vpnv4 vpn-instance** *vpn-instance-name*] **peer** [[*ipv4-address*] **verbose**] command to check information about BGP peers.
- Run the **display bgp group** [*group-name*] command to check information about the specified BGP peer group.
- Run the **display bgp vpnv4** { **all** | **vpn-instance** *vpn-instance-name* } **group** [*group-name*] command to check information about the BGP VPNv4 peer group.
- Run the **display bgp ipv6 bfd session** { [**vpnv6 vpn-instance** *vpn-instance-name*] **peer** *ipv6-address* | **all** } command to check information about the BFD sessions established between BGP peers.

9.12.3 Configuring the BGP GR Function

Context

BGP restart causes peer relationships reestablishment and traffic interruption. Graceful restart (GR) ensures uninterrupted traffic forwarding in the case of BGP restart.

NOTE

In practical application, in order to realize that business forwarding is not affected by motherboard failure, it is usually possible to configure BGP GR in the hardware environment of dual motherboard to make sense.

All the models support the GR Helper, and only AR3200 series support the GR Restarter.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **graceful-restart**

BGP GR is enabled.

By default, BGP GR is disabled.

Step 4 (Optional) Run **graceful-restart timer wait-for-rib** *timer*

The time during which the restarting speaker and receiving speaker wait for End-of-RIB messages is set.

By default, the time for waiting for End-of-RIB messages is 600 seconds.

Step 5 (Optional) Run **graceful-restart peer-reset**

The device is enabled to reset a BGP session in GR mode.

By default, a device is not enabled to reset a BGP connection in GR mode.

----End

Verifying the Configuration

- Run the **display bgp peer verbose** command to check detailed information about BGP GR.

9.13 Configuring BGP Route Summarization

On IPv4 networks, BGP supports automatic route summarization and manual route summarization. Manual route summarization takes precedence over automatic route summarization. On IPv6 networks, BGP supports only manual route summarization.

Pre-configuration Tasks

Before configuring BGP route summarization, complete the following task:

- **Configuring Basic BGP Functions**

Procedure

- Configure automatic route summarization.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Run **ipv4-family** { **unicast** | **multicast** }

     The IPv4 address family view is displayed.
  d. Run **summary automatic**

     BGP summarizes subnet routes based on natural mask.

     NOTE

     The command summarizes the routes imported by 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 using the **network** command.
- Configure manual route summarization.
  a. Run **system-view**

     The system view is displayed.
  b. Run **bgp** { *as-number-plain* | *as-number-dot* }

     The BGP view is displayed.
  c. Enter the corresponding address family view based on network type to configure BGP devices on networks.
     - Run **ipv4-family** { **unicast** | **multicast** }

       The IPv4 address family view is displayed.
     - Run **ipv6-family** [**unicast**]

       The IPv6 address family view is displayed.
  d. Perform any of the following operations to configure manual route summarization.
     - To advertise the summarized routes and specific routes, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } or the **aggregate** *ipv6-address prefix-length* command.
     - To advertise only the summarized routes, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **detail-suppressed** or the **aggregate** *ipv6-address prefix-length* **detail-suppressed** command.
     - To advertise the summarized routes and specific routes that meet the specified route-policy, run the **aggregate** *ipv4-address* { *mask* |

mask-length } **suppress-policy** *route-policy-name* or the **aggregate** *ipv6-address prefix-length* **suppress-policy** *route-policy-name* command.

- To advertise the summarized routes of which the AS_Set attribute helps detect routing loops, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **as-set** or the **aggregate** *ipv6-address prefix-length* **as-set** command.
- To set attributes for the summarized routes, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **attribute-policy** *route-policy-name* or the **aggregate** *ipv6-address prefix-length* **attribute-policy** *route-policy-name* command.
- To summarize the specific routes that meet the specified route-policy, run the **aggregate** *ipv4-address* { *mask* | *mask-length* } **origin-policy** *route-policy-name* or the **aggregate** *ipv6-address prefix-length* **origin-policy** *route-policy-name* command.

NOTE

Manual route summarization is valid for the routes in the local BGP routing table. For example, if the local BGP routing table does not contain routes with mask longer than 16 bits, such as 10.1.1.1/24, BGP will not generate an aggregated route for it even if the **aggregate 10.1.1.1 16** command is used.

----End

Verifying the Configuration

- Run the **display bgp routing-table** [*ipv4-address* [{ *mask* | *mask-length* } [**longer-prefixes**]]] command to check information about summarized routes.
- Run the **display bgp multicast routing-table** [*ip-address* [*mask-length* [**longer-prefixes**] | *mask* [**longer-prefixes**]]] command to check the MBGP routing table.

9.14 Configuring On-demand Route Advertisement

If a BGP device only wants to received required routes but its peer cannot maintain different export policies for connected devices, you can configure prefix-based BGP outbound route filtering (ORF) to meet this requirement.

Pre-configuration Tasks

Before configuring prefix-based BGP ORF, complete the following tasks:

- **Configuring Basic BGP Functions**
- **Configuring an IP Prefix List**

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **ipv4-family unicast**

The IPv4 unicast address family view is displayed.

Step 4 Run **peer** { *group-name* | *ipv4-address* } **ip-prefix** *ip-prefix-name* **import**

A prefix-based import policy is configured for a peer or peer group.

Step 5 Run **peer** { *group-name* | *ipv4-address* } **capability-advertise orf** [**non-standard-compatible**] **ip-prefix** { **both** | **receive** | **send** }

Prefix-based ORF is enabled for a peer or peer group.

By default, prefix-based ORF is disabled for a peer or peer group.

----End

Verifying the 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 ip-prefix** command to check prefix-based BGP ORF information received from a specified peer.

9.15 Configuring BGP to Advertise Default Routes to Peers

If a BGP device needs to send multiple routes to its peer, the BGP device can be configured to send only a default route with the local address as the next-hop address to its peer, regardless of whether there are default routes in the local routing table. This function reduces the number of network routes and saves memory and network resources.

Pre-configuration Tasks

Before configuring BGP to send default routes to peers, 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-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family** { **unicast** | **multicast** }

  The IPv4 address family view is displayed.
- Run **ipv6-family** [**unicast**]

  The IPv6 address family view is displayed.

Step 4 Run **peer** { *group-name* | *ipv4-address* | *ipv6-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>]

A BGP device is configured to send default routes to a peer or peer group.

NOTE

The **conditional-route-match-all** and **conditional-route-match-any** keywords are not supported in the IPv4 multicast address family view and the IPv6 address family view.

----End

Verifying the Configuration

- Run the **display bgp routing-table** [*ipv4-address* [*mask* | *mask-length* [**longer-prefixes**]]] command to check received BGP default routes.
- Run the **display bgp multicast routing-table** [*ip-address* [*mask-length* [**longer-prefixes**] | *mask* [**longer-prefixes**]]] command to check received MBGP default routes.

9.16 Configuring Path MTU Auto Discovery

BGP path maximum transmission unit (MTU) auto discovery can discover the minimum MTU (path MTU) on the network path from the source to the destination so that TCP can transmit BGP messages based on the path MTU.

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-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **peer** { *group-name* | *ipv4-address* } **path-mtu auto-discovery**

Path MTU auto discovery is enabled.

By default, path MTU auto discovery is disabled.

NOTE

The transmit and receive paths between two BGP peers may be different. Therefore, you are advised to run this command on both ends so that the two BGP peers can exchange messages based on the path MTU.

----End

Verifying the Configuration

- Run the **display bgp peer** [*ipv4-address*] **verbose** command to check whether path MTU auto discovery has been successfully configured.

9.17 Configuring MP-BGP

Multiprotocol BGP (MP-BGP) enables BGP to support IPv4 unicast networks, IPv4 multicast networks, and IPv6 unicast networks.

Pre-configuration Tasks

Before configuring MP-BGP, complete the following task:

- **9.6.1 Starting a BGP Process**

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

BGP is started, the local AS number is specified, and the BGP view is displayed.

Step 3 Enter the corresponding address family view based on network type to configure BGP devices on networks.

- Run **ipv4-family unicast**

  The BGP-IPv4 unicast address family view is displayed.
- Run **ipv4-family vpnv4**

  The BGP-VPNv4 address family view is displayed.
- Run **ipv4-family vpn-instance** *vpn-instance-name*

  The BGP-VPN instance IPv4 address family view is displayed.
- Run **ipv4-family multicast**

  The BGP-IPv4 multicast address family view is displayed.
- Run **ipv6-family unicast**

  The BGP-IPv6 unicast address family view is displayed.

NOTE

- Different extended BGP functions must be configured in their respective address family views, while common BGP functions are configured in the BGP view.
- The Router supports the following MBGP features: basic BGP functions, BGP security (MD5 authentication and keychain authentication), simplifying IBGP network connections (route reflector and confederation), BGP route selection and load balancing, controlling the receiving and advertisement of BGP routes, adjusting the BGP network convergence speed, BGP reliability, BGP route summarization, path MTU auto discovery, and advertising default routes to peers.
- Some BGP4+ functions can be configured in the BGP view, and some BGP4+ functions need to be configured in the IPv6 unicast address family view. For example, the following BGP4+ functions need to be configured in the IPv6 unicast address family view: load balancing, manual route summarization, route dampening, community, and route reflector.

----End

9.18 Configuring the Dynamic BGP Peer Function

Usage Scenario

If static BGP peers change frequently, the local device needs to add or delete BGP peer configurations in response to each change, which requires a heavy maintenance workload. To address this problem, configure the dynamic BGP peer function, which allows BGP to listen to BGP connection requests from a specified network segment, establish BGP peer relationships dynamically, and add the peers to a peer group. This spares the local device from adding or deleting BGP peer configurations in response to each change in the peer number, which reduces the maintenance workload.

Pre-configuration Tasks

Before configuring the dynamic BGP peer function, **configure basic BGP functions**.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 (Optional) Run **bgp dynamic-session-limit** *limit-value*

The maximum number of dynamic BGP peer sessions is configured.

If a large number of dynamic BGP peer sessions are established on the network segment, excessive system resources will be consumed. To prevent this problem, configure a maximum number for dynamic BGP peer sessions as required.

By default, the maximum number of dynamic BGP peer sessions is half of the total specification.

Step 4 (Optional) Run **ipv4-family vpn-instance** *vpn-instance-name*

The BGP-VPN instance IPv4 address family view is displayed.

Perform this step if you need to configure the dynamic BGP peer function in the BGP-VPN instance IPv4 address family view in a BGP/MPLS IP VPN scenario.

Step 5 Run **group** *group-name* [**external** | **internal**]

A BGP peer group is created.

Configure the following parameters as required:

- If the local device and its peers reside in the same AS, configure **internal** to create an IBGP peer group.
- If the local device and its peers reside in different ASs, configure **external** to create an EBGP peer group.

If neither **internal** nor **external** is configured, an IBGP peer group is created by default.

Step 6 Run **peer** *group-name* **listen-net** *network* { *mask* | *mask-length* }

BGP is configured to listen to BGP connection requests from a specified network segment and establish BGP peer relationships dynamically.

If you run the command multiple times, BGP listens to BGP connection requests from multiple network segments.

Step 7 Run **peer** *group-name* **as-number** { *as-number-plain* | *as-number-dot* } [**optional-as** { *optional-as-number-plain* | *optional-as-number-dot* } &<1-5>]

An AS number is configured for the peer group.

Configure the following parameters as required:

- If the dynamic peers in the peer group reside in the same AS, configure { *as-number-plain* | *as-number-dot* } to set a fixed AS number.
- If the dynamic peers in the peer group may reside in different ASs, in addition to a fixed AS number, you need to configure **optional-as** { *optional-as-number-plain* | *optional-as-number-dot* } &<1-5> to set an optional AS number. A maximum of five optional AS numbers can be set.

----End

Verifying the Configuration

After configuring the dynamic BGP peer function, check the configuration.

- Run the **display bgp** [**vpnv4** { **all** | **vpn-instance** *vpn-instance-name* }] **peer** [[*ipv4-address*] **verbose**] command to check BGP peer information.
- Run the **display bgp** [**vpnv4** { **all** | **vpn-instance** *vpn-instance-name* }] **group** [*group-name*] command to check BGP peer group information.

Display BGP peer information. The command output shows dynamic peer information.

```
<Huawei> display bgp peer
 Status codes: * - Dynamic
 BGP local router ID : 1.2.3.4
```



```
Local AS number : 10
Total number of peers : 2                 Peers in established state : 1
Total number of dynamic peers : 1
 Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv

 1.1.1.1         4   100        0        0     0 00:00:07        Idle       0
*1.2.5.6         4   200       32       35     0 00:17:49 Established       0
```

Display BGP peer group information. The command output shows dynamic peer information and the network segment from which BGP listens to BGP connection requests.

```
<Huawei> display bgp group my-peer
 BGP peer-group: my-peer
 Remote AS: 100

       listen-net: 10.1.1.0 24

 Authentication type configured: None
 Group's BFD has been enabled
 Type : internal
 Maximum allowed route limit: 100
 Threshold: 75%
 Configured hold timer value: 180
 Keepalive timer value: 60
 Connect-retry timer value: 32
 Minimum route advertisement interval is 15 seconds
 PeerSession Members:
   10.1.1.2

       Status codes: * - Dynamic

 Peer Preferred Value: 0
 No routing policy is configured
 Peer Members:
  Peer            V         AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
 *10.1.1.2        4        100       35       42     0 00:29:01 Established  0
```

9.19 Maintaining BGP

9.19.1 Configuring Alarm and Clear Alarm Thresholds for the Number of BGP Routes

Alarm and clear alarm thresholds for the number of BGP routes facilitate maintenance.

Context

The number of BGP routes that can be added to a routing table is limited. If the number exceeds a limit, new routes cannot be added to the routing table, which may interrupt services. To address this problem, configure alarm and clear alarm thresholds for the number of BGP routes. With the alarm and clear alarm thresholds, alarms are generated and cleared as expected. The alarms prompt you to check whether an exception occurs and to take preventive measures. You can configure the alarm and clear alarm thresholds as required.

Procedure

Step 1 Run **system-view**

The system view is displayed.

Step 2 Run **bgp** { *as-number-plain* | *as-number-dot* }

The BGP view is displayed.

Step 3 Run **routing-table limit threshold-alarm upper-limit** *upper-limit-value* **lower-limit** *lower-limit-value*

Alarm and clear alarm thresholds are configured for the number of BGP routes.

- *upper-limit-value* specifies the alarm threshold. If the ratio of BGP routes to the maximum number that is allowed exceeds the alarm threshold, an alarm is generated.
- *lower-limit-value* specifies the clear alarm threshold. If the ratio of BGP routes to the maximum number that is allowed falls below this threshold, the alarm is cleared.
- *upper-limit-value* must be greater than *lower-limit-value*; otherwise, alarms are generated and cleared repeatedly if route flapping occurs.

By default, *upper-limit-value* is 80%, and *lower-limit-value* is 70%.

----End

9.19.2 Resetting BGP Connections

Context

NOTICE

Running the **reset bgp** command to reset BGP connections will interrupt BGP peer relationships between BGP devices. Exercise caution when you use this command.

When the BGP routing policy changes, for example, the router does not support the route-refresh capability, reset BGP connections to make the modification take effect.

Procedure

- To reset all BGP connections, run the **reset bgp all** command in the user view.
- To reset the BGP connection with a specified AS, run the **reset bgp** { *as-number-plain* | *as-number-dot* } command in the user view.
- To reset the BGP connection with a specified peer, run the **reset bgp** *ipv4-address* command in the user view.
- To reset all EBGP connections, run the **reset bgp external** command in the user view.
- To reset the BGP connection with a specified peer group, run the **reset bgp group** *group-name* command in the user view.

- To reset all IBGP connections, run the **reset bgp internal** command in the user view.
- To reset the MBGP connection with a specified peer, run the **reset bgp multicast** *peer-address* command in the user view.
- To reset all MBGP connections, run the **reset bgp multicast all** command in the user view.
- To reset the MBGP connection with all the peers in a specified peer group, run the **reset bgp multicast group** *group-name* command in the user view.
- To reset all external connections, run the **reset bgp multicast external** command in the user view.
- To reset all internal connections, run the **reset bgp multicast internal** command in the user view.

----End

9.19.3 Clearing BGP Statistics

Context

NOTICE

BGP statistics cannot be restored after being cleared. Exercise caution when you reset BGP statistics.

Procedure

- To clear route flapping statistics, 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.
- To clear route flapping statistics on a specified peer, run the **reset bgp** *ipv4-address* **flap-info** command in the user view.
- To clear route dampening statistics and release suppressed routes, run the **reset bgp dampening** [*ipv4-address* [*mask* | *mask-length*]] command in the user view.
- To clear MBGP route dampening statistics, run the **reset bgp multicast dampening** [*ip-address* [*mask* | *mask-length*]] command in the user view.
- To clear MBGP route flapping statistics, run the **reset bgp multicast flap-info** [*ip-address* [*mask* | *mask-length*] | **as-path-filter** { *as-path-list-number* | *as-path-list-name* } | **regrexp** *regrexp*] command in the user view.

----End

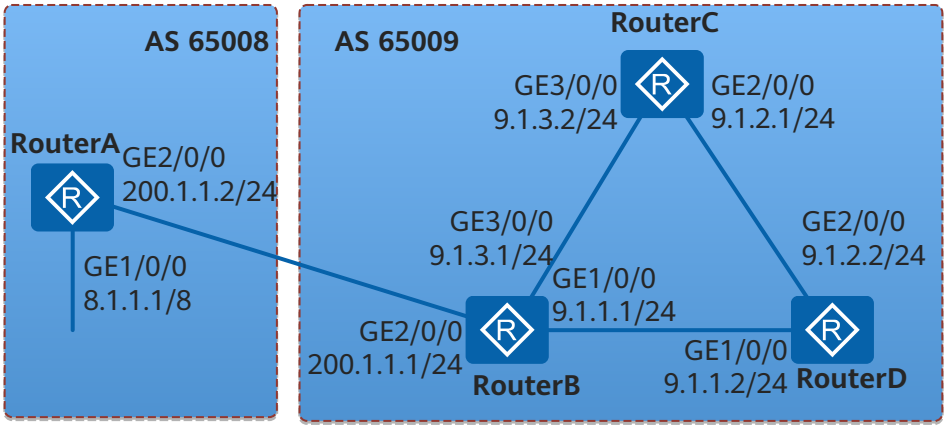
9.20 Configuration Examples for BGP

9.20.1 Example for Configuring Basic BGP Functions

Networking Requirements

As shown in **Figure 9-26**, BGP runs between Routers; an EBGP connection is established between Router A and Router B; IBGP full-mesh connections are established between Router B, Router C, and Router D.

Figure 9-26 Networking diagram of configuring basic BGP 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 connection between Router A and Router B.

Procedure

Step 1 Configure an IP address for each interface.

Configure Router A.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 8.1.1.1 8
[RouterA-GigabitEthernet1/0/0] quit
```

The configurations of Router B, Router C, and Router D are similar to the configuration of Router A, and are not mentioned here.

Step 2 Configure IBGP connections.

Configure Router B.

```
[RouterB] bgp 65009
[RouterB-bgp] router-id 2.2.2.2
[RouterB-bgp] peer 9.1.1.2 as-number 65009
[RouterB-bgp] peer 9.1.3.2 as-number 65009
```

Configure Router C.


```
[RouterC] bgp 65009
[RouterC-bgp] router-id 3.3.3.3
[RouterC-bgp] peer 9.1.3.1 as-number 65009
[RouterC-bgp] peer 9.1.2.2 as-number 65009
[RouterC-bgp] quit
```

Configure Router D.

```
[RouterD] bgp 65009
[RouterD-bgp] router-id 4.4.4.4
[RouterD-bgp] peer 9.1.1.1 as-number 65009
[RouterD-bgp] peer 9.1.2.1 as-number 65009
[RouterD-bgp] quit
```

Step 3 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
```

Configure Router B.

```
[RouterB-bgp] peer 200.1.1.2 as-number 65008
```

View the status of BGP peers.

```
[RouterB-bgp] 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
```

The preceding command output shows that BGP connections have been established between Router B and other Routers.

Step 4 Configure Router A to advertise route 8.0.0.0/8.

Configure Router A to advertise a route.

```
[RouterA-bgp] ipv4-family unicast
[RouterA-bgp-af-ipv4] network 8.0.0.0 255.0.0.0
[RouterA-bgp-af-ipv4] quit
```

View the routing table of Router A.

```
[RouterA-bgp] 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
```

View the routing table of Router B.

```
[RouterB-bgp] 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
```

View 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

The preceding command output shows that Router C has learned the route to destination 8.0.0.0 in AS 65008. The route, however, is invalid because the next hop 200.1.1.2 of this route is unreachable.

Step 5 Configure BGP to import direct routes.

Configure Router B.

```
[RouterB-bgp] ipv4-family unicast
[RouterB-bgp-af-ipv4] import-route direct
```

View the BGP routing table of Router A.

```
[RouterA-bgp] 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: 4
     Network          NextHop        MED        LocPrf    PrefVal Path/Ogn

*>   8.0.0.0          0.0.0.0          0                    0      i
*>   9.1.1.0/24       200.1.1.1        0                    0      65009?
*>   9.1.3.0/24       200.1.1.1        0                    0      65009?
     200.1.1.0        200.1.1.1        0                    0      65009?
```

View 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: 4
     Network          NextHop        MED        LocPrf    PrefVal Path/Ogn
```

```
*>i  8.0.0.0          200.1.1.2       0          100        0      65008i
*>i  9.1.1.0/24        9.1.3.1         0          100        0      ?
  i  9.1.3.0/24        9.1.3.1         0          100        0      ?
*>i  200.1.1.0         9.1.3.1         0          100        0      ?
```

The preceding command output shows that the route to destination 8.0.0.0 becomes valid because the next-hop address of this route is the address of Router A.

Run the **ping** command on Router C.

```
[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
 ip address 8.1.1.1 255.0.0.0
#
interface GigabitEthernet2/0/0
 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
  peer 200.1.1.1 enable
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 200.1.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 9.1.3.1 255.255.255.0
#
bgp 65009
 router-id 2.2.2.2
 peer 9.1.1.2 as-number 65009
 peer 9.1.3.2 as-number 65009
 peer 200.1.1.2 as-number 65008
 #
```



```
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 9.1.1.2 enable
  peer 9.1.3.2 enable
  peer 200.1.1.2 enable
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface GigabitEthernet2/0/0
 ip address 9.1.2.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 9.1.3.2 255.255.255.0
#
bgp 65009
 router-id 3.3.3.3
 peer 9.1.2.2 as-number 65009
 peer 9.1.3.1 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
  peer 9.1.2.2 enable
  peer 9.1.3.1 enable
#
return
```

- Configuration file of Router D

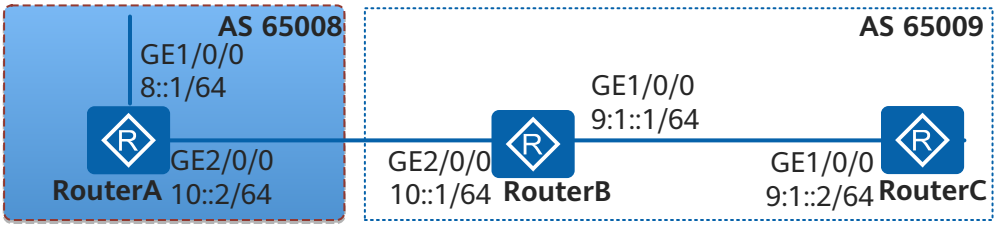
```
#
 sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 9.1.2.2 255.255.255.0
#
bgp 65009
 router-id 4.4.4.4
 peer 9.1.1.1 as-number 65009
 peer 9.1.2.1 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
  peer 9.1.1.1 enable
  peer 9.1.2.1 enable
#
return
```

9.20.2 Example for Configuring Basic BGP4+ Functions

Networking Requirements

As shown in **Figure 9-27**, there are two ASs: 65008 and 65009. Router A belongs to AS 65008; Router B, and Router C belong to AS65009. Routing Protocol is required to exchange the routing information between the two ASs.

Figure 9-27 Figure 1 Networking diagram of configuring basic BGP4+ functions



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the IBGP connection between Router B and Router C.
2. Configure the EBGP connection between Router A and Router B.

Procedure

Step 1 Assign an IPv6 address for each interface.

Configure IPv6 addresses for interfaces on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] ipv6
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipv6 enable
[RouterA-GigabitEthernet1/0/0] ipv6 address 8::1/64
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet1/0/0] ipv6 enable
[RouterA-GigabitEthernet1/0/0] ipv6 address 10::2/64
```

The configurations of RouterB and RouterC are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Configure the IBGP.

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] ipv6-family unicast
[RouterB-bgp-af-ipv6] peer 9:1::2 enable
[RouterB-bgp-af-ipv6] network 9:1:: 64
```

Configure Router C.

```
[RouterC] ipv6
[RouterC] bgp 65009
[RouterC-bgp] router-id 3.3.3.3
[RouterC-bgp] peer 9:1::1 as-number 65009
[RouterC-bgp] ipv6-family unicast
[RouterC-bgp-af-ipv6] peer 9:1::1 enable
[RouterC-bgp-af-ipv6] network 9:1:: 64
```

Step 3 Configure the EBGP.

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
```

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
```

Check the connection status of BGP4+ peers.

```
[RouterB] display bgp ipv6 peer

 BGP local router ID : 2.2.2.2
 Local AS number : 65009
 Total number of peers : 2                 Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv

  9:1::2          4       65009       10       14     0 00:07:10 Established       1
  10::2           4       65008        6        6     0 00:02:17 Established       2
```

The routing table shows that Router B has set up BGP4+ connections with 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: 4
 *>  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  : 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 routing table shows that Router A has learned the route from AS 65009. AS 65008 and AS 65009 can exchange their routing information.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 8::1/64
#
interface GigabitEthernet2/0/0
 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
  undo synchronization
  network 8:: 64
  network 10:: 64
  peer 10::1 enable
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 9:1::1/64
#
interface GigabitEthernet2/0/0
 ipv6 enable
 ipv6 address 10::1/64
#
bgp 65009
 router-id 2.2.2.2
 peer 9:1::2 as-number 65009
 peer 10::2 as-number 65008
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 9:1:: 64
  network 10:: 64
  peer 9:1::2 enable
  peer 10::2 enable
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
 ipv6
#
interface GigabitEthernet1/0/0
```



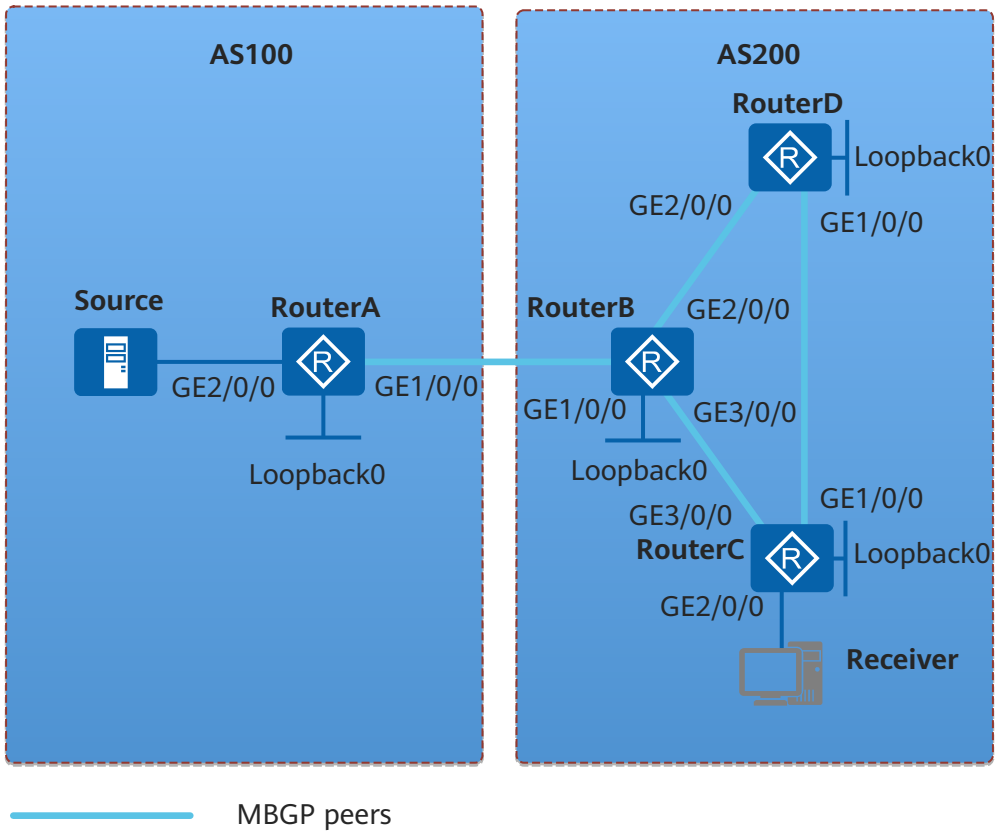
```
 ipv6 enable
 ipv6 address 9:1::2/64
#
bgp 65009
 router-id 3.3.3.3
 peer 9:1::1 as-number 65009
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 9:1:: 64
  peer 9:1::1 enable
#
return
```

9.20.3 Example for Configuring Basic MBGP Functions

Networking Requirements

As shown in **Figure 9-28**, the receiver receives VoD information in multicast mode. The receiver and the source reside in different ASs. Multicast routing information needs to be transmitted between ASs.

Figure 9-28 Networking diagram of configuring MBGP



Device	Interface	IP Address	Device	Interface	IP Address
RouterA	GE1/0/0	10.1.1.1/24	RouterC	GE1/0/0	10.4.1.1/24

Device	Interface	IP Address	Device	Interface	IP Address
	GE2/0/0	10.10.10.1/24		GE2/0/0	10.168.1.1/24
	Loopback0	1.1.1.1/32		GE3/0/0	10.2.1.1/24
RouterB	GE1/0/0	10.1.1.2/24		Loopback0	3.3.3.3/32
	GE2/0/0	10.3.1.2/24	RouterD	GE1/0/0	10.4.1.2/24
	GE3/0/0	10.2.1.2/24		GE2/0/0	10.3.1.1/24
	Loopback0	2.2.2.2/32		Loopback0	4.4.4.4/32

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure MBGP peers for inter-AS multicast transmission.
2. Configure the routes advertised by MBGP.
3. Enable the multicast function on each router.
4. Configure basic PIM-SM functions on each router in ASs and enable IGMP on receiver-side interfaces.
5. Configure a BSR boundary on the interfaces that connect to two ASs.
6. Configure MSDP peers to transmit inter-domain multicast source information.

Procedure

Step 1 Assign IP addresses to the interfaces on each router and configure OSPF in ASs.

Configure IP addresses and masks for the interfaces on each router according to **Figure 9-28** and configure OSPF on the routers in ASs. Ensure that Router B, Router C, Router D can communicate with the receiver at the network layer, learn routes to the loopback interfaces of each other, and dynamically update routes using a unicast routing protocol. Configure OSPF process 1. The configuration procedure is not mentioned here.

Step 2 Configure BGP, enable the MBGP protocol, and configure MBGP peers.

Configure BGP and the MBGP peer on Router A.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] bgp 100
[RouterA-bgp] peer 10.1.1.2 as-number 200
[RouterA-bgp] ipv4-family multicast
[RouterA-bgp-af-multicast] peer 10.1.1.2 enable
[RouterA-bgp-af-multicast] quit
[RouterA-bgp] quit
```

Configure BGP and the MBGP peer on Router B.

```
[RouterB] bgp 200
[RouterB-bgp] peer 10.1.1.1 as-number 100
[RouterB-bgp] peer 10.2.1.1 as-number 200
```



```
[RouterB-bgp] peer 10.3.1.1 as-number 200
[RouterB-bgp] ipv4-family multicast
[RouterB-bgp-af-multicast] peer 10.1.1.1 enable
[RouterB-bgp-af-multicast] peer 10.2.1.1 enable
[RouterB-bgp-af-multicast] peer 10.3.1.1 enable
[RouterB-bgp-af-multicast] quit
[RouterB-bgp] quit
```

Configure BGP and the MBGP peer on Router C.

```
[RouterC] bgp 200
[RouterC-bgp] peer 10.2.1.2 as-number 200
[RouterC-bgp] peer 10.4.1.2 as-number 200
[RouterC-bgp] ipv4-family multicast
[RouterC-bgp-af-multicast] peer 10.2.1.2 enable
[RouterC-bgp-af-multicast] peer 10.4.1.2 enable
[RouterC-bgp-af-multicast] quit
[RouterC-bgp] quit
```

Configure BGP and the MBGP peer on Router D.

```
[RouterD] bgp 200
[RouterD-bgp] peer 10.3.1.2 as-number 200
[RouterD-bgp] peer 10.4.1.1 as-number 200
[RouterD-bgp] ipv4-family multicast
[RouterD-bgp-af-multicast] peer 10.3.1.2 enable
[RouterD-bgp-af-multicast] peer 10.4.1.1 enable
[RouterD-bgp-af-multicast] quit
[RouterD-bgp] quit
```

Step 3 Configure the routes to be advertised.

Configure the routes to be advertised on Router A.

```
[RouterA] bgp 100
[RouterA-bgp] import-route direct
[RouterA-bgp] ipv4-family multicast
[RouterA-bgp-af-multicast] import-route direct
[RouterA-bgp-af-multicast] quit
[RouterA-bgp] quit
```

Configure the routes to be advertised on Router B.

```
[RouterB] bgp 200
[RouterB-bgp] import-route direct
[RouterB-bgp] import-route ospf 1
[RouterB-bgp] ipv4-family multicast
[RouterB-bgp-af-multicast] import-route direct
[RouterB-bgp-af-multicast] import-route ospf 1
[RouterB-bgp-af-multicast] quit
[RouterB-bgp] quit
```

Configure the routes to be advertised on Router C. The configuration of Router D is similar to the configuration of Router C, and is not mentioned here.

```
[RouterC] bgp 200
[RouterC-bgp] import-route direct
[RouterC-bgp] ipv4-family multicast
[RouterC-bgp-af-multicast] import-route direct
[RouterC-bgp-af-multicast] import-route ospf 1
[RouterC-bgp-af-multicast] quit
[RouterC-bgp] quit
```

Step 4 Enable the multicast function on each Router and interfaces on the Routers.

Configure Router A.

```
[RouterA] multicast routing-enable
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] pim sm
```



```
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] pim sm
[RouterA-GigabitEthernet2/0/0] quit
```

Configure Router B.

```
[RouterB] multicast routing-enable
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] pim sm
[RouterB-GigabitEthernet1/0/0] quit
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] pim sm
[RouterB-GigabitEthernet2/0/0] quit
[RouterB] interface gigabitethernet 3/0/0
[RouterB-GigabitEthernet3/0/0] pim sm
[RouterB-GigabitEthernet3/0/0] quit
```

Configure Router C.

```
[RouterC] multicast routing-enable
[RouterC] interface gigabitethernet 1/0/0
[RouterC-GigabitEthernet1/0/0] pim sm
[RouterC-GigabitEthernet1/0/0] quit
[RouterC] interface gigabitethernet 2/0/0
[RouterC-GigabitEthernet2/0/0] pim sm
[RouterC-GigabitEthernet2/0/0] igmp enable
[RouterC-GigabitEthernet2/0/0] quit
[RouterC] interface gigabitethernet 3/0/0
[RouterC-GigabitEthernet3/0/0] pim sm
[RouterC-GigabitEthernet3/0/0] quit
```

Configure Router D.

```
[RouterD] multicast routing-enable
[RouterD] interface gigabitethernet 1/0/0
[RouterD-GigabitEthernet1/0/0] pim sm
[RouterD-GigabitEthernet1/0/0] quit
[RouterD] interface gigabitethernet 2/0/0
[RouterD-GigabitEthernet2/0/0] pim sm
[RouterD-GigabitEthernet2/0/0] quit
```

Step 5 Configure the BSR and RP within each AS.

Configure Router A.

```
[RouterA] interface loopback 0
[RouterA-LoopBack0] ip address 1.1.1.1 255.255.255.255
[RouterA-LoopBack0] pim sm
[RouterA-LoopBack0] quit
[RouterA] pim
[RouterA-pim] c-bsr loopback 0
[RouterA-pim] c-rp loopback 0
[RouterA-pim] quit
```

Configure Router B.

```
[RouterB] interface loopback 0
[RouterB-LoopBack0] ip address 2.2.2.2 255.255.255.255
[RouterB-LoopBack0] pim sm
[RouterB-LoopBack0] quit
[RouterB] pim
[RouterB-pim] c-bsr loopback 0
[RouterB-pim] c-rp loopback 0
[RouterB-pim] quit
```

Step 6 Configure a BSR boundary on the interfaces that connect to two ASs.

Configure Router A.


```
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] pim bsr-boundary
[RouterA-GigabitEthernet1/0/0] quit
```

Configure Router B.

```
[RouterB] interface gigabitethernet 1/0/0
[RouterB-GigabitEthernet1/0/0] pim bsr-boundary
[RouterB-GigabitEthernet1/0/0] quit
```

Step 7 Configure MSDP peers.

Configure Router A.

```
[RouterA] msdp
[RouterA-msdp] peer 10.1.1.2 connect-interface gigabitethernet 1/0/0
[RouterA-msdp] quit
```

Configure Router B.

```
[RouterB] msdp
[RouterB-msdp] peer 10.1.1.1 connect-interface gigabitethernet 1/0/0
[RouterB-msdp] quit
```

Step 8 Verify the configuration.

Run the **display bgp multicast peer** command to view the MBGP peer relationship between Routers. For example, information about the MBGP peer relationship on Router A is as follows:

```
[RouterA] display bgp multicast peer
 BGP local router ID : 1.1.1.1
 Local AS number : 100
 Total number of peers : 1                 Peers in established state : 1
  Peer        V   AS   MsgRcvd  MsgSent  OutQ  Up/Down       State     PrefRcv
  10.1.1.2    4  200        82       75     0 00:30:29  Established         17
```

Run the **display msdp brief** command to view information about the MSDP peer relationship between Routers. For example, brief information about the MSDP peer relationship on Router B is as follows:

```
[RouterB] display msdp brief
MSDP Peer Brief Information of VPN-Instance: public net
  Configured      Up        Listen       Connect      Shutdown     Down
  1               1         0            0            0
  Peer's Address     State     Up/Down time    AS     SA Count    Reset Count
  10.1.1.1           Up        00:07:17        100    1           0
```

----End

Configuration Files

- Configuration file of Router A

```
#
sysname RouterA
#
multicast routing-enable
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
 pim bsr-boundary
 pim sm
#
interface GigabitEthernet2/0/0
 ip address 10.10.10.1 255.255.255.0
 pim sm
#
```



```
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 pim sm
#
pim
 c-bsr Loopback0
 c-rp Loopback0
#
bgp 100
 peer 10.1.1.2 as-number 200
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.1.1.2 enable
#
 ipv4-family multicast
  undo synchronization
  import-route direct
  peer 10.1.1.2 enable
#
msdp
 peer 10.1.1.2 connect-interface GigabitEthernet1/0/0
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
multicast routing-enable
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
 pim bsr-boundary
 pim sm
#
interface GigabitEthernet2/0/0
 ip address 10.3.1.2 255.255.255.0
 pim sm
#
interface GigabitEthernet3/0/0
 ip address 10.2.1.2 255.255.255.0
 pim sm
#
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
 pim sm
#
pim
 c-bsr Loopback0
 c-rp Loopback0
#
ospf 1
 area 0.0.0.0
  network 10.2.1.0 0.0.0.255
  network 10.3.1.0 0.0.0.255
  network 2.2.2.2 0.0.0.0
#
bgp 200
 peer 10.1.1.1 as-number 100
 peer 10.2.1.1 as-number 200
 peer 10.3.1.1 as-number 200
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 10.1.1.1 enable
  peer 10.2.1.1 enable
```



```
 peer 10.3.1.1 enable
#
 ipv4-family multicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 10.1.1.1 enable
  peer 10.2.1.1 enable
  peer 10.3.1.1 enable
#
msdp
 peer 10.1.1.1 connect-interface GigabitEthernet1/0/0
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
multicast routing-enable
#
interface GigabitEthernet1/0/0
 ip address 10.4.1.1 255.255.255.0
 pim sm
#
interface GigabitEthernet2/0/0
 ip address 10.168.1.1 255.255.255.0
 pim sm
 igmp enable
#
interface GigabitEthernet3/0/0
 ip address 10.2.1.1 255.255.255.0
 pim sm
#
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
 pim sm
#
ospf 1
 area 0.0.0.0
  network 10.2.1.0 0.0.0.255
  network 10.4.1.0 0.0.0.255
  network 10.168.1.0 0.0.0.255
  network 3.3.3.3 0.0.0.0
#
bgp 200
 peer 10.2.1.2 as-number 200
 peer 10.4.1.2 as-number 200
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.2.1.2 enable
  peer 10.4.1.2 enable
#
 ipv4-family multicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 10.2.1.2 enable
  peer 10.4.1.2 enable
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
multicast routing-enable
#
```



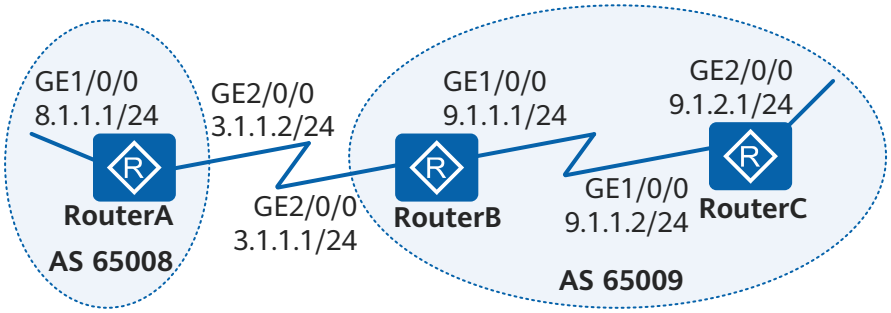
```
interface GigabitEthernet1/0/0
 ip address 10.4.1.2 255.255.255.0
 pim sm
#
interface GigabitEthernet2/0/0
 ip address 10.3.1.1 255.255.255.0
 pim sm
#
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
 pim sm
#
ospf 1
 area 0.0.0.0
  network 10.3.1.0 0.0.0.255
  network 10.4.1.0 0.0.0.255
  network 4.4.4.4 0.0.0.0
#
bgp 200
 peer 10.3.1.2 as-number 200
 peer 10.4.1.1 as-number 200
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.3.1.2 enable
  peer 10.4.1.1 enable
#
 ipv4-family multicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 10.3.1.2 enable
  peer 10.4.1.1 enable
#
return
```

9.20.4 Example for Configuring BGP to Interact with an IGP

Networking Requirements

The network shown in **Figure 9-29** is divided into AS 65008 and AS 65009. In AS 65009, an IGP is used to calculate routes. In this example, OSPF is used as an IGP. The two ASs need to communicate with each other.

Figure 9-29 Networking diagram for configuring BGP to interact with an IGP



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on Routers B and C so that these devices can access each other.
2. Establish an EBGP connection between Routers A and B so that these devices can exchange routing information.
3. Configure BGP and OSPF to import routes from each other on Router B so that the two ASs can communicate with each other.
4. (Optional) Configure BGP route summarization on Router B to simplify the BGP routing table.

Procedure

Step 1 Configure an IP address for each interface.

Configure an IP address to each interface as shown in **Figure 9-29**. For details about the configuration, see the following configuration files.

Step 2 Configuring OSPF

Configure Router B.

```
<Huawei> system-view
[Huawei] sysname RouterB
[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] 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] quit
[RouterC-ospf-1] quit
```

Step 3 Establish 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
```

Configure Router B.

```
[RouterB] bgp 65009
[RouterB-bgp] router-id 2.2.2.2
[RouterB-bgp] peer 3.1.1.2 as-number 65008
```

Step 4 Configure BGP to interact with an IGP

On Router B, configure BGP to import OSPF routes.

```
[RouterB-bgp] ipv4-family unicast
[RouterB-bgp-af-ipv4] import-route ospf 1
[RouterB-bgp-af-ipv4] quit
[RouterB-bgp] quit
```

View 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        0                    0         65009?
 *>   9.1.2.0/24         3.1.1.1        2                    0         65009?
```

On Router B, configure OSPF to import BGP routes.

```
[RouterB] ospf
[RouterB-ospf-1] import-route bgp
[RouterB-ospf-1] quit
```

View the routing table of Router C.

```
[RouterC] 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

        8.1.1.0/24  O_ASE  150  1           D  9.1.1.1         GigabitEthernet1/0/0
        9.1.1.0/24  Direct 0    0           D  9.1.1.2         GigabitEthernet1/0/0
        9.1.1.2/32  Direct 0    0           D  127.0.0.1       GigabitEthernet1/0/0
        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       GigabitEthernet2/0/0
      127.0.0.0/8   Direct 0    0           D  127.0.0.1       InLoopBack0
      127.0.0.1/32  Direct 0    0           D  127.0.0.1       InLoopBack0
```

Step 5 (Optional) Configure automatic route summarization.

BGP is used to transmit routing information on large-scale networks. BGP route summarization can be configured to simplify routing tables of devices on these networks.

Configure Router B.

```
[RouterB] bgp 65009
[RouterB-bgp] ipv4-family unicast
[RouterB-bgp-af-ipv4] summary automatic
```

View 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: 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?
```

Run the **ping -a 8.1.1.1 9.1.2.1** command on Router A.

```
[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 Router A
#
interface GigabitEthernet1/0/0
 ip address 8.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 Router B
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 Router C
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 9.1.2.1 255.255.255.0
#
```



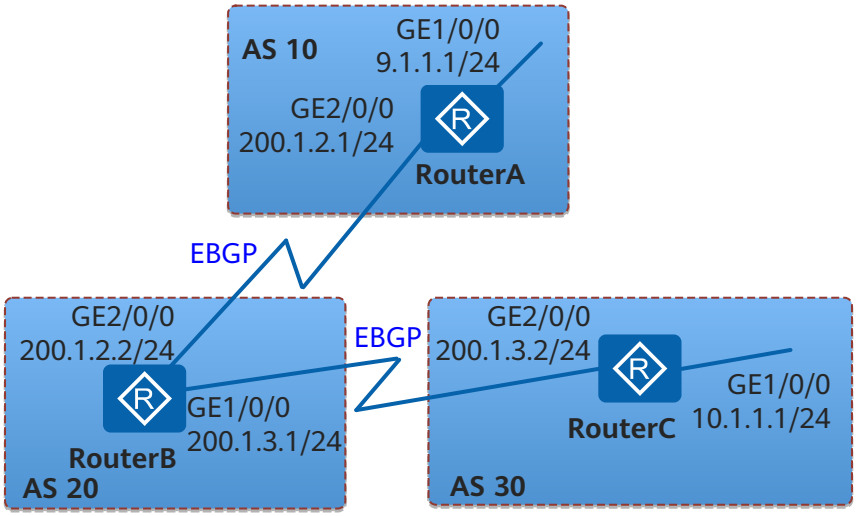
```
ospf 1
 area 0.0.0.0
  network 9.1.1.0 0.0.0.255
  network 9.1.2.0 0.0.0.255
#
return
```

9.20.5 Example for Configuring AS_Path Filters

Networking Requirements

On the network shown in **Figure 9-30**, Router B establishes EBGP connections with Routers A and C. The user wants to disable the devices in AS 10 from communicating with devices in AS 30.

Figure 9-30 Networking diagram for configuring AS_Path filters



Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between Routers A and B and between Routers B and C and configure these devices to import direct routes so that the ASs can communicate with each other through these EBGP connections.
2. Configure AS_Path filters on Router B and use filtering rules to prevent AS 20 from advertising routes of AS 30 to AS 10 or routes of AS 10 to AS 30.

Procedure

Step 1 Configure an IP address for each interface.

Configure IP addresses for all interfaces of Router A.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 9.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
```



```
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 200.1.2.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

The configurations of RouterB and RouterC are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Establish 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] import-route direct
```

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] 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] import-route direct
[RouterC-bgp] quit
```

View routes advertised by Router B. Routes advertised by Router B to Router C are used as an example. You can see that Router B advertises the direct route imported by AS 10.

```
<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: 5
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   9.1.1.0/24         200.1.3.1                      0      20 10?
 *>   10.1.1.0/24        200.1.3.1                      0      20 30?
 *>   200.1.2.0          200.1.3.1      0               0      20?
 *>   200.1.2.1/32       200.1.3.1      0               0      20?
 *>   200.1.3.0/24       200.1.3.1      0               0      20?
```

View the routing table of Router C. You can see that Router C has learned the direct route from 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: 9
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   9.1.1.0/24         200.1.3.1                      0      20 10?
```



```
*>   10.1.1.0/24       0.0.0.0         0                  0      ?
*>   10.1.1.1/32       0.0.0.0         0                  0      ?
*>   127.0.0.0         0.0.0.0         0                  0      ?
*>   127.0.0.1/32      0.0.0.0         0                  0      ?
*>   200.1.2.0         200.1.3.1       0                  0      20?
*>   200.1.3.0/24      0.0.0.0         0                  0      ?
*                      200.1.3.1       0                  0      20?
*>   200.1.3.2/32      0.0.0.0         0                  0      ?
```

Step 3 Configure AS_Path filters on Router B and apply the AS_Path filters to routes to be advertised by Router B.

Create AS_Path filter 1 to deny the routes carrying AS number 30. The regular expression "_30_" indicates any AS list that contains AS 30 and "*" matches any character.

```
[RouterB] ip as-path-filter path-filter1 deny _30_
[RouterB] ip as-path-filter path-filter1 permit .*
```

Create AS_Path filter 2 to deny the routes carrying AS 10.

```
[RouterB] ip as-path-filter path-filter2 deny _10_
[RouterB] ip as-path-filter path-filter2 permit .*
```

Apply the AS_Path filters to routes to be advertised by Router B.

```
[RouterB] bgp 20
[RouterB-bgp] peer 200.1.2.1 as-path-filter path-filter1 export
[RouterB-bgp] peer 200.1.3.2 as-path-filter path-filter2 export
[RouterB-bgp] quit
```

Step 4 # View routes advertised by Router B.

View routes advertised by Router B to AS 30. You can see that Router B does not advertise the direct route imported by AS 10.

```
<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: 2
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          200.1.3.1      0                  0      20?
 *>   200.1.3.0/24       200.1.3.1      0                  0      20?
```

The route does not exist in 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

 *>   10.1.1.0/24        0.0.0.0        0                  0      ?
 *>   10.1.1.1/32        0.0.0.0        0                  0      ?
 *>   127.0.0.0          0.0.0.0        0                  0      ?
 *>   127.0.0.1/32       0.0.0.0        0                  0      ?
 *>   200.1.2.0          200.1.3.1      0                  0      20?
```

```
*>   200.1.3.0/24       0.0.0.0         0                  0      ?
*                    200.1.3.1       0                  0      20?
*>   200.1.3.2/32       0.0.0.0         0                  0      ?
```

View routes advertised by Router B to AS 10. You can see that Router B does not advertise the direct route imported by AS 30.

```
<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: 2
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   200.1.2.0          200.1.2.2      0                  0      20?
 *>   200.1.3.0/24       200.1.2.2      0                  0      20?
```

The route does not exist in 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

 *>   9.1.1.0/24         0.0.0.0         0                  0      ?
 *>   9.1.1.1/32         0.0.0.0         0                  0      ?
 *>   127.0.0.0          0.0.0.0         0                  0      ?
 *>   127.0.0.1/32       0.0.0.0         0                  0      ?
 *>   200.1.2.0          0.0.0.0         0                  0      ?
 *                     200.1.2.2       0                  0      20?
 *>   200.1.2.1/32       0.0.0.0         0                  0      ?
 *>   200.1.3.0/24       200.1.2.2       0                  0      20?
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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
  import-route direct
  peer 200.1.2.2 enable
#
return
```


- Configuration file of Router B

```
#
 sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 200.1.3.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 path-filter1 export
  peer 200.1.3.2 enable
  peer 200.1.3.2 as-path-filter path-filter2 export
#
 ip as-path-filter path-filter1 deny _30_
 ip as-path-filter path-filter1 permit .*
 ip as-path-filter path-filter2 deny _10_
 ip as-path-filter path-filter2 permit .*
#
return
```

- Configuration file of Router C

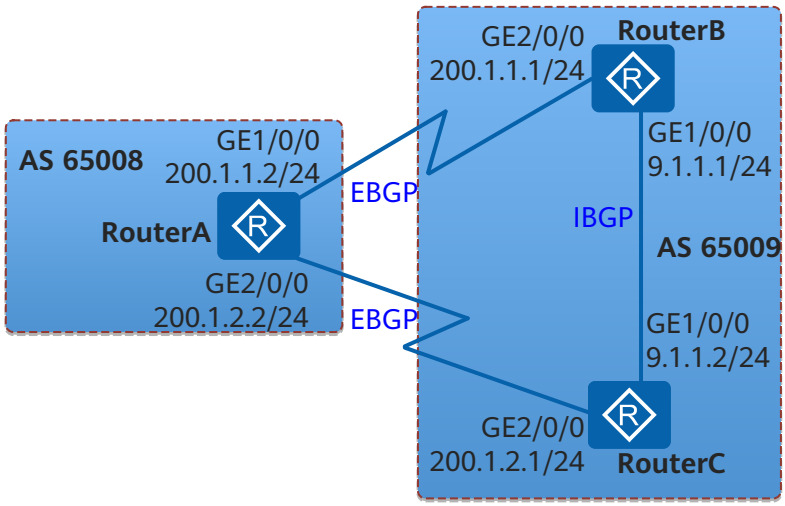
```
#
 sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 200.1.3.2 255.255.255.0
#
bgp 30
 router-id 3.3.3.3
 peer 200.1.3.1 as-number 20
#
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 200.1.3.1 enable
#
return
```

9.20.6 Example for Configuring MED Attributes to Control BGP Route Selection

Networking Requirements

As shown in **Figure 9-31**, BGP is configured on all routers; 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. After a period, traffic from AS 65008 to AS 65009 needs to first pass through RouterC.

Figure 9-31 Networking diagram for configuring MED attributes of routes to control route selection



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. Apply a routing policy to increase the MED value of the route sent by Router B to Router A so that Router A will send traffic to AS 65009 through Router C.

Procedure

Step 1 Configure an IP address for each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 200.1.1.2 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 200.1.2.2 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

The configurations of RouterB and RouterC are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Establish 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] 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] 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] quit
[RouterC-bgp] quit
```

View 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: 00h00m56s
 Direct Out-interface: GigabitEthernet1/0/0
 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: 00h00m06s
 Direct Out-interface: GigabitEthernet2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 AS-path 65009, origin igp, MED 0, pref-val 0, valid, external, pre 255, not preferred for router ID
 Not advertised to any peer yet
```

The preceding command output shows that there are two valid routes to destination 9.1.1.0/24. The route with the next-hop address of 200.1.1.1 is the optimal route because the router ID of Router is smaller.

Step 3 Set MED attributes for routes.

Apply a routing policy to set an MED value for the route advertised by Router B to Router A (the default MED value of a route is 0).

```
[RouterB] route-policy policy10 permit node 10
[RouterB-route-policy] apply cost 100
[RouterB-route-policy] quit
[RouterB] bgp 65009
[RouterB-bgp] peer 200.1.1.2 route-policy policy10 export
```

View 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: 00h07m45s
Direct Out-interface: GigabitEthernet2/0/0
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: 00h00m08s
Direct Out-interface: GigabitEthernet1/0/0
Original nexthop: 200.1.1.1
Qos information : 0x0
AS-path 65009, origin igp, MED 100, pref-val 0, valid, external, pre 255, not preferred for MED
Not advertised to any peer yet
```

The preceding command output shows that the MED value of the route with the next-hop address of 200.1.1.1 (Router B) is 100 and the MED value of the route with the next-hop address of 200.1.2.1 is 0. The route with the smaller MED value is selected.

----End

Configuration Files

- Configuration file of Router A

```
#
sysname Router A
#
interface GigabitEthernet1/0/0
 ip address 200.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 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
  undo synchronization
  peer 200.1.1.1 enable
  peer 200.1.2.1 enable
#
return
```

- Configuration file of Router B

```
#
sysname Router B
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 policy10 export
#
route-policy policy10 permit node 10
 apply cost 100
#
return
```

- Configuration file of Router C

```
#
sysname Router C
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 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
```

9.20.7 Example for Configuring a BGP Route Reflector

Networking Requirements

As shown in **Figure 9-32**, seven Routers need to form an IBGP network. Full-mesh BGP connections have been established between Router B, Router D, and Router E. Users require that the IBGP network be formed without interrupting full-mesh BGP connections between Router B, Router D, and Router E and require simplified device configuration and management.

Figure 9-32 Networking diagram of configuring a BGP RR

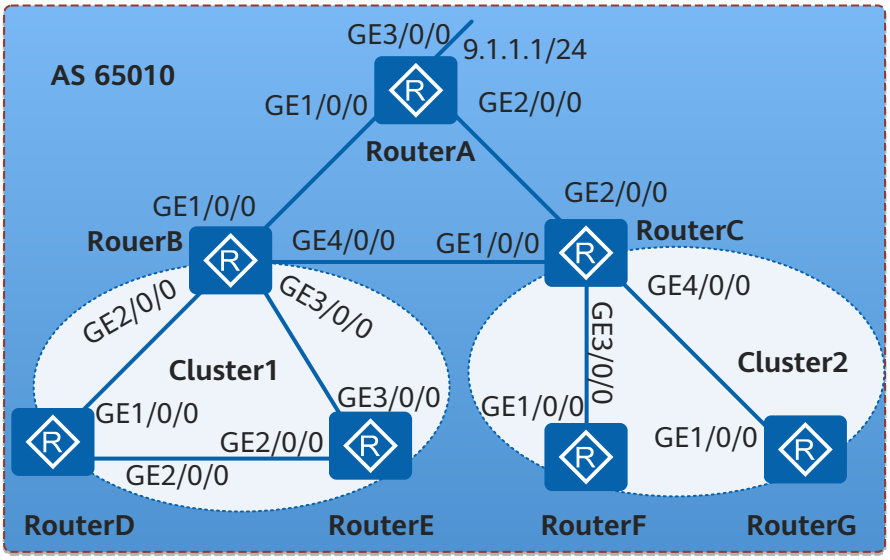


Table 9-8 The BGP RR parameters

Device	Interface	IP address
RouterA	GE 1/0/0	10.1.1.2/24
	GE 2/0/0	10.1.3.2/24
	GE 3/0/0	9.1.1.1/24
RouterB	GE 1/0/0	10.1.1.1/24
	GE 2/0/0	10.1.4.1/24
	GE 3/0/0	10.1.5.1/24
	GE 4/0/0	10.1.2.1/24
RouterC	GE 1/0/0	10.1.2.2/24
	GE 2/0/0	10.1.3.1/24
	GE 3/0/0	10.1.7.1/24
	GE 4/0/0	10.1.8.1/24
RouterD	GE 1/0/0	10.1.4.2/24
	GE 2/0/0	10.1.6.1/24
RouterE	GE 2/0/0	10.1.6.2/24
	GE 3/0/0	10.1.5.2/24
RouterF	GE 1/0/0	10.1.7.2/24
RouterG	GE 1/0/0	10.1.8.2/24

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure RouterB as the route reflector of Cluster1 and RouterD and RouterE as the clients of RouterB. Prohibit communication between the clients to form an IBGP network without interrupting full-mesh BGP connections between RouterB, RouterD, and RouterE.
2. Configure RouterC as the route reflector of Cluster2 and RouterF and RouterG, as the clients of RouterC to simplify device configuration and management.

Procedure

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

Step 2 Configure the IBGP connections between the clients and the RR and between the non-clients and the RR. The configuration details are not mentioned here.

Step 3 Configure the RR.

Configure Router B.

```
<Huawei> system-view
[Huawei] sysname RouterB
[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] 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] ipv4-family unicast
[RouterC-bgp-af-ipv4] peer in_rr reflect-client
[RouterC-bgp-af-ipv4] reflector cluster-id 2
[RouterC-bgp-af-ipv4] quit
```

Display 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
```

You can view that Router D has learned the route advertised by Router A from Router B. For details, see the Originator and Cluster_ID attributes of the route.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.3.2 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 9.1.1.1 255.255.255.0
#
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 GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.4.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 10.1.5.1 255.255.255.0
#
interface GigabitEthernet4/0/0
 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 GigabitEthernet1/0/0
 ip address 10.1.2.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.3.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 10.1.7.1 255.255.255.0
#
interface GigabitEthernet4/0/0
 ip address 10.1.8.1 255.255.255.0
#
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
 #
 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
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 10.1.4.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 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

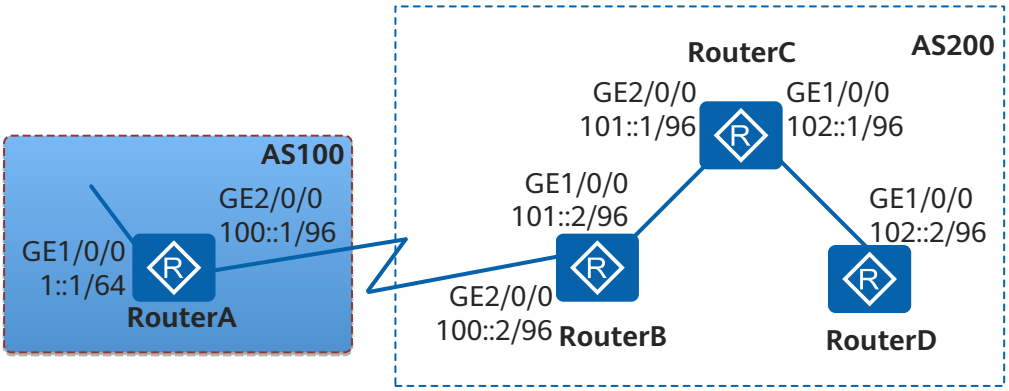
The configuration file of other routers is similar to that of Router D and is omitted here.

9.20.8 Example for Configuring a BGP4+ Route Reflection

Networking Requirements

As shown in **Figure 9-33**, four devices belong to two ASs. You are required to perform simplified configuration to ensure that the two ASs communicate with each other.

Figure 9-33 Networking diagram of configuring BGP4+ route reflection



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP4+ functions to allow BGP neighbors to communicate.
2. Configure RouterC as a route reflector so that no IBGP connection needs to be established between RouterB and RouterD. This simplifies the configuration.

Procedure

Step 1 Configure an IP address for each interface.

Configure IPv6 addresses for interfaces on RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] ipv6
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ipv6 enable
[RouterA-GigabitEthernet1/0/0] ipv6 address 1::1/64
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ipv6 enable
[RouterA-GigabitEthernet2/0/0] ipv6 address 100::1/96
```


The configurations of RouterB, RouterC and RouterD are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Configure basic BGP4+ functions.

Configure RouterA.

```
[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] network 100:: 96
[RouterA-bgp-af-ipv6] quit
[RouterA-bgp] quit
```

Configure RouterB.

```
[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] network 100:: 96
[RouterB-bgp-af-ipv6] network 101:: 96
[RouterB-bgp-af-ipv6] quit
[RouterB-bgp] quit
```

Configure RouterC.

```
[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] network 101:: 96
[RouterC-bgp-af-ipv6] network 102:: 96
```

Configure RouterD.

```
[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] network 102:: 96
[RouterD-bgp-af-ipv6] quit
[RouterD-bgp] quit
```

Step 3 Configure the route reflector.

Configure RouterC as a route reflector, and RouterB and RouterD serve as its clients.

```
[RouterC-bgp-af-ipv6] peer 101::2 reflect-client
[RouterC-bgp-af-ipv6] peer 102::2 reflect-client
```

Check the routing table of RouterB.

```
[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 RouterD.

```
[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 routing table shows that RouterD and RouterB learn the routing information advertised by RouterA from RouterC.

----End

Configuration Files

- Configuration file of RouterA

```
#
 sysname RouterA
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 1::1/64
#
interface GigabitEthernet2/0/0
 ipv6 enable
 ipv6 address 100::1/96
#
bgp 100
 router-id 1.1.1.1
 peer 100::2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 1:: 64
  network 100:: 96
  peer 100::2 enable
#
return
```

- Configuration file of RouterB

```
#
 sysname RouterB
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 101::2/96
#
interface GigabitEthernet2/0/0
 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
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 100:: 96
  network 101:: 96
  peer 100::1 enable
```



```
 peer 101::1 enable
#
return
```

- Configuration file of RouterC

```
#
 sysname RouterC
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 102::1/96
#
interface GigabitEthernet2/0/0
 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
 #
 ipv4-family unicast
  undo synchronization
 #
 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 RouterD

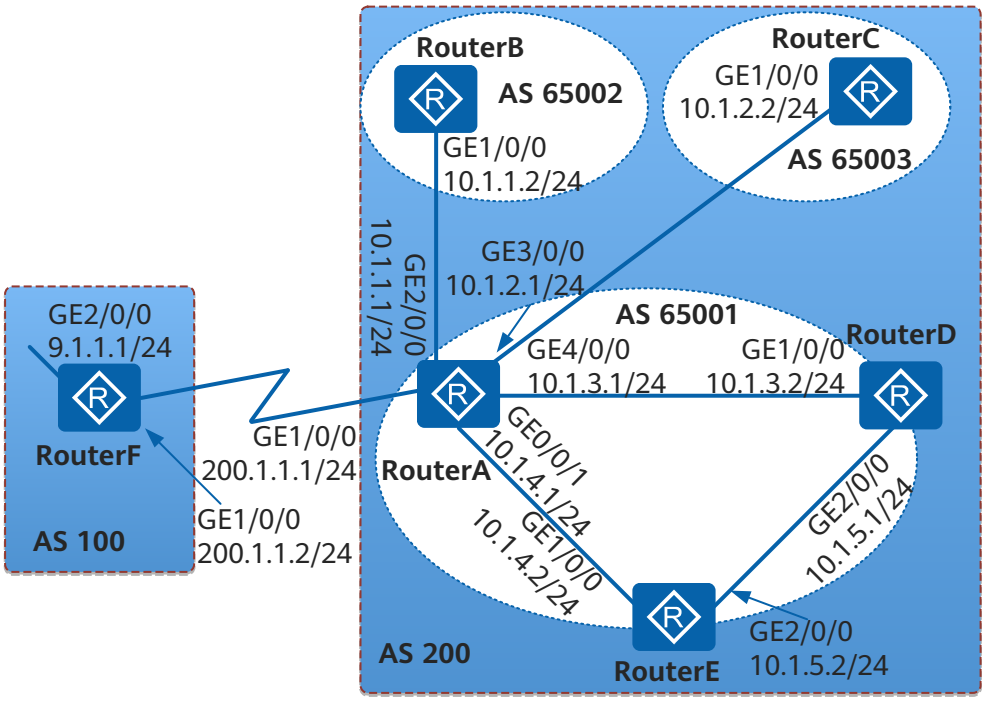
```
#
 sysname RouterD
#
 ipv6
#
interface GigabitEthernet1/0/0
 ipv6 enable
 ipv6 address 102::2/96
#
bgp 200
 router-id 4.4.4.4
 peer 102::1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 102:: 96
  peer 102::1 enable
#
return
```

9.20.9 Example for Configuring a BGP Confederation

Networking Requirements

As shown in **Figure 9-34**, there are multiple BGP routers in AS 200. It is required that the number of IBGP connections be reduced.

Figure 9-34 Networking diagram of configuring the confederation



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a BGP confederation on each router in AS 200 to divide AS 200 into three sub-ASs: AS 65001, AS 65002, and AS 65003. Three routers in AS 65001 establish full-mesh IBGP connections to reduce the number of IBGP connections.

Procedure

Step 1 Configure an IP address to each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 0/0/1
[RouterA-GigabitEthernet0/0/1] ip address 10.1.4.1 255.255.255.0
[RouterA-GigabitEthernet0/0/1] quit
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 200.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
[RouterA] interface gigabitethernet 3/0/0
[RouterA-GigabitEthernet3/0/0] ip address 10.1.2.1 255.255.255.0
[RouterA-GigabitEthernet3/0/0] quit
[RouterA] interface gigabitethernet 4/0/0
[RouterA-GigabitEthernet4/0/0] ip address 10.1.3.1 255.255.255.0
[RouterA-GigabitEthernet4/0/0] quit
```


The configurations of RouterB, RouterC, RouterD, RouterE and RouterF are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Configure the BGP confederation.

Configure Router A.

```
[RouterA] bgp 65001
[RouterA-bgp] router-id 1.1.1.1
[RouterA-bgp] confederation id 200
[RouterA-bgp] confederation peer-as 65002 65003
[RouterA-bgp] peer 10.1.1.2 as-number 65002
[RouterA-bgp] peer 10.1.2.2 as-number 65003
[RouterA-bgp] ipv4-family unicast
[RouterA-bgp-af-ipv4] peer 10.1.1.2 next-hop-local
[RouterA-bgp-af-ipv4] peer 10.1.2.2 next-hop-local
[RouterA-bgp-af-ipv4] quit
```

Configure Router B.

```
[RouterB] bgp 65002
[RouterB-bgp] router-id 2.2.2.2
[RouterB-bgp] confederation id 200
[RouterB-bgp] confederation peer-as 65001
[RouterB-bgp] peer 10.1.1.1 as-number 65001
[RouterB-bgp] quit
```

Configure Router C.

```
[RouterC] bgp 65003
[RouterC-bgp] router-id 3.3.3.3
[RouterC-bgp] confederation id 200
[RouterC-bgp] confederation peer-as 65001
[RouterC-bgp] peer 10.1.2.1 as-number 65001
[RouterC-bgp] quit
```

Step 3 Configure IBGP connections inside AS 65001.

Configure Router A.

```
[RouterA] bgp 65001
[RouterA-bgp] peer 10.1.3.2 as-number 65001
[RouterA-bgp] peer 10.1.4.2 as-number 65001
[RouterA-bgp] ipv4-family unicast
[RouterA-bgp-af-ipv4] peer 10.1.3.2 next-hop-local
[RouterA-bgp-af-ipv4] peer 10.1.4.2 next-hop-local
[RouterA-bgp-af-ipv4] quit
```

Configure Router D.

```
[RouterD] bgp 65001
[RouterD-bgp] router-id 4.4.4.4
[RouterD-bgp] confederation id 200
[RouterD-bgp] peer 10.1.3.1 as-number 65001
[RouterD-bgp] peer 10.1.5.2 as-number 65001
[RouterD-bgp] quit
```

Configure Router E.

```
[RouterE] bgp 65001
[RouterE-bgp] router-id 5.5.5.5
[RouterE-bgp] confederation id 200
[RouterE-bgp] peer 10.1.4.1 as-number 65001
[RouterE-bgp] peer 10.1.5.1 as-number 65001
[RouterE-bgp] quit
```

Step 4 Configure the EBGP connection between AS 100 and AS 200.

Configure Router A.


```
[RouterA] bgp 65001
[RouterA-bgp] peer 200.1.1.2 as-number 100
[RouterA-bgp] quit
```

Configure Router F.

```
[RouterF] bgp 100
[RouterF-bgp] router-id 6.6.6.6
[RouterF-bgp] peer 200.1.1.1 as-number 200
[RouterF-bgp] ipv4-family unicast
[RouterF-bgp-af-ipv4] network 9.1.1.0 255.255.255.0
[RouterF-bgp-af-ipv4] quit
```

Step 5 Verify the configuration.

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
 *>i  9.1.1.0/24         10.1.1.1       0          100       0       (65001) 100i
[RouterB] display bgp routing-table 9.1.1.0
BGP local router ID : 2.2.2.2
 Local AS number : 65002
 Paths:   1 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 10.1.1.1 (1.1.1.1)
 Route Duration: 00h12m29s
 Relay IP Nexthop: 0.0.0.0
 Relay IP Out-Interface: GigabitEthernet1/0/0
 Original nexthop: 10.1.1.1
 Qos information : 0x0
 AS-path (65001) 100, origin igp, MED 0, localpref 100, pref-val 0, valid, external-confed, best, select,
active, pre 255
 Not advertised to any peer yet
```

Check the BGP routing table of Router D.

```
[RouterD] display bgp 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: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn
 *>i  9.1.1.0/24         10.1.3.1       0          100       0       100i
[RouterD] display bgp routing-table 9.1.1.0
BGP local router ID : 4.4.4.4
 Local AS number : 65001
 Paths:   1 available, 1 best, 1 select
 BGP routing table entry information of 9.1.1.0/24:
 From: 10.1.3.1 (1.1.1.1)
 Route Duration: 00h23m57s
 Relay IP Nexthop: 0.0.0.0
 Relay IP Out-Interface: GigabitEthernet1/0/0
 Original nexthop: 10.1.3.1
 Qos information : 0x0
 AS-path 100, origin igp, MED 0, localpref 100, pref-val 0, valid, internal-confed, best, select, active, pre 255
 Not advertised to any peer yet
```

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface GigabitEthernet0/0/1
 ip address 10.1.4.1 255.255.255.0
#
interface GigabitEthernet1/0/0
 ip address 200.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.1.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 ip address 10.1.2.1 255.255.255.0
#
interface GigabitEthernet4/0/0
 ip address 10.1.3.1 255.255.255.0
#
bgp 65001
 router-id 1.1.1.1
 confederation id 200
 confederation peer-as 65002 65003
 peer 200.1.1.2 as-number 100
 peer 10.1.1.2 as-number 65002
 peer 10.1.2.2 as-number 65003
 peer 10.1.3.2 as-number 65001
 peer 10.1.4.2 as-number 65001
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.1.2 enable
  peer 10.1.1.2 enable
  peer 10.1.1.2 next-hop-local
  peer 10.1.2.2 enable
  peer 10.1.2.2 next-hop-local
  peer 10.1.3.2 enable
  peer 10.1.3.2 next-hop-local
  peer 10.1.4.2 enable
  peer 10.1.4.2 next-hop-local
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
#
bgp 65002
 router-id 2.2.2.2
 confederation id 200
 confederation peer-as 65001
 peer 10.1.1.1 as-number 65001
 #
 ipv4-family unicast
  undo synchronization
  peer 10.1.1.1 enable
#
return
```

NOTE

The configuration file of Router C is similar to that of Router B, and is not mentioned here.

- Configuration file of Router D

```
#
 sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 10.1.3.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 10.1.5.1 255.255.255.0
#
bgp 65001
 router-id 4.4.4.4
 confederation id 200
 peer 10.1.3.1 as-number 65001
 peer 10.1.5.2 as-number 65001
 #
 ipv4-family unicast
  undo synchronization
  peer 10.1.3.1 enable
  peer 10.1.5.2 enable
#
return
```

NOTE

The configuration file of Router E is similar to that of Router D, and is not mentioned here.

- Configuration file of Router F

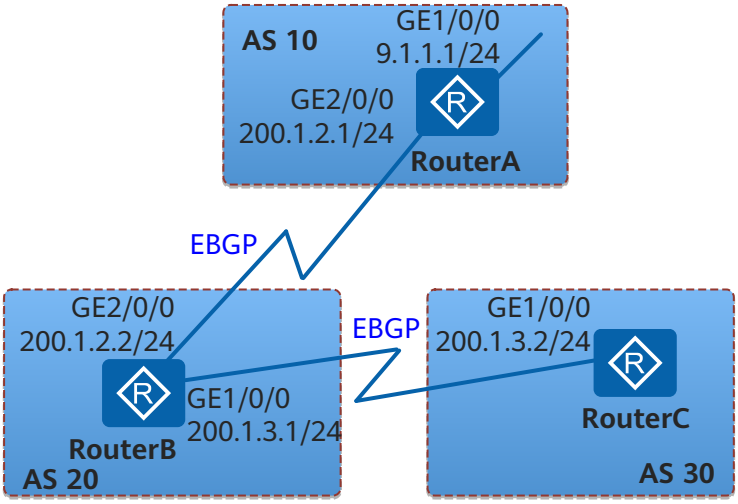
```
#
 sysname RouterF
#
interface GigabitEthernet1/0/0
 ip address 200.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 9.1.1.1 255.255.255.0
#
bgp 100
 router-id 6.6.6.6
 peer 200.1.1.1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  network 9.1.1.0 255.255.255.0
  peer 200.1.1.1 enable
#
return
```

9.20.10 Example for Configuring the BGP Community Attribute

Networking Requirements

As shown in **Figure 9-35**, EBGP connections are established between Router B and Router A, and between Router B and Router C. It is required that AS 20 not advertise the routes advertised by AS 10 to AS 30.

Figure 9-35 Networking diagram of configuring the BGP community



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a route-policy on RouterA to advertise the No_Export attribute so that AS 20 does not advertise the routes advertised by AS 10 to AS 30.

Procedure

Step 1 Configure an IP address for each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 9.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 200.1.2.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

The configurations of RouterB and RouterC are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Establish 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] quit
[RouterA-bgp] 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] 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] quit
```

On Router B, view detailed information about route 9.1.1.0/24.

```
[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: 00h00m42s
 Direct Out-interface: GigabitEthernet2/0/0
 Original nexthop: 200.1.2.1
 Qos information : 0x0
 AS-path 10, origin igp, MED 0, pref-val 0, valid, external, best, select, active, pre 255
 Advertised to such 2 peers:
   200.1.2.1
   200.1.3.2
```

The preceding command output shows that Router B advertises the received BGP route to Router C in AS 30.

View 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: 1
      Network            NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   9.1.1.0/24         200.1.3.1                   0         20 10i
```

The preceding command output shows that Router C has learned route 9.1.1.0/24 from Router B.

Step 3 Configure a BGP community attribute.

Configure a routing policy on Router A to prevent BGP routes to be advertised by Router A to Router B from being advertised to any other AS.

```
[RouterA] route-policy comm_policy permit node 10
[RouterA-route-policy] apply community no-export
[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
```

On Router B, view detailed information about route 9.1.1.0/24.

```
[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: 00h00m09s
Direct Out-interface: GigabitEthernet2/0/0
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, active, pre 255
Not advertised to any peer yet
```

The preceding command output shows that route 9.1.1.0/24 carries the configured community attribute and Router B does not advertise this route to any other AS.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
interface GigabitEthernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 GigabitEthernet1/0/0
 ip address 200.1.3.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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
  peer 200.1.2.1 enable
  peer 200.1.3.2 enable
#
return
```


- Configuration file of Router C

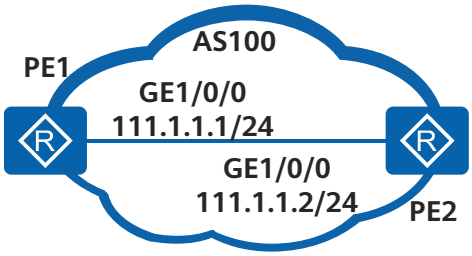
```
#
 sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 200.1.3.2 255.255.255.0
#
bgp 30
 router-id 3.3.3.3
 peer 200.1.3.1 as-number 20
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.3.1 enable
#
return
```

9.20.11 Example for Configuring Prefix-based BGP ORF

Networking Requirements

As shown in **Figure 9-36**, PE1 and PE2 belong to AS 100. PE2 needs to advertise only the routes that match the import policy of PE1 without having to maintain export policies.

Figure 9-36 Networking diagram of configuring prefix-based BGP ORF



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure prefix-based BGP ORF so that PE2 can advertise only the routes that match the import policy of PE1 without having to maintain export policies.

Procedure

Step 1 Establish an IPv4 unicast peer relationship between PE1 and PE2.

Configure PE1.

```
<Huawei> system-view
[Huawei] sysname PE1
[PE1] interface gigabitethernet 1/0/0
[PE1-GigabitEthernet1/0/0] ip address 111.1.1.1 255.255.255.0
[PE1-GigabitEthernet1/0/0] quit
[PE1] bgp 100
[PE1-bgp] peer 111.1.1.2 as-number 100
[PE1-bgp] quit
```


Configure PE2.

```
<Huawei> system-view
[Huawei] sysname PE2
[PE2] interface gigabitethernet 1/0/0
[PE2-GigabitEthernet1/0/0] ip address 111.1.1.2 255.255.255.0
[PE2-GigabitEthernet1/0/0] quit
[PE2] bgp 100
[PE2-bgp] peer 111.1.1.1 as-number 100
[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] 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 static
[PE1-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      111.1.1.2     0          100       0       ?
 *>   4.4.4.4/32      111.1.1.2     0          100       0       ?
 *>   5.5.5.5/32      111.1.1.2     0          100       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] 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] 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
  Received mac routes: 0
  Advertised total routes: 0
  Port:  Local - 54845    Remote - 179
  Configured: Connect-retry Time: 32 sec
  Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Received  : Active Hold Time: 180 sec
  Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Peer optional capabilities:
  Peer supports bgp multi-protocol extension
  Peer supports bgp route refresh capability
  Peer supports bgp 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: 2011/09/25 18:48:15
Last keepalive sent    : 2011/09/25 18:48:19
Last update    received: 2011/09/25 16:11:28
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         111.1.1.2      0          100       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 GigabitEthernet1/0/0
 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
#
 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 GigabitEthernet1/0/0
 ip address 111.1.1.2 255.255.255.0
#
bgp 100
```



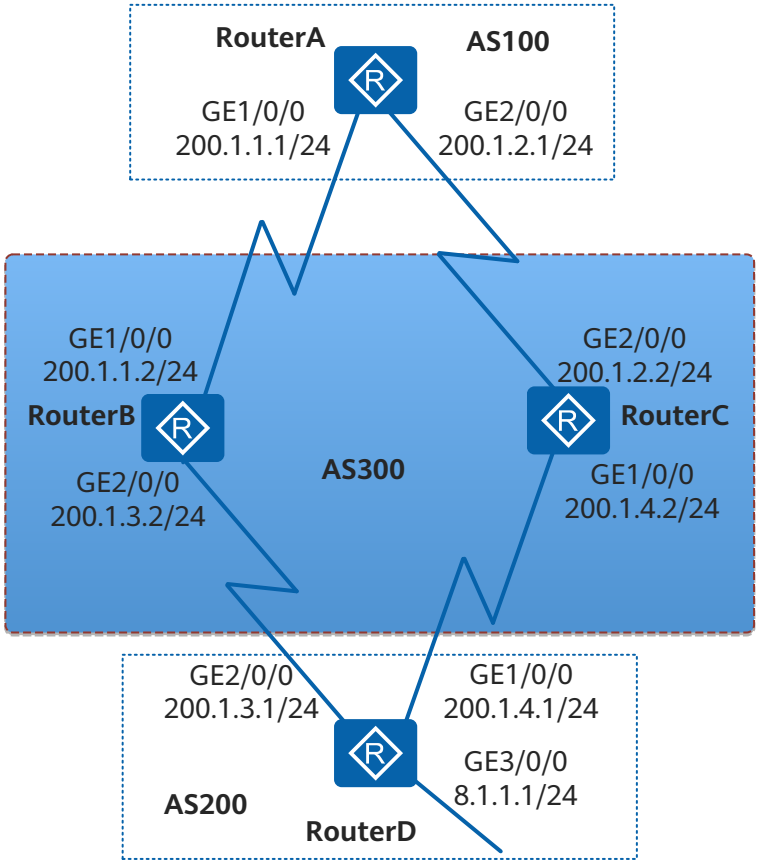
```
 peer 111.1.1.1 as-number 100
#
 ipv4-family unicast
  undo synchronization
  import-route static
  peer 111.1.1.1 enable
  peer 111.1.1.1 capability-advertise orf ip-prefix both
#
 ip route-static 3.3.3.3 255.255.255.255 NULL0
 ip route-static 4.4.4.4 255.255.255.255 NULL0
 ip route-static 5.5.5.5 255.255.255.255 NULL0
#
return
```

9.20.12 Example for Configuring BGP Load Balancing

Networking Requirements

On the network shown in **Figure 9-37**, BGP is configured on all routers. RouterA is in AS 100. RouterB and RouterC are in AS 300. RouterD is in AS 200. Network congestion from RouterA to destination address 8.1.1.0/24 needs to be relieved and network resources need to be fully utilized.

Figure 9-37 Networking diagram of configuring BGP load balancing



Configuration Roadmap

The configuration roadmap is as follows:

1. Establish EBGP connections between RouterA and RouterB and between RouterA and RouterC, between RouterD and RouterB and between RouterD and RouterC to enable ASs to communicate with each other using BGP.
2. Configuring load balancing on RouterA so that RouterA can send traffic to RouterD through either RouterB or RouterC.

Procedure

Step 1 Configure an IP address for each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 200.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 200.1.2.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

The configurations of RouterB, RouterC and RouterD are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Establish BGP connections.

Configure RouterA.

```
[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] quit
```

Configure RouterB.

```
[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] quit
```

Configure RouterC.

```
[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] quit
```

Configure RouterD.

```
[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] quit
[RouterD-bgp] quit
```

View the routing table of RouterA.

```
[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: 00h00m50s
Direct Out-interface: GigabitEthernet1/0/0
Original nexthop: 200.1.1.2
Qos information : 0x0
AS-path 300 200, origin igp, pref-val 0, valid, external, best, select, active, 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: 00h00m51s
Direct Out-interface: GigabitEthernet2/0/0
Original nexthop: 200.1.2.2
Qos information : 0x0
AS-path 300 200, origin igp, pref-val 0, valid, external, pre 255, not preferred for router ID
Not advertised to any peer yet
```

The preceding command output shows that there are two valid routes from RouterA to destination 8.1.1.0/24. The route with the next-hop address of 200.1.1.2 is the optimal route because the router ID of RouterB is smaller.

Step 3 Configure BGP load balancing.

Configure load balancing on RouterA.

```
[RouterA] bgp 100
[RouterA-bgp] ipv4-family unicast
[RouterA-bgp-af-ipv4] maximum load-balancing 2
[RouterA-bgp-af-ipv4] quit
[RouterA-bgp] quit
```

Step 4 Verify the configuration.

View the routing table of RouterA.

```
[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: 00h03m55s
Direct Out-interface: GigabitEthernet1/0/0
Original nexthop: 200.1.1.2
Qos information : 0x0
AS-path 300 200, origin igp, pref-val 0, valid, external, best, select,  active, 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: 00h03m56s
Direct Out-interface: GigabitEthernet2/0/0
Original nexthop: 200.1.2.2
Qos information : 0x0
AS-path 300 200, origin igp, pref-val 0, valid, external, select,  active, pre 255, not preferred for router ID
Not advertised to any peer yet
```

The preceding command output shows that BGP route 8.1.1.0/24 has two next hops: 200.1.1.2 and 200.1.2.2. Both of them are optimal routes.

----End

Configuration Files

- Configuration file of RouterA

```
#
sysname RouterA
#
interface GigabitEthernet1/0/0
 ip address 200.1.1.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 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
  undo synchronization
  maximum load-balancing 2
  peer 200.1.1.2 enable
  peer 200.1.2.2 enable
#
return
```

- Configuration file of RouterB

```
#
sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 200.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 RouterC

```
#
sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 200.1.4.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 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 RouterD

```
#
sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 200.1.4.1 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 200.1.3.1 255.255.255.0
#
interface GigabitEthernet3/0/0
 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
```

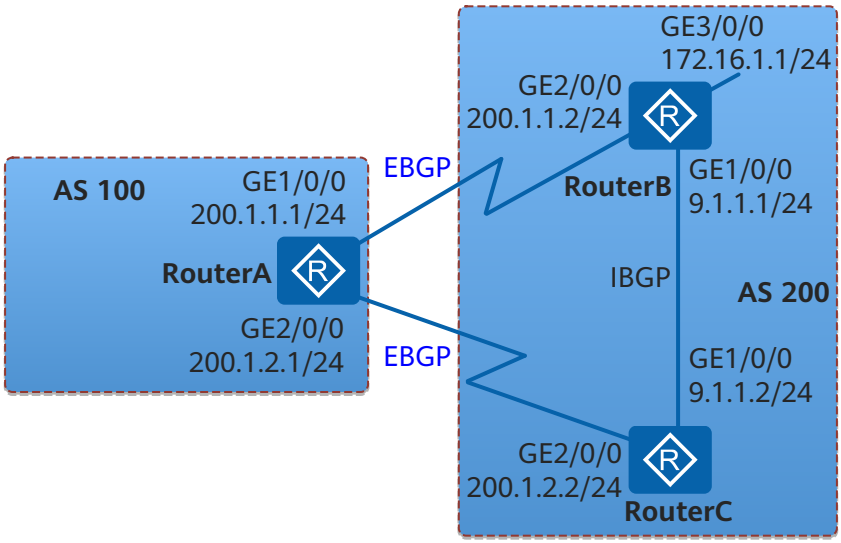
9.20.13 Example for Associating BGP with BFD

Networking Requirements

As shown in **Figure 9-38**, RouterA belongs to AS 100, RouterB and RouterC belong to AS 200. EBGP connections are established between RouterA and RouterB, and between RouterA and RouterC.

Service traffic is transmitted along the primary link RouterA->RouterB. The link RouterA->RouterC->RouterB functions as the backup link. Fast fault detection is required to allow traffic to be fast switched from the primary link to the backup link.

Figure 9-38 Networking diagram of configuring BFD for BGP



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 RouterA and RouterB.

NOTE

If two routers establish an EBGP peer relationship over a direct link, BFD for BGP does not need to be configured. This is because the **ebgp-interface-sensitive** command is enabled by default for directly-connected EBGP peers.

Procedure

Step 1 Configure an IP address for each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 200.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
[RouterA] interface gigabitethernet 2/0/0
[RouterA-GigabitEthernet2/0/0] ip address 200.1.2.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
```

The configurations of RouterB and RouterC are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Configure basic BGP functions. Establish EBGP peer relationships between RouterA and RouterB, and between RouterA and RouterC and an IBGP peer relationship between RouterB and RouterC.

Configure RouterA.


```
[RouterA] bgp 100
[RouterA-bgp] router-id 1.1.1.1
[RouterA-bgp] peer 200.1.1.2 as-number 200
[RouterA-bgp] peer 200.1.1.2 ebgp-max-hop
[RouterA-bgp] peer 200.1.2.2 as-number 200
[RouterA-bgp] peer 200.1.2.2 ebgp-max-hop
[RouterA-bgp] quit
```

Configure RouterB.

```
[RouterB] bgp 200
[RouterB-bgp] router-id 2.2.2.2
[RouterB-bgp] peer 200.1.1.1 as-number 100
[RouterB-bgp] peer 200.1.1.1 ebgp-max-hop
[RouterB-bgp] peer 9.1.1.2 as-number 200
[RouterB-bgp] network 172.16.1.0 255.255.255.0
[RouterB-bgp] quit
```

Configure RouterC.

```
[RouterC] bgp 200
[RouterC-bgp] router-id 3.3.3.3
[RouterC-bgp] peer 200.1.2.1 as-number 100
[RouterC-bgp] peer 200.1.2.1 ebgp-max-hop
[RouterC-bgp] peer 9.1.1.1 as-number 200
[RouterC-bgp] quit
```

Check the status of BGP peer relationships on RouterA. The command output shows that the BGP peer relationships are in the Established state.

```
<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 route sent from RouterC or RouterB to RouterA by using a routing policy.

Configure RouterB.

```
[RouterB] route-policy 10 permit node 10
[RouterB-route-policy] apply cost 100
[RouterB-route-policy] quit
[RouterB] bgp 200
[RouterB-bgp] peer 200.1.1.1 route-policy 10 export
```

Configure RouterC.

```
[RouterC] route-policy 10 permit node 10
[RouterC-route-policy] apply cost 150
[RouterC-route-policy] quit
[RouterC] bgp 200
[RouterC-bgp] peer 200.1.2.1 route-policy 10 export
```

Check BGP routing information on RouterA.

```
<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: 5
     Network          NextHop        MED        LocPrf    PrefVal Path/Ogn
*>  9.1.1.0/24       200.1.2.2       150                0      200?
*>  172.16.1.0/24    200.1.1.2       100                0      200i
*                    200.1.2.2       150                0      200i
*>   200.1.2.0       200.1.1.2       100                0      200?
                     200.1.2.2       150                0      200?
```

As shown in the BGP routing table, the next-hop address of the route to 172.16.1.0/24 is 200.1.1.2, and service traffic is transmitted on the primary link between RouterA and RouterB.

Step 4 Configure BFD, and set the interval for transmitting BFD packets, the interval for receiving BFD packets, and the local detection multiplier.

Enable BFD on RouterA. Set the minimum intervals for transmitting and receiving BFD 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
```

Enable BFD on RouterB. Set the minimum intervals for transmitting and receiving BFD 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
```

Display all BFD sessions on RouterA.

```
<RouterA> display bgp bfd session all
 Local_Address      Peer_Address      LD/RD      Interface
 200.1.1.1          200.1.2.2         8201/8201  GigibitEthernet1/0/0
 Tx-interval(ms)    Rx-interval(ms)   Multiplier  Session-State
 100                100               4          Up
 Wtr-interval(m)
 0
```

Step 5 Verify the configuration.

Run the **shutdown** command on GE 2/0/0 of RouterB to simulate a fault on the primary link.

```
[RouterB] interface gigabitethernet 2/0/0
[RouterB-Gigabitethernet2/0/0] shutdown
```

Check the BGP routing table on RouterA.

```
<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

*>  9.1.1.0/24       200.1.2.2      150                0      200?
```



```
*> 172.16.1.0/24      200.1.2.2       150             0      200i
   200.1.2.0          200.1.2.2       150             0      200?
```

As shown in the BGP routing table, the backup link of RouterA -> RouterC -> RouterB takes effect after the primary link fails, and the next-hop address of the route to 172.16.1.0/24 is 200.1.2.2.

----End

Configuration Files

- Configuration file of RouterA

```
#
 sysname RouterA
#
  bfd
#
interface Gigabitethernet1/0/0
 ip address 200.1.1.1 255.255.255.0
#
interface Gigabitethernet2/0/0
 ip address 200.1.2.1 255.255.255.0
#
bgp 100
 router-id 1.1.1.1
 peer 200.1.1.2 as-number 200
 peer 200.1.1.2 ebgp-max-hop 255
 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
 peer 200.1.2.2 ebgp-max-hop 255
 #
 ipv4-family unicast
  undo synchronization
  peer 200.1.1.2 enable
  peer 200.1.2.2 enable
#
return
```

- Configuration file of RouterB

```
#
 sysname RouterB
#
 bfd
#
interface Gigabitethernet1/0/0
 ip address 9.1.1.1 255.255.255.0
#
interface Gigabitethernet2/0/0
 ip address 200.1.1.2 255.255.255.0
#
interface Gigabitethernet3/0/0
 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 ebgp-max-hop 255
 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 RouterC

```
#
 sysname RouterC
#
 bfd
#
interface Gigabitethernet1/0/0
 ip address 9.1.1.2 255.255.255.0
#
interface Gigabitethernet2/0/0
 ip address 200.1.2.2 255.255.255.0
#
bgp 200
 router-id 3.3.3.3
 peer 9.1.1.1 as-number 200
 peer 200.1.2.1 as-number 100
 peer 200.1.2.1 ebgp-max-hop 255
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  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
```

9.20.14 Example for Configuring QPPB (BGP)

This section provides an example for configuring QPPB.

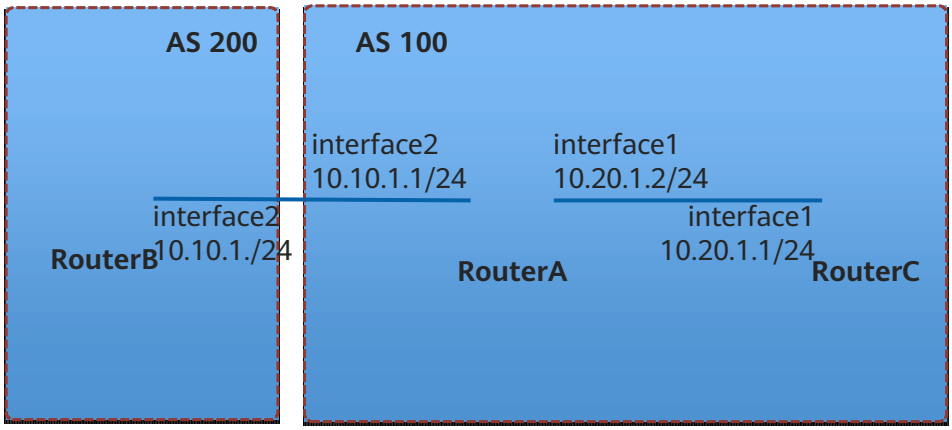
Networking Requirements

In **Figure 9-39**, RouterB advertises BGP routes with the community attribute to RouterA. After RouterA receives these routes, it configures associated QoS local IDs for the routes that match the BGP community list and applies a local QPPB policy to traffic that traverses RouterA.

Traffic from RouterB to RouterC needs to traverse RouterA. RouterB is the BGP route sender, while RouterA is the BGP route receiver.

Source-based QPPB must be deployed on RouterA.

Figure 9-39 Configuring QPPB



Configuration Guidelines

None

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure basic BGP functions.
2. On RouterB, configure a routing policy to set the community attribute for the routes to be sent and advertise routing policies through BGP.
3. Apply the routing policy on RouterA to match route attributes and set associated QoS local IDs for routes.
4. Apply a QPPB policy to the inbound interface of RouterA.

Data Preparation

To complete the configuration, you need the following data:

- IP addresses of interfaces
- Routing policy name, match rules, and route attributes
- QPPB policy name

Procedure

Step 1 Configure basic BGP functions on RouterA and RouterB.

Configure loopback interfaces on RouterA and RouterB.

```
<RouterA> system-view
[RouterA] interface loopback 0
[RouterA-LoopBack0] ip address 1.1.1.1 255.255.255.255
[RouterA-LoopBack0] quit
<RouterB> system-view
[RouterB] interface loopback 0
[RouterB-LoopBack0] ip address 2.2.2.2 255.255.255.255
```


Configure the interfaces that directly connect RouterA and RouterB and the interfaces that directly connect RouterA and RouterC.

```
<RouterA> system-view
[RouterA] interface GigabitEthernet 2/0/0
[RouterA-GigabitEthernet2/0/0] undo shutdown
[RouterA-GigabitEthernet2/0/0] ip address 10.10.1.1 255.255.255.0
[RouterA-GigabitEthernet2/0/0] quit
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] undo shutdown
[RouterA-GigabitEthernet1/0/0] ip address 10.20.1.2 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
<RouterB> system-view
[RouterB] interface GigabitEthernet 1/0/0
[RouterB-GigabitEthernet1/0/0] undo shutdown
[RouterB-GigabitEthernet1/0/0] ip address 10.10.1.2 255.255.255.0
<RouterC> system-view
[RouterC] interface gigabitethernet1/0/0
[RouterC-GigabitEthernet1/0/0] undo shutdown
[RouterC-GigabitEthernet1/0/0] ip address 10.20.1.1 255.255.255.0
[RouterC-GigabitEthernet1/0/0] return
```

Enable OSPF to advertise the routes to the IP addresses of interfaces.

```
<RouterA> system-view
[RouterA] ospf
[RouterA-ospf-1] area 0
[RouterA-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[RouterA-ospf-1-area-0.0.0.0] network 10.10.1.0 0.0.0.255
[RouterA-ospf-1-area-0.0.0.0] network 10.20.1.0 0.0.0.255
[RouterA-ospf-1-area-0.0.0.0] quit
[RouterA-ospf-1] return
<RouterB> system-view
[RouterB] ospf
[RouterB-ospf] area 0
[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] network 10.10.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] return
<RouterC> system-view
[RouterC] ospf
[RouterC-ospf] area 0
[RouterC-ospf-1-area-0.0.0.0] network 10.20.1.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.0] return
```

Configure BGP and establish an EBGP connection between RouterA and RouterB.

```
<RouterA> system-view
[RouterA] bgp 100
[RouterA-bgp] peer 2.2.2.2 as-number 200
[RouterA-bgp] peer 2.2.2.2 connect-interface loopback 0
[RouterA-bgp] import-route direct
[RouterA-bgp] return
<RouterB> system-view
[RouterB] bgp 200
[RouterB-bgp] peer 1.1.1.1 as-number 100
[RouterB-bgp] peer 1.1.1.1 ebgp-max-hop 2
[RouterB-bgp] peer 1.1.1.1 connect-interface loopback 0
[RouterB-bgp] import-route direct
[RouterB-bgp] return
```

Configure BGP and establish an EBGP connection between RouterA and RouterC.

```
<RouterA> system-view
[RouterA] bgp 100
[RouterA-bgp] peer 10.20.1.1 as-number 100
```



```
[RouterA-bgp] import-route direct
[RouterA-bgp] quit
<RouterC> system-view
[RouterC] bgp 100
[RouterC-bgp] peer 10.20.1.2 as-number 100
[RouterC-bgp] import-route direct
[RouterC-bgp] quit
```

After the configurations are complete, RouterA can communicate with RouterB and RouterC.

Step 2 Configure a routing policy on the route sender RouterB and apply the routing policy.

Configure an IP prefix for the route sender.

```
<RouterB> system-view
[RouterB] ip ip-prefix bb permit 66.1.1.1 32
[RouterB] return
```

Configure a routing policy on the route sender.

```
<RouterB> system-view
[RouterB] route-policy aa permit node 10
[RouterB-route-policy] if-match ip-prefix bb
[RouterB-route-policy] apply community 10:10
[RouterB-route-policy] return
```

Advertise the routing policy through BGP on the route sender.

```
<RouterB> system-view
[RouterB] bgp 200
[RouterB-bgp] peer 1.1.1.1 route-policy aa export
[RouterB-bgp] peer 1.1.1.1 advertise-community
[RouterB-bgp] return
```

Step 3 Configure a route receiving policy on the route receiver RouterA and apply the associated traffic behavior to the traffic that matches route attributes.

Configure a QoS policy, namely, the associated traffic behavior.

```
<RouterA> system-view
[RouterA] traffic behavior dd
[RouterA-behavior-dd] remark dscp af11
[RouterA-behavior-dd] return
```

Configure a route receiving policy to enforce the associated traffic behavior on the routes that match the community attribute.

```
<RouterA> system-view
[RouterA] ip community-filter 10 permit 10:10
[RouterA] route-policy aa permit node 10
[RouterA-route-policy] if-match community-filter 10
[RouterA-route-policy] apply qos-local-id 1
[RouterA-route-policy] return
```

Configure a QPPB policy on RouterA.

```
<RouterA> system-view
[RouterA] qppb local-policy ac
[RouterA-localpolicy-ac] qos-local-id 1 behavior dd
[RouterA-localpolicy-ac] return
```

On RouterA, apply the route receiving policy to the routes advertised by RouterB.

```
<RouterA> system-view
[RouterA] bgp 100
```



```
[RouterA-bgp] peer 2.2.2.2 route-policy aa import
[RouterA-bgp] return
```

Step 4 On RouterA, apply the QPPB policy to the inbound interface of traffic.

```
<RouterA> system-view
[RouterA] interface GigabitEthernet 1/0/0
[RouterA-GigabitEthernet1/0/0] qppb-policy ac enable
[RouterA-GigabitEthernet1/0/0] return
```

----End

Configuration Files

- Router A configuration file

```
#
 sysname RouterA
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.20.1.2 255.255.255.0
 qppb-policy ac enable
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 10.10.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 200
 peer 2.2.2.2 connect-interface LoopBack0
 peer 10.20.1.1 as-number 100
#
ipv4-family unicast
  undo synchronization
  import-route direct
  peer 2.2.2.2 enable
  peer 2.2.2.2 route-policy aa import
  peer 10.20.1.1 enable
#
ospf 1
 area 0.0.0.0
  network 1.1.1.1 0.0.0.0
  network 10.10.1.0 0.0.0.255
  network 10.20.1.0 0.0.0.255
#
traffic behavior dd
 remark dscp af11
#
route-policy aa permit node 10
 if-match community-filter 10
 apply qos-local-id 1
#
 ip community-filter 10 permit 10:10
#
qppb local-policy ac
 qos-local-id 1 behavior dd
return
```

- RouterB configuration file

```
#
 sysname RouterB
#
interface GigabitEthernet2/0/0
 undo shutdown
 ip address 10.10.1.2 255.255.255.0
#
```



```
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
interface LoopBack10
 ip address 66.1.1.1 255.255.255.255
#
bgp 200
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 ebgp-max-hop 2
 peer 1.1.1.1 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 1.1.1.1 enable
  peer 1.1.1.1 route-policy aa export
  peer 1.1.1.1 advertise-community
  quit
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 10.10.1.0 0.0.0.255
#
route-policy aa permit node 10
 if-match ip-prefix bb
 apply community 10:10
#
 ip ip-prefix bb index 10 permit 66.1.1.1 32
#
return
```

- RouterC configuration file

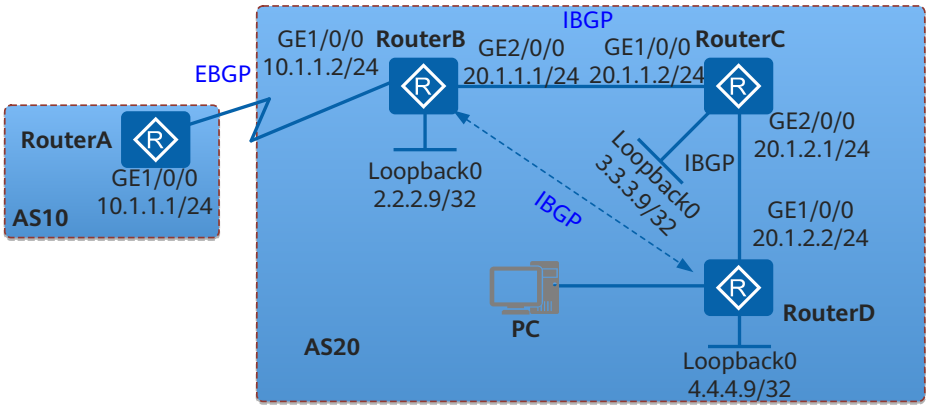
```
#
 sysname RouterC
#
interface GigabitEthernet1/0/0
 undo shutdown
 ip address 10.20.1.1 255.255.255.0
#
bgp 100
 peer 10.20.1.2 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  peer 10.20.1.2 enable
#
ospf 1
 area 0.0.0.0
  network 10.20.1.0 0.0.0.255
#
return
```

9.20.15 Example for Configuring BGP GTSM

Networking Requirements

As shown in **Figure 9-40**, Router A belongs to AS 10, and Router B, Router C, and Router D belong to AS 20. BGP is run in the network and it is required to protect Router B against CPU-utilization attacks.

Figure 9-40 Figure 1 Networking diagram of configuring BGP GTSM



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on Router B, Router C, and Router D to implement interworking in AS 20.
2. Set up an EBGP connection between Router A and Router B, and set up IBGP connections between Router B, Router C, and Router D through loopback interfaces.
3. Configure GTSM on Router A, Router B, Router C, and Router D so that it can protect Router B against CPU-utilization attacks.

Procedure

Step 1 Configure an IP address to each interface.

Configure IP addresses for all interfaces of RouterA.

```
<Huawei> system-view
[Huawei] sysname RouterA
[RouterA] interface gigabitethernet 1/0/0
[RouterA-GigabitEthernet1/0/0] ip address 10.1.1.1 255.255.255.0
[RouterA-GigabitEthernet1/0/0] quit
```

The configurations of RouterB, RouterC and RouterD are similar to the configuration of RouterA, and are not mentioned here.

Step 2 Configure OSPF.

Configure RouterB.

```
[RouterB] ospf
[RouterB-ospf-1] area 0
[RouterB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[RouterB-ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] area 1
[RouterB-ospf-1-area-0.0.0.1] network 2.2.2.9 0.0.0.0
[RouterB-ospf-1-area-0.0.0.1] quit
[RouterB-ospf-1] quit
```

Configure RouterC.

```
[RouterC] ospf
```



```
[RouterC-ospf-1] area 0
[RouterC-ospf-1-area-0.0.0.0] network 20.1.2.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.0] quit
[RouterC-ospf-1] area 1
[RouterC-ospf-1-area-0.0.0.1] network 20.1.1.0 0.0.0.255
[RouterC-ospf-1-area-0.0.0.1] quit
[RouterC-ospf-1] area 2
[RouterC-ospf-1-area-0.0.0.1] network 3.3.3.9 0.0.0.0
[RouterC-ospf-1-area-0.0.0.1] quit
[RouterC-ospf-1] quit
```

Configure RouterD.

```
[RouterD] ospf
[RouterD-ospf-1] area 0
[RouterD-ospf-1-area-0.0.0.0] network 20.1.2.0 0.0.0.255
[RouterD-ospf-1-area-0.0.0.0] quit
[RouterD-ospf-1] area 1
[RouterD-ospf-1-area-0.0.0.1] network 4.4.4.9 0.0.0.0
[RouterD-ospf-1-area-0.0.0.1] quit
[RouterD-ospf-1] quit
```

Step 3 Configure an IBGP connection.

Configure Router B.

```
[RouterB] bgp 20
[RouterB-bgp] router-id 2.2.2.9
[RouterB-bgp] peer 3.3.3.9 as-number 20
[RouterB-bgp] peer 3.3.3.9 connect-interface LoopBack0
[RouterB-bgp] peer 3.3.3.9 next-hop-local
[RouterB-bgp] peer 4.4.4.9 as-number 20
[RouterB-bgp] peer 4.4.4.9 connect-interface LoopBack0
[RouterB-bgp] peer 4.4.4.9 next-hop-local
```

Configure Router C.

```
[RouterC] bgp 20
[RouterC-bgp] router-id 3.3.3.9
[RouterC-bgp] peer 2.2.2.9 as-number 20
[RouterC-bgp] peer 2.2.2.9 connect-interface LoopBack0
[RouterC-bgp] peer 4.4.4.9 as-number 20
[RouterC-bgp] peer 4.4.4.9 connect-interface LoopBack0
```

Configure Router D.

```
[RouterD] bgp 20
[RouterD-bgp] router-id 4.4.4.9
[RouterD-bgp] peer 2.2.2.9 as-number 20
[RouterD-bgp] peer 2.2.2.9 connect-interface LoopBack0
[RouterD-bgp] peer 3.3.3.9 as-number 20
[RouterD-bgp] peer 3.3.3.9 connect-interface LoopBack0
```

Step 4 Configure an EBGP connection.

Configure Router A.

```
[RouterA] bgp 10
[RouterA-bgp] router-id 1.1.1.9
[RouterA-bgp] peer 10.1.1.2 as-number 20
```

Configure Router B.

```
[RouterB-bgp] peer 10.1.1.1 as-number 10
```

Display the connection status of the BGP peers.

```
[RouterB-bgp] display bgp peer
 BGP local router ID : 2.2.2.9
 Local AS number : 20
```



```
Total number of peers : 3                 Peers in established state : 3

 Peer            V    AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv

 3.3.3.9         4    20        8        7     0 00:05:06 Established     0
 4.4.4.9         4    20        8       10     0 00:05:33 Established     0
 10.1.1.1        4    10        7        7     0 00:04:09 Established     0
```

You can view that Router B has set up BGP connections with other routers.

Step 5 Configure GTSM on Router A and Router B. Router A and Router B are directly connected, so the range of the TTL value between the two routers is [255, 255]. The value of **valid-ttl-hops** is 1.

Configure GTSM on Router A.

```
[RouterA-bgp] peer 10.1.1.2 valid-ttl-hops 1
```

Configure GTSM of the EBGP connection on Router B.

```
[RouterB-bgp] peer 10.1.1.1 valid-ttl-hops 1
```

Check the GTSM configuration.

```
[RouterB-bgp] display bgp peer 10.1.1.1 verbose
      BGP Peer is 10.1.1.1,  remote AS 10
      Type: EBGP link
      BGP version 4, Remote router ID 1.1.1.9

      Update-group ID : 2
      BGP current state: Established, Up for 00h49m35s
      BGP current event: RecvKeepalive
      BGP last state: OpenConfirm
      BGP Peer Up count: 1
      Received total routes: 0
      Received active routes total: 0
      Received mac routes: 0
      Advertised total routes: 0
      Port:  Local - 179      Remote - 52876
      Configured: Connect-retry Time: 32 sec
      Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
      Received  : Active Hold Time: 180 sec
      Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
      Peer optional capabilities:
      Peer supports bgp multi-protocol extension
      Peer supports bgp route refresh capability
      Peer supports bgp 4-byte-as capability
      Address family IPv4 Unicast: advertised and received
Received: Total 59 messages
            Update messages              0
            Open messages                2
            KeepAlive messages           57
            Notification messages        0
            Refresh messages             0
Sent: Total 79 messages
            Update messages              5
            Open messages                2
            KeepAlive messages           71
            Notification messages        1
            Refresh messages             0
Authentication type configured: None
Last keepalive received: 2011/09/25 16:41:19
Last keepalive sent    : 2011/09/25 16:41:22
Last update    received: 2011/09/25 16:11:28
Last update    sent    : 2011/09/25 16:11:32
Minimum route advertisement interval is 30 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
```



```
GTSM has been enabled, valid-ttl-hops: 1
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

You can view that GTSM is enabled, the valid hop count is 1, and the BGP connection is in the Established state.

Step 6 Configure GTSM on Router B and Router C. Router B and Router C are directly connected, so the range of the TTL value between the two routers is [255, 255]. The value of **valid-ttl-hops** is 1.

Configure GTSM on Router B.

```
[RouterB-bgp] peer 3.3.3.9 valid-ttl-hops 1
```

Configure GTSM of the IBGP connection on Router C.

```
[RouterC-bgp] peer 2.2.2.9 valid-ttl-hops 1
```

View the GTSM configuration.

```
[RouterB-bgp] display bgp peer 3.3.3.9 verbose
  BGP Peer is 3.3.3.9,  remote AS 20
  Type: IBGP link
  BGP version 4, Remote router ID 3.3.3.9

  Update-group ID : 0
  BGP current state: Established, Up for 00h54m36s
  BGP current event: KATimerExpired
  BGP last state: OpenConfirm
  BGP Peer Up count: 1
  Received total routes: 0
  Received active routes total: 0
  Received mac routes: 0
  Advertised total routes: 0
  Port:  Local - 54998    Remote - 179
  Configured: Connect-retry Time: 32 sec
  Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Received  : Active Hold Time: 180 sec
  Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Peer optional capabilities:
  Peer supports bgp multi-protocol extension
  Peer supports bgp route refresh capability
  Peer supports bgp 4-byte-as capability
  Address family IPv4 Unicast: advertised and received
Received: Total 63 messages
          Update messages              0
          Open messages                1
          KeepAlive messages           62
          Notification messages        0
          Refresh messages             0
Sent: Total 69 messages
          Update messages              10
          Open messages                1
          KeepAlive messages           58
          Notification messages        0
          Refresh messages             0
Authentication type configured: None
Last keepalive received: 2011/09/25 16:46:19
Last keepalive sent    : 2011/09/25 16:46:21
Last update    received: 2011/09/25 16:11:28
Last update    sent    : 2011/09/25 16:11:32
Minimum route advertisement interval is 15 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Nexthop self has been configured
Connect-interface has been configured
```



```
GTSM has been enabled, valid-ttl-hops: 1
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

You can view that GTSM is enabled, the valid hop count is 1, and the BGP connection is in the Established state.

Step 7 Configure GTSM on Router C and Router D. Router C and Router D are directly connected, so the range of the TTL value between the two routers is [255, 255]. The value of **valid-ttl-hops** is 1.

Configure GTSM of the IBGP connection on Router C.

```
[RouterC-bgp] peer 4.4.4.9 valid-ttl-hops 1
```

Configure GTSM of the IBGP connection on Router D.

```
[RouterD-bgp] peer 3.3.3.9 valid-ttl-hops 1
```

Check the GTSM configuration.

```
[RouterC-bgp] display bgp peer 4.4.4.9 verbose
      BGP Peer is 4.4.4.9,  remote AS 20
      Type: IBGP link
      BGP version 4, Remote router ID 4.4.4.9

      Update-group ID : 1
      BGP current state: Established, Up for 00h56m06s
      BGP current event: KATimerExpired
      BGP last state: OpenConfirm
      BGP Peer Up count: 1
      Received total routes: 0
      Received active routes total: 0
      Received mac routes: 0
      Advertised total routes: 0
      Port:  Local - 179      Remote - 53758
      Configured: Connect-retry Time: 32 sec
      Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
      Received  : Active Hold Time: 180 sec
      Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
      Peer optional capabilities:
      Peer supports bgp multi-protocol extension
      Peer supports bgp route refresh capability
      Peer supports bgp 4-byte-as capability
      Address family IPv4 Unicast: advertised and received
Received: Total 63 messages
              Update messages              0
              Open messages                1
              KeepAlive messages           62
              Notification messages        0
              Refresh messages             0
Sent: Total 63 messages
              Update messages              0
              Open messages                2
              KeepAlive messages           61
              Notification messages        0
              Refresh messages             0
Authentication type configured: None
Last keepalive received: 2011/09/25 16:47:19
Last keepalive sent    : 2011/09/25 16:47:21
Last update    received: 2011/09/25 16:11:28
Last update    sent    : 2011/09/25 16:11:32
Minimum route advertisement interval is 15 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Connect-interface has been configured
GTSM has been enabled, valid-ttl-hops: 1
```



```
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

You can view that GTSM is enabled, the valid hop count is 1, and the BGP connection is in the Established state.

Step 8 Configure GTSM on Router B and Router D. Router B and Router D are connected by Router C, so the range of the TTL value between the two routers is [254, 255]. The value of **valid-ttl-hops** is 2.

Configure GTSM of the IBGP connection on Router B.

```
[RouterB-bgp] peer 4.4.4.9 valid-ttl-hops 2
```

Configure GTSM on Router D.

```
[RouterD-bgp] peer 2.2.2.9 valid-ttl-hops 2
```

Check the GTSM configuration.

```
[RouterB-bgp] display bgp peer 4.4.4.9 verbose
  BGP Peer is 4.4.4.9,  remote AS 20
  Type: IBGP link
  BGP version 4, Remote router ID 4.4.4.9

  Update-group ID : 0
  BGP current state: Established, Up for 00h57m48s
  BGP current event: RecvKeepalive
  BGP last state: OpenConfirm
  BGP Peer Up count: 1
  Received total routes: 0
  Received active routes total: 0
  Received mac routes: 0
  Advertised total routes: 0
  Port:  Local - 53714    Remote - 179
  Configured: Connect-retry Time: 32 sec
  Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Received  : Active Hold Time: 180 sec
  Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Peer optional capabilities:
  Peer supports bgp multi-protocol extension
  Peer supports bgp route refresh capability
  Peer supports bgp 4-byte-as capability
  Address family IPv4 Unicast: advertised and received
Received: Total 72 messages
          Update messages              0
          Open messages                1
          KeepAlive messages           71
          Notification messages        0
          Refresh messages             0
Sent: Total 82 messages
          Update messages              10
          Open messages                1
          KeepAlive messages           71
          Notification messages        0
          Refresh messages             0
Authentication type configured: None
Last keepalive received: 2011/09/25 16:47:19
Last keepalive sent    : 2011/09/25 16:47:21
Last update    received: 2011/09/25 16:11:28
Last update    sent    : 2011/09/25 16:11:32
Minimum route advertisement interval is 15 seconds
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Nexthop self has been configured
Connect-interface has been configured
GTSM has been enabled, valid-ttl-hops: 2
```



```
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

You can view that GTSM is configured, the valid hop count is 2, and the BGP connection is in the Established state.

NOTE

- In this example, if the value of **valid-ttl-hops** of either Router B or Router D is smaller than 2, the IBGP connection cannot be set up.
- GTSM must be configured on the two ends of the BGP connection.

Step 9 Verify the configuration.

Run the **display gtsm statistics all** command on Router B to check the GTSM statistics of Router B. By default, Router B does not discard any packet when all packets match the GTSM policy.

```
[RouterB-bgp] display gtsm statistics all
GTSM Statistics Table
------------------------------------------------------------------
SlotId  Protocol  Total Counters  Drop Counters  Pass Counters
------------------------------------------------------------------
0       BGP       17              0              17
0       BGPv6     0               0              0
0       OSPF      0               0              0
0       LDP       0               0              0
1       BGP       0               0              0
1       BGPv6     0               0              0
1       OSPF      0               0              0
1       LDP       0               0              0
2       BGP       0               0              0
2       BGPv6     0               0              0
2       OSPF      0               0              0
2       LDP       0               0              0
3       BGP       0               0              0
3       BGPv6     0               0              0
3       OSPF      0               0              0
3       LDP       0               0              0
4       BGP       32              0              32
4       BGPv6     0               0              0
4       OSPF      0               0              0
4       LDP       0               0              0
5       BGP       0               0              0
5       BGPv6     0               0              0
5       OSPF      0               0              0
5       LDP       0               0              0
7       BGP       0               0              0
7       BGPv6     0               0              0
7       OSPF      0               0              0
7       LDP       0               0              0
------------------------------------------------------------------
```

If the host simulates the BGP packets of Router A to attack Router B, the packets are discarded because their TTL value is not 255 when reaching Router B. In the GTSM statistics of Router B, the number of dropped packets increases accordingly.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
```

```
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
#
bgp 10
 router-id 1.1.1.9
 peer 10.1.1.2 as-number 20
 peer 10.1.1.2 valid-ttl-hops 1
 #
 ipv4-family unicast
  undo synchronization
  peer 10.1.1.2 enable
#
return
```

- Configuration file of Router B

```
#
 sysname RouterB
#
interface GigabitEthernet1/0/0
 ip address 10.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 20.1.1.1 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.9 255.255.255.255
#
bgp 20
 router-id 2.2.2.9
 peer 3.3.3.9 as-number 20
 peer 3.3.3.9 valid-ttl-hops 1
 peer 3.3.3.9 connect-interface LoopBack0
 peer 4.4.4.9 as-number 20
 peer 4.4.4.9 valid-ttl-hops 2
 peer 4.4.4.9 connect-interface LoopBack0
 peer 10.1.1.1 as-number 10
 peer 10.1.1.1 valid-ttl-hops 1
#
 ipv4-family unicast
  undo synchronization
  peer 3.3.3.9 enable
  peer 3.3.3.9 next-hop-local
  peer 4.4.4.9 enable
  peer 4.4.4.9 next-hop-local
  peer 10.1.1.1 enable
#
ospf 1
 area 0.0.0.0
  network 20.1.1.0 0.0.0.255
  network 2.2.2.9 0.0.0.0
#
return
```

- Configuration file of Router C

```
#
 sysname RouterC
#
interface GigabitEthernet1/0/0
 ip address 20.1.1.2 255.255.255.0
#
interface GigabitEthernet2/0/0
 ip address 20.1.2.1 255.255.255.0
#
interface LoopBack0
 ip address 3.3.3.9 255.255.255.255
#
bgp 20
 router-id 3.3.3.9
 peer 2.2.2.9 as-number 20
```



```
 peer 2.2.2.9 valid-ttl-hops 1
 peer 2.2.2.9 connect-interface LoopBack0
 peer 4.4.4.9 as-number 20
 peer 4.4.4.9 valid-ttl-hops 1
 peer 4.4.4.9 connect-interface LoopBack0
#
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.9 enable
  peer 4.4.4.9 enable
#
ospf 1
 area 0.0.0.0
  network 20.1.2.0 0.0.0.255
  network 20.1.1.0 0.0.0.255
  network 3.3.3.9 0.0.0.0
#
return
```

- Configuration file of Router D

```
#
 sysname RouterD
#
interface GigabitEthernet1/0/0
 ip address 20.1.2.2 255.255.255.0
#
interface LoopBack0
 ip address 4.4.4.9 255.255.255.255
#
bgp 20
 router-id 4.4.4.9
 peer 2.2.2.9 as-number 20
 peer 2.2.2.9 valid-ttl-hops 2
 peer 2.2.2.9 connect-interface LoopBack0
 peer 3.3.3.9 as-number 20
 peer 3.3.3.9 valid-ttl-hops 1
 peer 3.3.3.9 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.9 enable
  peer 3.3.3.9 enable
#
ospf 1
 area 0.0.0.0
  network 20.1.2.0 0.0.0.255
  network 4.4.4.9 0.0.0.0
#
return
```

9.20.16 Example for Configuring BFD for BGP4+

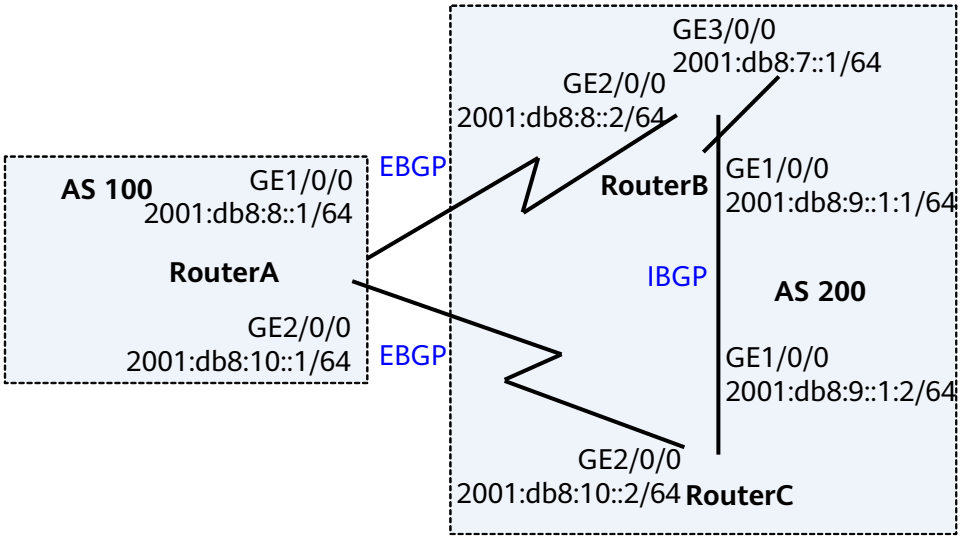
After BFD for BGP4+ is configured, BFD can fast detect the fault on the link between BGP4+ peers and notify it to BGP4+ so that service traffic can be transmitted through the backup link.

Networking Requirements

- As shown in **Figure 9-41**, Router A belongs to AS 100, Router B to AS 200, and Router C to AS 200. Establish an EBGP connection between Router A and Router B and that between Router A and Router C.
- Traffic is transmitted on the active link Router A → Router B. The link Router A → Router C → Router B acts as the standby link.

- Use BFD to detect the BGP session between Router A and Router B. When the link between Router A and Router B fails, BFD can rapidly detect the failure and notify BGP of the failure. Traffic is transmitted on the standby link.

Figure 9-41 Networking diagram of configuring BFD for BGP4+



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the basic BGP4+ functions on each router.
2. Configure MED attributes to control the routing selection of the routers.
3. Enable the BFD 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 the basic BGP4+ functions. Establish an EBGP connection between Router A and Router B, that between Router A and Router C. 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 2001:db8:8::2 as-number 200
[RouterA-bgp] peer 2001:db8:10::2 as-number 200
[RouterA-bgp] ipv6-family unicast
[RouterA-bgp-af-ipv6] peer 2001:db8:8::2 enable
[RouterA-bgp-af-ipv6] peer 2001:db8:10::2 enable
[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 2001:db8:8::1 as-number 100
[RouterB-bgp] peer 2001:db8:9::1:2 as-number 200
[RouterB-bgp] ipv6-family unicast
[RouterB-bgp-af-ipv6] peer 2001:db8:8::1 enable
[RouterB-bgp-af-ipv6] peer 2001:db8:9::1:2 enable
[RouterB-bgp-af-ipv6] network 2001:db8:7::1 64
[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 2001:db8:10::1 as-number 100
[Routerc-bgp] peer 2001:db8:9::1:1 as-number 200
[RouterC-bgp] ipv6-family unicast
[RouterC-bgp-af-ipv6] peer 2001:db8:10::1 enable
[RouterC-bgp-af-ipv6] peer 2001:db8:9::1:1 enable
[RouterC-bgp-af-ipv6] quit
[RouterC-bgp] quit
```

Display the established BGP neighbors on Router A.

```
[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

  2001:db8:8::2   4         200       12       11     0 00:07:26 Established   0
  2001:db8:10::2  4         200       12       12     0 00:07:21 Established   0
```

Step 3 Configure MED attributes.

Set the value of MED sent by Router B and Router C to Router A by using the policy.

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 2001:db8:8::1 route-policy 10 export
[RouterB-bgp-af-ipv6] quit
[RouterB-bgp] quit
```

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 2001:db8:10::1 route-policy 10 export
[RouterC-bgp-af-ipv6] quit
[RouterC-bgp] quit
```

Display all BGP routing information 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  : 2001:db8:7::                    PrefixLen : 64
     NextHop  : 2001:db8:8::2                   LocPrf    :
     MED      : 100                         PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
 *
     NextHop  : 2001:db8:10::2                   LocPrf    :
     MED      : 150                         PrefVal   : 0
     Label    :
     Path/Ogn : 200  i
```

As shown in the BGP routing table, the next hop address of the route to 2001:db8:7::1/64 is 2001:db8:8::2 and traffic is transmitted on the active link Router A → Router B.

Step 4 Configure the BFD detection function, the interval for sending the packets, the interval for receiving the packets, and the local detection time multiple.

Enable BFD on Router A, set the minimum interval for sending the packets and the minimum interval for receiving the packets to 100 ms, and set the local detection time multiple to 4.

```
[RouterA] bfd
[RouterA-bfd] quit
[RouterA] bgp 100
[RouterA-bgp] peer 2001:db8:8::2 bfd enable
[RouterA-bgp] peer 2001:db8:8::2 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
[RouterA-bgp] quit
```

Enable BFD on Router B, set the minimum interval for sending the packets and the minimum interval for receiving the packets to 100 ms, and set the local detection time multiple to 4.

```
[RouterB] bfd
[RouterB-bfd] quit
[RouterB] bgp 200
[RouterB-bgp] peer 2001:db8:8::1 bfd enable
[RouterB-bgp] peer 2001:db8:8::1 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
[RouterB-bgp] quit
```

Display all BFD sessions set up by BGP on Router A.

```
[RouterA] display bgp ipv6 bfd session all
 Local_Address  : 2001:db8:8::1
 Peer_Address   : 2001:db8:8::2
 Tx-interval(ms): 100         Rx-interval(ms): 100
 Multiplier     : 4           Interface      : GigabitEthernet1/0/0
 LD/RD          : 8192/8192   Session-State  : Up
 Wtr-interval(m):0
```

Step 5 Verify the Configuration.

Run the **shutdown** command on GE 2/0/0 of Router B to simulate the active link failure.

```
[RouterB] interface gigabitethernet 2/0/0
[RouterB-GigabitEthernet2/0/0] shutdown
```

Step 6 # Display the 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  : 2001:db8:7::                    PrefixLen : 64
    NextHop  : 2001:db8:10::2                  LocPrf    :
    MED      : 150                           PrefVal   : 0
    Label    :
    Path/Ogn : 200  i
```

As shown in the BGP routing table, the standby link Router A → Router C → Router B takes effect after the active link fails. The next hop address of the route to 2001:db8:7::1/64 becomes 2001:db8:10::2.

----End

Configuration Files

- Configuration file of Router A

```
#
 sysname RouterA
#
 ipv6
#
 bfd
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001:db8:8::1/64
#
interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001:db8:10::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 2001:db8:8::2 as-number 200
 peer 2001:db8:8::2 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
 peer 2001:db8:8::2 bfd enable
 peer 2001:db8:10::2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  peer 2001:db8:8::2 enable
  peer 2001:db8:10::2 enable
#
return
```

- Configuration file of Router B

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



```
 bfd
#
interface interface GigabitEthernet2/0/0
 shutdown
 ipv6 enable
 ipv6 address 2001:db8:8::2/64
#
interface GigabitEthernet1/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001:db8:9::1:1/64
#
interface GigabitEthernet3/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001:db8:7::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 2001:db8:8::1 as-number 100
 peer 2001:db8:8::1 bfd min-tx-interval 100 min-rx-interval 100 detect-multiplier 4
 peer 2001:db8:8::1 bfd enable
 peer 2001:db8:9::1:2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
 #
 ipv6-family unicast
  undo synchronization
  network 2001:db8:7:: 64
  peer 2001:db8:8::1 enable
  peer 2001:db8:8::1 route-policy 10 export
  peer 2001:db8:9::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 2001:db8:9::1:2/64
#
interface interface GigabitEthernet2/0/0
 undo shutdown
 ipv6 enable
 ipv6 address 2001:db8:10::2/64
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 200
 router-id 3.3.3.3
 peer 2001:db8:9::1:1 as-number 200
 peer 2001:db8:10::1 as-number 100
 #
 ipv4-family unicast
  undo synchronization
```



```
 #
 ipv6-family unicast
  undo synchronization
  peer 2001:db8:9::1:1 enable
  peer 2001:db8:10::1 enable
  peer 2001:db8:10::1 route-policy 10 export
#
route-policy 10 permit node 10
 apply cost 150
#
return
```

9.21 FAQ About BGP

9.21.1 Why Are Loopback Addresses Used to Establish BGP Peer Relationships?

Loopback interfaces are logical interfaces. Compared with physical interfaces, loopback interfaces are not affected by links and can reduce the Border Gateway Protocol (BGP) flapping.

9.21.2 Why Is the BGP Connection Not Interrupted Immediately After the Interfaces Connecting Two Peers Are Shut Down?

When External Border Gateway Protocol (EBGP) peers are directly connected and the **ebgp-interface-sensitive** command is run in the Border Gateway Protocol (BGP) view, the BGP peer relationship is interrupted immediately after the interfaces connecting the two peers are shut down. By default, the **ebgp-interface-sensitive** command is run in the BGP view. In other cases, the BGP peer relationship is not interrupted until the Hold timer times out.

9.21.3 Why Are All the Next Hop Addresses in the BGP Routing Table Displayed as 0.0.0.0 After BGP Imports Routes from Other Protocols?

Only routes that exist in the Border Gateway Protocol (BGP) routing table can be imported by the BGP. When importing such routes, BGP does not add new routes, but increases the reference count based on the routing table.

