Routing Configuration

Table of Contents

Chapter 2 Configuring RIP

1.1 Overview

The section describes how to configure the RIP. For details about RIP commands, refer to the setion "RIP Commands" in "Network Protocol Command Reference".

The routing information protocol (RIP) is still a commonly used interior gateway protocol (IGP), mainly applied to small-scale networks of the same type. RIP is a classical distance vector routing protocol, which appears in RFC 1058.

RIP uses the broadcast of the UDP packet to exchange the routing information. In the routing switch, the update of the routing information is performed every 30 seconds. If a switch does not receive the update information from the neighboring switches in 180 seconds, the switch is to label the route in the routing table from the neighboring switch as "unavailable". If the update information is still not received in the following 120 seconds, the switch will delete the route from the routing table.

RIP uses the hop count to balance the weight of different routes. The hop count is the number of switches that a packet gets through from the information source and the information sink. The routing weight of the directly-connected network is **0**. The routing weight of the unreachable network is **16**. Because the range of RIP-using routing weight is small, it is not suitable for the large-scale network.

If the switch has a default route, the RIP declares the route to the pseudo-network 0.0.0.0. In fact, network 0.0.0.0 does not exist. It is just used in RIP to realize the default route. If RIP learns a default route, or the default gateway and the default weight are configured in a switch, the switch is to declare the default network.

RIP sends the routing update information to the designated network interface. If the network that the interface resides is not designated, the network cannot be declared in any RIP update information.

The RIP-2 of our switches supports plain text, MD5 authentication, routing summary, CIDR and VLSM.

1.2 Configuring RIP Task List

To configure RIP, the following tasks must be complete first. The task to activate RIP is mandatory, while other tasks are optional.

- Starting up RIP

- Allowing RIP routing to update the single program broadcast

- Applying the offset to the routing weight

- Adjusting the timer

- Specifying the RIP version number

- Activating RIP authentication

- Forbidding routing summary

- Forbidding the authentication of the source IP address

- Configuring the maximum number of routes

- Activating or forbidding horizon split.

- Monitoring and maintaining RIP

1.3   Configuring RIP Tasks

1.3.1   Starting up RIP

Run the following command in global configuration mode to activate RIP:

Command	Purpose
router rip	Activates the RIP routing process and enters the switch configuration mode.
network *network-number* <network-mask>	Specifies the network number related to the RIP routing process.

1.3.2   Allowing RIP Routing to Update the Single-Program Broadcast

Normally, RIP is a broadcast protocol. To enable the RIP routing update to reach the non-broadcast network, you must configure the switch to enable it to exchange the routing information.

Run the following command in switch configuration mode to enable the routing information exchange:

Command	Purpose
neighbor *ip-address*	Defines a neighboring switch to exchange routing information with the known switch.

Additionally, you can run **ip rip passive** to specify ports to forbid sending the route update information.

1.3.3   Applying the Offset to the Routing Weight

The offset list is used to add an offset for the outgoing routes or the incoming routes learned by the RIP. It provides a local mechanism to add the routing weight. You also can use the access list or the interface to limit the offset list. Run the following command in switch configuration mode to add the routing weight.

Command	Purpose
offset〔[interface-type number]|* 〕{**in**|**out**}	Adds an offset for the routing weight.

access-list-name offset	

1.3.4 Adjusting the Timer

The routing protocol uses several timers to judge the frequency of sending route update information, how much time is needed for the route to become ineffective and other parameters. You can adjust these timers to improve the performance of the routing protocol.

You also can adjust the routing protocol to speed up the convergent time of all IP routing arithmetic, rapidly backing up the redundancy switch and ensuring the minimum breakdown time in case of quick recovery. Run the following command in switch configuration mode to adjust the timer:

Command	Purpose
timers holddown *value*	It means how much time is needed for a route to be deleted from the routing table.
timers expire *value*	It means what interval is needed for a route to be declared ineffective.
timers update *value*	It means the transmission frequency of the routing update information.

1.3.5 Specifying the RIP Version Number

The RIP-2 of our switches supports authentication, PIN management, routing summary, CIDR and VLSM. By default, the switch receives RIP-1 and RIP-2, but the switch only sends RIP-1. Through configuration, the switch can receive and send only the packet RIP-1, or only the packet RIP-2. To meet the previous demand, run the following command in switch configuration mode:

Command	Purpose
version {1	2}	The switch sends and receives only RIP-1 or only RIP-2.

The previous tasks control the default actions of the RIP. You also can configure a certain interface to change the default actions.

Run the following commands in VLAN configuration mode to control the interface whether to send RIP-1 or RIP-2.

Command	Purpose
ip rip send version 1	The configured interface only sends RIP-1.
ip rip send version 2	The configured interface only sends RIP-2.
ip rip send version compatibility	Sends the RIP-2 update message in the form of broadcast.

Run the following commands in interface configuration mode to control the interface whether to receive packet RIP-1 or packet RIP-2

Command	Purpose
ip rip receive version 1	The configured interface only receives RIP-1.

ip rip receive version 2	The configured interface only receives RIP-2.
ip rip receive version 1 2	The configured interface receives RIP-1 and RIP-2.

1.3.6 Activating RIP Authentication

RIP-1 does not support authentication. To receive and send the RIP-2 packet, you can activate the RIP authentication on the interface.

On the activated interface, two authentication modes are provided: plain text authentication and MD5 authentication. Each RIP-2 packet uses the plain authentication by default.

Note:

For the purpose of security, do not use the plain authentication in the RIP packet because the unencrypted authentication PIN is sent to each RIP-2 packet. You can use the plain authentication without security concern.

Run the following commands in VLAN configuration mode to configure the RIP plain text authentication.

Command	Purpose
ip rip authentication simple	Configures the interface to use the plain authentication.
ip rip password [string]	Configures the PIN of the plain authentication.

Run the following commands in interface configuration mode to configure the MD5 authentication of the RIP:

Command	Purpose
ip rip authentication message-digest	Configures the interface to use the MD5 authentication.
ip rip message-digest-key [*key-ID*] **md5** [*key*]	Configures the PIN and ID of the md5 authentication.

1.3.7 Forbidding Routing summary

RIP-2 supports the automatic routing summary by default. RIP-2 routes are collected when passing the boundaries of different networks. The RIP-1 automatic collection function is always in **positive** state.

If there is a separated subnet, you need to forbid the routing summary function to declare the subnet. If the routing summary function is disabled, the switch is to send the routing information of the subnet and the host when passing through the boundaries of different networks. Run the following command in switch configuration mode to disable the automatic routing summary function.

Command	Purpose
no auto-summary	Disables the automatic routing summary function.

1.3.8 Forbidding the Authentication of the Source IP Address

By default, the switch authenticates the source IP address in the RIP routing update information. If the address is illegal, the routing update is dropped.

When a switch wants to receive its own update information and the network and neighbor are not configured on the switch of the receiving side, you can forbid the authentication of the source IP address. Normally, you are not recommended to use the command.Run the following command in switch configuration mode to forbid authenticating the source IP address of the incoming routing information:

Command	Purpose
no validate-update-source	Forbids authenticating the source IP address of the incoming routing information.

1.3.9 Configuring the Maximum Number of Routes

By default, the local RIP routing table contains up to 1024 routes. When the route number exceeds the maximum number, you cannot add new routes to the routing table. At the same time, the system notifies you that the route number has already reached the maximum number set for the routing table.Run the following command in switch configuration mode to configure the maximum number of routes for the local RIP routing table:

Command	Purpose
maximum-count *number*	Configures the maximum number of routes for the local RIP routing table.
no maximum-count	Resumes the default maximum number of routes.

1.3.10 Activating or Forbidding Horizon Split

Normally, the switch that connects to the broadcast IP network and adopts the distance vector routing protocol adopts the horizon split to reduce the possibility of the routing loop. The information about the routing loop of horizon split declares itself to the interface that receives the routing information. In this way, the communication among multiple routing switches is improved, especially when the loop breaks. However, it is not so good as to the non-broadcast network. At this time, you may forbid the horizon split.

If the assistant IP address is configured on the interface and the horizon split is activated, the source IP address of the routing update may not conclude all assistant addresses. The source IP address of one routing update contains only one network number.

Run the following commands in VLAN configuration mode to activate or forbid the horizon split.

Command	Purpose
ip rip split-horizon	Activates the horizon split.
no ip rip split-horizon	Forbids the horizon split.

By default, the horizon split is activated on the point-to-point interface; the point-to-multiple interface is forbidden.

Foe details, refer to the section "Horizon Split Example".

Note:

In normal case, do not change the default configuration unless you are sure that the programs need to change states. Remember that if the horizon split is forbidden in a serial port that connects a packet switching network, you must forbid the horizon split in the switches in relative multiple-program group of a network.

1.3.11 Monitoring and Maintaining RIP

Monitoring and maintaining RIP needs to display network statistics information, such as RIP parameter configuration, real-time network track. These information help you judge the network usage, solve network problem and the reachabilitiy of network nodes.

Run the following commands in management mode to display all routing statistics information:

Command	Purpose
show ip rip	Display the current state of the RIP protocol.
show ip rip database	Displays all RIP routes.
show ip rip protocol	Displays all RIP-relative information.

Run the following commands in management mode to track routing protocol information:

Command	Purpose
debug ip rip database	Tracks information about adding RIP route to the routing table, deleting route from the routing table and changing route.
debug ip rip protocol	Tracks RIP message.

1.4   RIP Configuration Example

Device A and device B are configured as follows:

Device A:

interface vlan 11
ip address 192.168.20.81 255.255.255.0
!
interface loopback 0
ip address 10.1.1.1 255.0.0.0
!
router rip
network 192.168.20.0

network 10.0.0.0
!

Device B:

interface vlan 11
ip address 192.168.20.82 255.255.255.0
interface loopback 0
ip address 20.1.1.1 255.0.0.0
!
router rip
network 192.168.20.0
network 20.0.0.0
!

Chapter 2 Configuring BEIGRP

2.1 Overview

Technologies used by BEIGRP are similar to the distance vector protocol:

- The router makes routing decision according to the information provided by the directly-connecting neighbor;

- The router provides its routing information to its directly-connecting neighbor.

However, BEIGRP has more advantages compared with the distance vector protocol:

- BEIGRP saves all routes sent by all neighbors in the topology, not just saving the best route received up to now.

- BEIGRP can query neighbors when it cannot access the destination and has no replaceable route. Therefore, the convergence speed of BEIGRP is as fast as that of the best-link-state protocol.

Diffused Update Algorithm (DUAL) of BEIGRP is the core reason why BEIGRP is better than other traditional distance vector protocols. It always in **active** state and queries the neighbors when it cannot access the destination and there is no replaceable route. Therefore, the collection speed of BEIGRP is rapid.

BEIGRP is a special transmission protocol designed on the basis of EIGRP requirements. BEIGRP is created on the IP protocol. The following requirements are satisfied by BEIGRP:

- The disappearance of new or old neighbors is dynamically detected through the hello message.

- All data transmission is reliable.

- The transmission protocol allows the single-program or multiple-program transmission.

- The transmission protocol can adapt to the change of network conditions and neighbor response.

- BEIGRP can limit its bandwidth occupancy rate according to requirements.

2.2 BEIGRP Configuration Task List

The BEIGRP configuration includes the following tasks. Among the tasks, the task to activate the BEIGRP is mandatory; other tasks can be selectively performed according to requirements.

- Activating BEIGRP

- Configuring bandwidth occupancy percent

- Regulating account coefficient of BEIGRP compound distance

- Regulating the compound distance through offset

- Disabling automatic route summary

- Customizing route summary

- Configuring forwarding route

- Configuring other BEIGRP parameters

- Monitoring and maintaining the running of BEIGRP

2.2   BEIGRP Configuration Task

2.3.1   Activating BEIGRP

Perform the following operations to create a BEIGRP process:

Command	Purpose
router beigrp *as-number*	Adds a BEIGRP process in global configuration mode.
network *network-number network-mask*	Adds network segment to the BEIGRP process in route configuration mode.

After the above configuration is complete, BEIGRP starts to run on all interfaces of the network segment. BEIGRP finds new neighbors through hello message and interacts original routes through update information.

2.3.2   Configuring Bandwidth Occupancy Percent

In default state, BEIGRP occupies up to 50% of bandwidth. You can run the following command in VLAN interface configuration mode to adjust the bandwidth that can be used by BEIGRP.

Command	Purpose
ip beigrp bandwidth-percent *percent*	Configures the maximum bandwidth percent for the BEIGRP message.

2.3.3   Regulating Coefficient of BEIGRP Compound Distance

In some cases, the coefficient of BEIGRP compound distance need be regulated, which finally affects the routing strategy. Though the default coefficient used by BEIGRP is suitable for most network conditions, you need to regulate it in some special cases. The regulation may cause great change of the whole network. Be careful when you perform this regulation.

Run the following command in route configuration mode:

Command	Purpose
metric weights *k1 k2 k3 k4 k5*	Regulates the coefficient of the BEIGRP compound distance.

2.3.4 Regulating the Compound Distance Through Offset

You can add all incoming and outgoing routes purposively according to requirements using the offset table, or add compound distances of several suitable routes. The purpose is to affect the routing result of the router. In the configuration process, you can selectively specify the access list or the application interface in the offset list to further confirm routes which the offset is added to.

Command	Purpose
offset{type number	*} {**in**	**out**} access-list-name offset	Applies a offset table.

2.3.5 Disabling Automatic Route summary

The automatic collection of BEIGRP is different from that of other dynamic routing protocols. It complies with the following regulations:

● When multiple networks in a BEIGRP process are defined, a summary route of the defined network is generated if at least one subnet of the network is in the BEIGRP topology table.

● The created summary route is oriented to the Null0 interface has the minimum distance of all subnets. The summary route is also added to the main IP routing table. Its management distance is 5 (which cannot be configured).

● When the update information is sent to neighbors in different main IP networks, the subnet of the summary route of rule 1 and rule 2 is canceled. Only the summary route is sent.

● Subnets that do not belong to any network defined in the BEIGRP procedure are not be collected.

In some network conditions, you may hope to notify neighbors of each detailed route. In this case, you need run the following command:

Command	Purpose
no auto-summary	Disables the automatic routing summary.

2.3.6 Customizing Routing summary

When the automatic routing summary cannot meet the requirements, you can configure routing summary on every interface where BEIGRP runs, and specify the destination network segments that are to perform routing summary. The interfaces where routing summary is configured will not send any detailed routing update information that belongs to the routing summary network segment. Other interfaces do not get affected.

The routing summary operations comply with the following regulations:

● After a routing summary command is configured on an interface, a summary route of the defined network is generated if at least one subnet of the network is in the BEIGRP topology table.

● The created summary route is oriented to the Null0 interface has the minimum distance of all subnets. The summary route is also added to the main IP routing table. Its management distance is 5 (which cannot be configured).

● When the routing update information is sent on the interface where routing summary is configured, the detailed routes belonging to routing summary network segment are to cancelled. Other routing update information will not be affected.

Command	Purpose
ip beigrp summary-address *ip address address mask*	Configures routing summary on the interface.

2.3.7 Configuring Forwarding Route

When BEIGRP forwards other types of routes, BEIGRP complies with the following regulations:

● If the present route is static or directly-connected, the command **default-metric** need not be configured and other compound distance parameters (bandwidth, delay, reliability, effective load and MTU) are directly obtained from the current port.

● If the present routes are routes of other BEIGRP processes, the **default-metric** command need not be configured and its compound distance parameters are directly obtained from the BEIGRP process.

● The **default-metric** command must be configured when routes of other protocols such as rip and ospf are sent. The suitable distance of the route forwarding is determined by the configuration value. If the command is not configured, the route forwarding cannot function.

On the switch where BEIGRP and RIP simultaneously run, to make BEIGRP neighbors learn the routes learned by the RIP protocol in the local switch, run the following command.

Command	Purpose
default-metric bandwidth delay reliability loading mtu	Configures the default vector distance for route forwarding.
redistribute *protocol* [**route-map** *name*]	Forwards routes to the BEIGRP protocol.

2.3.8 Configuring Other BEIGRP Parameters

To adjust to different network conditions and make BEIGRP more efficient, you need modify the following parameters:

● Modify the interval for BEIGRP to send hello message and neighbor timeout time.

● Disable the horizon split.

2.3.8.1 Modify the interval for BEIGRP to send hello message and neighbor timeout time

The following objectives needed by the BEIGRP hello protocol to perform correct BEIGRP operations are listed:

● It can find new accessible neighbor. The detection of neighbor is an automatic process without any manual configuration.

● It authenticates neighbor configuration and only allows communication between neighbors that are configured in compatible mode.

● It continuously monitors neighbor's usability and detects the disappearance of neighbors.

The router sends the hello multiple-program broadcast packet on the interfaces where BEIGRP runs. Each BEIGRP-supporting router receives these multiple-program broadcast packets. Therefore, all neighbors can be found.

The Hello protocol uses two timers to detect the disappearance of neighbors. The hello interval specifies the transmission frequency of the BEIGRP hello message on the interface of the router. **hold timer** specifies the time to declare the neighbor is dead when the router cannot receives data from the designated neighbor. After any type of the BEIGRP packet is received from the neighboring router, the value of **hold timer** needs to be reset.

Different network types or network bandwidth adopt different default values of the hello timer.

Interface Type Packaging		Hello Timer (second)	Hold Timer (second)
LAN Interface	Any	5	15

In the Hello protocol, different default values of the timer may cause BEIGRP neighbors that connect the same IP subnet to use different hello timers or hold timers. To prevent the problem from occurring, you need to specify the hold timer in the hello packet of each router. Each BEIGRP router uses the hold timer specified in the hello packet of the neighboring router to judge whether the neighbor times out. In this way, trouble-detecting timers of different neighbors appears in one WAN topology. In special cases, the default value of the timer cannot fulfill actual requirements. To modify the interval to send the hello message, run the following command:

Command	Purpose
ip beigrp hello-interval *seconds*	Modifies the interval to send the hello message on the interface.

To modify the timeout timer of the neighbor, run the following command:

Command	Purpose
ip beigrp hold-time *seconds*	Modifies the timeout time of the neighbor.

2.3.8.2   Disabling the horizon split

The horizon split function is normally adopted. It prevents a received routing information from broadcasting out from the same interface, avoiding route loop. In some cases, the horizon split function is not the best choice, so you can run the following command to disable the horizon split function:

Command	Purpose
no ip beigrp split-horizon	Disables the horizon split function.

2.3.9 Monitoring and Maintaining BEIGRP

Run the following command to clear the neighboring relationship.

Command	Purpose
clear ip beigrp neighbors [*interface*]	Clear the neighboring relationship.

Run the following commands to display all BEIGRP statistics information:

Command	Purpose
show ip beigrp interfaces [*interface*] [*as-number*]	Displays the information about BEIGRP interface.
show ip beigrp neighbors [*as-number* | *interface*]	Displays the information about BEIGRP neighbors.
show ip beigrp topology [*as-number* | all-link | summary | active]	Displays the information about BEIGRP topology table.

2.4 BEIGRP Configuration Example

In the following example, the summary route that sends network segment 10.0.0.0/8 on VLAN11 is configured. All subnet routs of the network segment will not be notified of the neighbor. At the same time, the automatic summary of the BEIGRP process is disabled.

interface vlan 11

ip beigrp summary-address 1 10.0.0.0 255.0.0.0

!

router beigrp 1

network 172.16.0.0 255.255.0.0

no auto-summary

Chapter 3 Configuring OSPF

3.1 Overview

The chapter describes how to configure the OSPF. For the details of OSPF commands, refer to relative sections about OSPF commands.

OSPF is a IGP routing protocol developed by the OSPF team of IETF. OSPF designed for the IP network supports IP subnets and exterior routing information identifier, message authentication and IP multicast.

The OSPF function of our switches complies with the requirements of OSPF V2 (See RFC2328). The following table lists some key features in reality.

Key Feature	Description
Stub domain	Support the stub domain.
Rout forwarding	Routes that are learned by any routing protocol can be forwarded to other routing protocol domain, which means that OSPF can enter routes that RIP learned in the automatic domain. The routes that OSPF learns also can be exported to RIP. Among the automatic domains, OSPF can enter the routes that BGP learns; OSPF routes also can be exported to BGP.
Authentication	Among neighboring switches in a domain, the plain text and MD5 authentication are supported.
Routing interface parameters	The configurable interface parameters include the output cost, resending interval, interface output delay, the priority of the switch, the interval to judge the shutdown of the switch, the interval of the hello packet and the authentication PIN.
Virtual link	The virtual link is supported.
NSSA area	See RFC 1587.
OSPF in the on-demand circuit	See RFC 1793.

3.2 OSPF Configuration Task List

OSPF requires the routing data exchange among switches, ABR and ASBR in the whole domain. To simplify the configuration, you can make them run in the default settings without authentication. However, if you modify a certain parameter, make sure that the modified parameter is the same on all switches.

You need to complete the following tasks to configure OSPF. The task to activate OSPF is mandatory, while other configurations are optional.

- Starting up OSPF

- Configuring interface parameters of OSPF

- Configuring OSPF in different physical networks

- Configuring OSPF area parameters

- Configuring NSSA domain of OSPF

- Configuring routing summary in the OSPF area

- Configuring forwarded routing summary

- Generating default route

- Choosing route ID on the LOOPBACK interface

- Configuring the management distance of OSPF

- Configuring timer for route calculation

- Monitoring and maintaining OSPF

For route forwarding configuration, refer to relevant content about the IP routing protocol configuration

3.3 OSPF Configuration Task

3.3.1 Starting up OSPF

Similar to other routing protocols, before activating OSPF, you have to create the OSPF routing process. In the creation of the routing process, An IP address range related to the processing and a relevant domain ID need be distributed.

Run the following commands in global configuration mode to start up OSPF:

Command	Purpose
router ospf *process-id*	Activates the OSPF routing protocol and enters the switch configuration mode.
network *address mask* **area** *area-id*	Configures the running interface of OSPF and the relevant interface domain ID.

3.3.2 Configuring Interface Parameters of OSPF

You are allowed to modify OSPF parameters of the interface according to actual requirements. When you modify a parameter, make sure that the parameter on all switches of the interconnected network is same.

Run the following commands in interface configuration mode to configure the interface parameters:

Command	Purpose

ip ospf cost *cost*	Configures the value of the transmission packet on the OSPF interface.
ip ospf retransmit-interval *seconds*	Configures the seconds of LSA resending between neighbors on the same OSPF interface.
ip ospf transmit-delay *seconds*	Configures the time to send LSA on an OSPF interface (unit: second).
ip ospf priority *number*	Configures the priority number for the routing switch to become the OSPF DR routing switch.
ip ospf hello-interval *seconds*	Configures the interval to send the hello packet on the OSPF interface.
ip ospf dead-interval *seconds*	Configures the dead interval. In the prescribed interval, if the hello packet from neighbors is not received, the neighboring switch is considered to be in shutdown state.
ip ospf authentication-key *key*	Represents the authentication password of the neighboring route in a network segment. The OSPF authentication password is adopted.
ip ospf message-digest-key *keyid* md5 *key*	Requires OSPF to use the MD5 authentication.
ip ospf passive	Configures the state of the hello message on a port.

3.3.3 Configuring OSPF in Different Physical Networks

OSPF divides the physical media of the network into the following classes:

- Broadcast network (Ethernet, Token Ring, FDDI)

- Non-broadcast and multi-access network (SMDS, Frame Relay, X.25)

- Point-to-point network (HDLC, PPP)

The X.25 and frame-relay network provides optional broadcast capability. You can configure the OSPF to run in the broadcast network through the **map** command. For details of the map command, refer to the description of the map command in *WAN Command Reference*.

3.3.4 Configuring OSPF Network Type

No matter what physical media type your network belongs to, you can configure your network to be the broadcast network or the non-broadcast and multi-access network. This feature allows you configure the network flexibly. You can configure the broadcast network to the non-broadcast and multi-access network; you also can configure the non-broadcast network, such as X.25, Frame Relay and SMDS, to the broadcast

network. The feature also eases the neighbor's configuration. For details, refer to contents about OSPF configuration of non-broadcast network.

Configuring the non-broadcast and multi-access network to a broadcast network or a non-broadcast network is to suppose that the virtual link exists between two random switches or to suppose that the network is a mesh network. The previous configuration is unreal because it costs too much. You may configure the non-broadcast and multi-access network to a partly meshed network. To save the expense, you can configure the non-broadcast and multi-access network to a point-to-multipoint network. The disconnected switches can exchange the routing information with each other through the virtual link.

The interface connecting the OSPF point to other points is defined as the point-to-multipoint network interface. It creates lots of host routes. Comparing with the non-broadcast and multi-access network or the point-to-point network, the OSPF point-to-multipoint network has the following advantages:

● The point-to-multipoint network is easy to configure. The configuration does not need the neighbor configuration commands. It only needs an IP address and there is no DR.

● The point-to-multipoint network does not need the wholly meshed network's topology, so the expense is smaller.

● It is more reliable. The connection can keep working even if the virtual link fails.

Run the following commands in interface configuration mode to configure the type of the OSPF network.

Command	Purpose
ip ospf network {broadcast | non-broadcast | {point-to-multipoint [non-broadcast] }}	Configures the network type of the OSPF.

The network of the switch is a broadcast network.

3.3.5 Configuring OSPF Area Parameters

The configurable area parameters include authentication, stub area and the value of the default routing summary. The authentication is based on the password protection.

Stub area is an area which exterior routes are not sent to. ABR generates a default exterior route to enter the stub area, enabling stub area to connect exterior networks out of the automatic area. To utilize the feature that OSPF stub supports, the default route must be used in the stub area. To further reduce the LSAs to enter the stub area, you need select the option **No Summary** in the ABR.

Run the following command in switch configuration mode to set area parameters:

Command	Purpose
area *area-id* **authentication simple**	Activates the authentication of the OSPF area.
area *area-id* **authentication message-digest**	Specifies the MD5 authentication as the authentication OSPF.

area *area-id* **stub** [no-summary]	Defines a stub area.
area *area-id* **default-cost** *cost*	Set the value of the default route in the stub area.

3.3.6 Configuring Routing Summary in the OSPF Area

The feature enables the ABR to broadcast a summary route to other areas. In OSPF, ABR is to broadcast every network to other areas. If the network number is distributed serially according to some method, you can configure ABR to broadcast a summary route to other areas. The summary route can cover all networks in a certain range.

Run the following command in switch configuration mode to set the address range:

Command	Purpose
area *area-id* **range** *address mask*	Sets the address range of the summary area.

3.3.7 Configuring Forwarded Routing Summary

When the routes are distributed from other areas to the OSPF area, each route will be uniquely broadcast in the exterior LSA method. However, you can configure the switch to broadcast a route, which can cover a certain address area. This method reduces the size of the OSPF link state database.

Run the following command in switch configuration mode to configure the summary route:

Command	Purpose
summary-address *prefix mask* [not advertise]	Describes the address and mask covering the distributed route. Only one summary route is broadcast.

3.3.8 Generating Default Route

ASBR requires generating a default route to enter the OSPF route area. When you configure the switch to distribute the route to the OSPF area, the route automatically becomes ASBR. However, default ASBR does not generate the default route to enter the OSPF routing area.

Run the following command in switch configuration mode to force ASBR to generate the default route.

Command	Purpose
default-information originate [always] [route-map *map-name*]	Forces ASBR to generate the default route.

3.3.9 Choosing Route ID Through the LOOPBACK Interface

OSPF takes the maximum IP address configured on the interface as the switch ID. If the interface connecting the IP address changes to the Down state, or the IP address

is cancelled, the OSPF process is to recalculate the new switch ID and resend the routing information from all interfaces.

If a **loopback** interface is configured with the IP address, the switch takes its IP address as its ID. The **loopback** interface will never be at the **down** state. Therefore, the routing table is stable.

The switch preferentially takes the **loopback** interface as the switch ID. It also chooses the maximum IP address as the switch ID. If the loopback interface does not exist, the maximum IP address of the switch is taken as the switch ID. You cannot specify OSPF to use any special interface.

Run the following command in global configuration mode to configure the IP loopback interface:

Command	Purpose
interface loopback 0	Creates a loopback interface and enters the interface configuration mode.
ip address *ip-address mask*	Distributes an IP address for an interface.

3.3.10 Configuring OSPF Management Distance

The management distance stands for the credit level of the routing information source, such as the single switch or a group of switches. Generally, the management distance is an integer between 0 and 255. The bigger the number is, the lower the credit level is. If the management distance is 255, the routing information source is not trusted or should be omitted.

OSPF uses three kinds of different management distances: intra-area, inter-area and external. The routes in an area are called **intra-area** routes; routes to other areas are called **inter-area** routes; routes that are distributed from other routing protocol areas are called **external** routes. The default value of each type of routes is 110.

Run the following command in switch configuration mode to configure the distance vale of OSPF.

Command	Purpose
distance ospf [intra-area *dist1*] [inter-area *dist2*] [external *dist3*]	Modifies the management distance value of the intra-area routes, inter-area routes and external routes.

3.3.11 Configuring Timer for Routing Calculation

You can configure the delay between when OSPF receives the topology change information and when the calculation is started. You also can configure the interval of continuously calculating SPF. Run the following command in switch configuration mode.

Command	Purpose
timers delay *delaytime*	Sets the delay of routing calculation in an area.
timers hold *holdtime*	Sets the minimum interval of routing

	calculation in an area.

3.3.12 Monitoring and Maintaining OSPF

The network statistics information includes the content of IP routing table, cache and database. All information help you to judge the usage of network resources, solve network problems, learn the reachability of network nodes and to find routes where packets get through the network.

Run the following commands to display all routing statistics information:

Command	Purpose
Show ip ospf [*process-id*]	Displays the general information of the OSPF process.
Show ip ospf [*process-id*] **database** **show ip ospf** [*process-id*] **database [router]** [*link-state-id*] **show ip ospf** [*process-id*] **database [router] [self-originate]** **show ip ospf** [*process-id*] **database [router] [adv-router** [*ip-address*]] **show ip ospf** [*process-id*] **database [network]** [*link-state-id*] **show ip ospf** [*process-id*] **database [summary]** [*link-state-id*] **show ip ospf** [*process-id*] **database [asbr-summary]** [*link-state-id*] **show ip ospf** [*process-id*] **database [external]** [*link-state-id*] **show ip ospf** [*process-id*] **database [database-summary]**	Displays the relative information about OSPF database.
show ip ospf border-routers	Displays internal items in the routing table between ABR and ASBR.
show ip ospf interface	Displays information about the OSPF interface.
show ip ospf neighbor	Displays information about neighbors of OSPF according to the interface.
debug ip ospf adj	Monitors the procedure of establishing OSPF adjacency.
debug ip ospf events	Monitors the OSPF interface and neighboring events.
debug ip ospf flood	Monitors the flooding of OSPF database.
debug ip ospf lsa-generation	Monitors the LSA generation of OSPF.

debug ip ospf packet	Monitors the OSPF message.
debug ip ospf retransmission	Monitors the message resending of OSPF.
debug ip ospf spf debug ip ospf spf intra debug ip ospf spf inter debug ip ospf spf external	Monitors the SPF calculation route of OSPF.
debug ip ospf tree	Monitors SPF tree establishment of OSPF.

3.4   OSPF Configuration Example

3.4.1 VLSM Configuration Example

OSPF and static routes support VLSMs. Through VLSMs, different masks on different interfaces can use the same network number. The IP address is thus saved and the address space is effectively utilized.

In the following example, a 30-digit subnet mask is used. A 2-digit address space is reserved for the host address of the serial port. Two host addresses are enough for the point-to-point serial link.

interface vlan 10

   ip address 131.107.1.1 255.255.255.0

! 8 bits of host address space reserved for ethernets

interface vlan 11

   ip address 131.107.254.1 255.255.255.252

! 2 bits of address space reserved for serial lines

! Router is configured for OSPF and assigned AS 107

router ospf 107

! Specifies network directly connected to the router

   network 131.107.0.0 0.0.255.255 area 0.0.0.0

3.4.2 OSPF Route and Route Distribution Configuration Example

OSPF demands to exchange information among internal switches, ABR and ASBR. In the minimum configuration, the OSPF-based switch can work with default parameter settings. There is no authentication demand.

The following are three configuration examples:

The first example shows basic OSPF commands.

The second example shows how to configure internal routing switches, ABR and ASBR in an automatic system.

The third example shows how to use all kinds of OSPF tools.

3.4.2.1   Basic OSPF Configuration Example

The following example shows how to configure a simple OSPF. Activate the routing process 9; connect Ethernet interface 0 to area 0.0.0.0; meanwhile, send RIP to OSPF or send OSPF to RIP.

interface vlan 10

ip address 130.130.1.1 255.255.255.0

ip ospf cost 1

!

interface vlan 10

ip address 130.130.1.1 255.255.255.0

!

router ospf 90

network 130.130.0.0 255.255.0.0 area 0

redistribute rip

!

router rip

network 130.130.0.0

redistribute ospf 90

3.4.2.2   Example to Basic Configuration of Internal Routing Switch, ABR and ASBR

In the following example, four area Ids are distributed to four IP address ranges. The routing process 109 is activated. Four areas are area 10.9.50.0, area 0, area 2 and area 3. The masks of areas 10.9.50.0, 2 and 3 are designated with address range. Area 0 includes all networks.

router ospf 109

  network 131.108.20.0 255.255.255.0 area 10.9.50.0

  network 131.108.0.0 255.255.0.0 area 2

  network 131.109.10.0 255.255.255.0 area 3

  network 0.0.0.0 0.0.0.0 area 0

!

! Interface vlan10 is in area 10.9.50.0:

interface vlan 10

  ip address 131.108.20.5 255.255.255.0

!

! Interface vlan11 is in area 2:

interface vlan 11

  ip address 131.108.1.5 255.255.255.0

!

! Interface vlan12 is in area 2:

interface vlan 12

  ip address 131.108.2.5 255.255.255.0

!

! Interface vlan13 is in area 3:

interface vlan 13

  ip address 131.109.10.5 255.255.255.0

```
!
! Interface vlan14 is in area 0:
interface vlan 14
   ip address 131.109.1.1 255.255.255.0
!
! Interface vlan 100 is in area 0:
interface vlan 100
   ip address 10.1.0.1 255.255.0.0
```

The function of network area configuration command has its order, so the sequence of the commands is important. The switch matches the IP address/mask pair according to the order. For details, refer to section *OSPF Commands*.
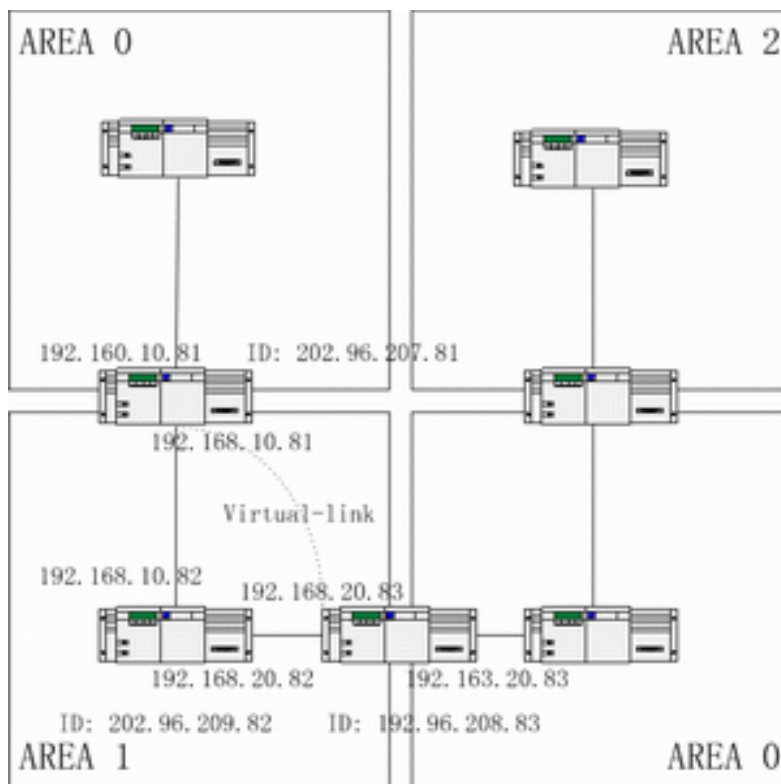
Check the first network area. The interface subnet 131.108.20.0 configured for area ID 10.9.50.0 is 131.108.20.0. The Ethernet interface is configured to 0. The interface is therefore in area 10.9.50.0.

In the second area, if the previous process is adopted to analyze other interfaces, interface 1 is matched. Therefore, interface 1 connects area 2.

Continue matching other network areas. Note that the last network area command is an exception, which means that all the remnant interfaces connect network area 0.

3.4.2.3   Complex Configuration of Interior Switches, ABR and ASBR

The following example shows how to configure multiple switches in a single OSPF automatic system. The following figure shows the network topology of the configuration example.

Configure switches according to the previous figure.

RTA:

interface loopback 0

   ip address 202.96.207.81 255.255.255.0

!

interface vlan 10

   ip address 192.168.10.81 255.255.255.0

!

interface vlan 10

   ip address 192.160.10.81 255.255.255.0

!

router ospf 192

   network 192.168.10.0 255.255.255.0 area 1

   network 192.160.10.0 255.255.255.0 area 0

!

RTB:

interface loopback 0

   ip address 202.96.209.82 255.255.255.252

!

interface vlan 10

   ip address 192.168.10.82 255.255.255.0

!

interface vlan 11

   ip address 192.160.20.82 255.255.255.0

!

router ospf 192

   network 192.168.20.0 255.255.255.0 area 1
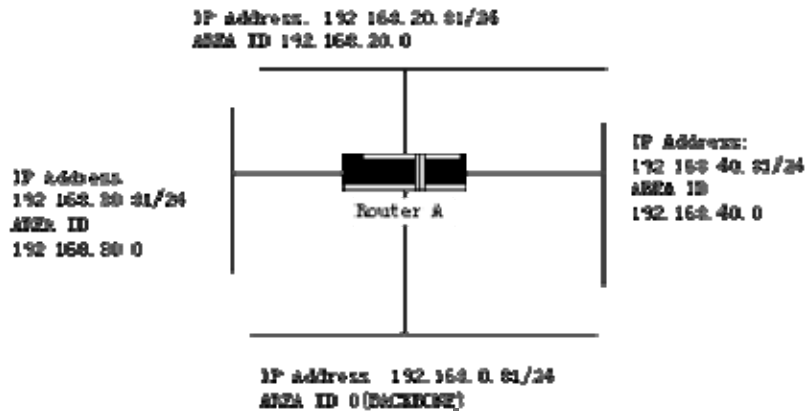
   network 192.168.10.0 255.255.255.0 area 1

!

RTC:

interface loopback 0

ip address 202.96.208.83 255.255.255.252

!

interface vlan 10

ip address 192.163.20.83 255.255.255.0

!

interface vlan 11

ip address 192.160.20.83 255.255.255.0

!

router ospf 192

network 192.168.20.0 255.255.255.0 area 1

network 192.163.20.0 255.255.255.0 area 0

!

3.4.3 Configuring Complex OSPF on ABR Switch

The following case describes ABR configuration tasks.

● Configuring basic OSPF

● Distributing routes

The following figure shows the address range and area distribution.

The following are basic configuration tasks:

(1) Configuring the address range for Ethernets 0 to 3

(2) Activating OSPF on every interface

(3) Setting the authentication password for each area and network

(4) Setting the link state value and other interface parameters

Note:

Use one **area** command respectively to set authentication parameters and stub area. You can use one command to set these parameters.

● Set backbone area (Area 0).

The configuration tasks relative with the distribution are listed in the following:

● Distribute IGRP routes and RIP routes to enter OSPF parameter setting (including metric-type, metric, tag and subnet).

● Distribute IGRP routes and OSPF routes to RIP.

The following is an OSPF configuration example.

```
interface vlan 10
  ip address 192.168.20.81 255.255.255.0
  ip ospf password GHGHGHG
  ip ospf cost 10
```

```
!
interface vlan 11
   ip address 192.168.30.81 255.255.255.0
   ip ospf password ijklmnop
   ip ospf cost 20
   ip ospf retransmit-interval 10
   ip ospf transmit-delay 2
   ip ospf priority 4
!
interface vlan 12
   ip address 192.168.40.81 255.255.255.0
   ip ospf password abcdefgh
   ip ospf cost 10
!
interface vlan 13
   ip address 192.168.0.81 255.255.255.0
   ip ospf password ijklmnop
   ip ospf cost 20
   ip ospf dead-interval 80
!
router ospf 192
   network 192.168.0.0 255.255.255.0 area 0
   network 192.168.20.0 255.255.255.0 area 192.168.20.0
   network 192.168.30.0 255.255.255.0 area 192.168.30.0
   network 192.168.40.0 255.255.255.0 area 192.168.40.0
   area 0 authentication simple
   area 192.168.20.0 stub
   area 192.168.20.0 authentication simple
   area 192.168.20.0 default-cost 20
   area 192.168.20.0 authentication simple
   area 192.168.20.0 range 36.0.0.0 255.0.0.0
   area 192.168.30.0 range 192.42.110.0 255.255.255.0
   area 0 range 130.0.0.0 255.0.0.0
   area 0 range 141.0.0.0 255.0.0.0
   redistribute rip
```

RIP is in network 192.168.30.0.

```
router rip
   network 192.168.30.0
   redistribute ospf 192
!
```

Chapter 4 Configuring BGP

The chapter describes how to configure the Boundary Gateway Protocol (BGP). For details about BGP commands, refer to section "BGP Commands". BGP is an Exterior Gateway Protocol (EGP) defined in RFC1163, 1267 and 1771. BGP allows to create a routing selection mechanism among the autonomous systems. The routing selection mechanism can ensure automatic exchange of routing selection information among the auto-managed system without loop.

4.1 Overview

4.1.1 BGP Introduction

In BGP, each route contains a network number, auto-managed system list that the route passes (as-path) and other attribute lists. Our switch software supports version 4 BGP defined in RFC1771. The basic function of BGP is to exchange network reachable information with other BGP systems, including information about the AS routing table. The information about AS routing table can be used to construct the AS connection figure and apply AS-level routing strategy through the AS connection figure. BGP Version 4 supports CIDR. CIDR reduces the size of the routing table by creating the summary route. The super network, therefore, is generated. CIDR cancels the notion of BGP network class and supports IP prefix broadcast. The CIDR can be transmitted through OSPF, enhanced IGRP, ISIS-IP and RIP2.

EGP is different from IGP with its enhanced control capability. BGP provides multiple optional methods to control the routes.

- Use neighbor-based access-list, aspath-list and prefix-list to filter the route. Or use port-based access-list and prefix-list to filter the route or the **Nexthop** attribute of the route.

- Use route-map to modify BGP route's attributes such as MED, Local Preference and Weight.

- To interact with dynamic IGRPs such as ospf and rip, you can use the **distribute** command to redistribute the route. The BGP routing information is thus automatically generated. The BGP route can be generated by manually configuring **network** and **aggregate**. When the BGP route is generated, you can use **route-map** to set the attribute of the route.

- To control the priority of BGP routes in the system, run the distance command to set the management distance of the BGP route.

4.1.2 BGP Route Selection

The decision procedure of BGP is based on route attribute comparing. When there are multiple routes to reach the same network, BGP selects the optimal route. The procedure of BGP selecting the optimal route is shown as follows:

- If the next hop cannot be reached, the optimal route is considered.

- If the route is an internal one and synchronization is activated, the optimal route is not considered when the route is not in IGP.

- The route with maximum weight is preferentially selected.

- If all routes have the same weight, the route with maximum local priority is preferentially selected.

- If all routes have the same local priority, the route generated by the local router is preferentially selected. For example, routes may be generated when the local router runs the **network** command or the **aggregate** command or the IGP routes are forwarded.

- If the local priority is same, or if the routes are not generated by the local router, the route with the shortest AS path is first selected.

- If the AS paths are same, the route with the smallest Origin attribute value (IGP < EGP < INCOMPLETE) is first selected.

- If the Origin attribute values is the same, the route with the smallest MED value is first selected. The MED value compare is for the routes from the same neighboring AS unless **bgp always-compare-med** is activated.

- If all routes have the same MED, the EBGP is first selected. All paths in the autonomous system are taken as IBGP.

If each route has the same connection attribute, the route with the smallest **router-id** is first selected.

4.2   BGP Configuration Task

4.2.1 Configuring Basic BGP Characteristic

BGP configuration tasks can be classified into two groups: basic tasks and advanced tasks. The first two items of basic tasks are mandatory for BGP configuration. Other items in basic tasks and advanced tasks are optional.

4.2.1.1   Activating BGP Routing Choice

Run the following commands in global configuration mode to activate BGP route selecting:

Command	Purpose
router bgp *autonomous-system*	Activates the BGP routing process in router configuration mode.
network *network-number/masklen [route-map route-map-name]*	Marks the network as the local autonomous system and adds it to the BGP table.

Note:

1)   For EGP, when you use the router configuration command **network** to configure an IP network, you can control which network can get notification. It is contrary for IGP. For example, The RIP protocol uses the **network** command to decide where the update is sent.

2)   You can use the **network** command to add the IGP route to the BGP routing table. The router resources, such as the configured RAM, decide the upper limit of the available **network** command. As an additional choice, you also can run the **redistribute** command.

4.2.1.2   Configuring BGP Neighbor

To exchange routing information with the outside, the BGP neighbor must be configured.

BGP supports two neighbors: IBGP and EBGP. The interior neighbors are in the same AS. The exterior neighbors are in a different AS. In general, exterior neighbors are closely adjacent and share a subnet; interior neighbors are in anyplace of the same AS.

Run the router configuration command to configure the BGP neighbors:

Command	Purpose
neighbor {*ip-address* } **remote-as** *number*	Designates a BGP neighbor.

For details, refer to the section "BGP Neighbor Configuration Example".

4.2.1.3   Configuring BGP Soft Reconfiguration

In general, BGP neighbors exchange all routes only when the connection is created; they then exchange only the changed routes later. If the configured routing policy is changed, you must clear the BGP sessions before you apply the changed routing policy to the received routes. However, clearing the BGP session can disable the high-speed cache and seriously undermine network running. You are recommended to adopt the soft reconfiguration function because it helps to configure and activate policy without clearing BGP sessions. Currently, the new soft reconfiguration function can be applied to each neighbor. The new soft reconfiguration is applied to the incoming update generated by neighbors, it is called incoming soft reconfiguration. When the new soft reconfiguration is used to send the outgoing update to the neighbor, it is called outgoing soft reconfiguration. After the incoming soft reconfiguration is run, new input policies validates. After the outgoing soft reconfiguration is run, the new local output policy validates without resetting BGP session.

In order to generate the incoming update without resetting BGP session, the router of the local BGP session should restore the received incoming update without modification. Whether the incoming update is received or declined by the current incoming policy is not in the consideration. In this case, the memory will be badly occupied. The outgoing reconfiguration has no extra memory cost, so it is always valid. You can trigger the outgoing soft reconfiguration on the other side of the BGP session to validate the new local incoming policy.

To permit the incoming soft reconfiguration, you need to configure BGP to restore all received routing update. The outgoing soft reconfiguration does not require pre-configuration.

Run the following command to configure BGP soft reconfiguration:

Command	Purpose
Neighbor { ip-address } soft-reconfiguration [inbound]	Configures BGP soft reconfiguration.

4.2.1.4   Resetting BGP Connection

Once two routers are defined as BGP neighbors, they will create a BGP connection and exchange route choice information. If the BGP routing policy is modified afterwards, or if other configuration is changed, you must reset the BGP connection to validate the changed configuration. Run one of the following commands to reset the BGP connection.

Command	Purpose
clear ip bgp *	Resets all BGP connections.
clear ip bgp *address*	Resets a special BGP connection.

4.2.1.5   Configuring Synchronization Between BGP and IGPs

If an AS sends information at the third AS through your AS, the internal routing state of your AS must be the same as the routing information that the AS broadcasts to other ASs. For example, before all routers in your AS learn the routes through IGP, your AS may receive routing information from your BGP that some routers cannot route. The synchronization between BGP and IGP is that the BGP does not broadcast the routing information until all IGP routers in the AS learn the routing information. The synchronization is activated by default.

In some cases, you need not to perform the synchronization between BGP and IGP. If other ASs are not allowed to send data through your AS, or if all routers in your AS run BGP, the synchronization will be cancelled. After the synchronization is cancelled, your IGP can carry a few routes and BGP will aggregate more rapidly.

Run the following command to cancel the synchronization:

Command	Purpose
no synchronization	Cancel the synchronization between BGP and IGP.

When cancelling the synchronization, you need to run the command **clear ip bgp** to clear BGP sessions.

For details, refer to the section "Example for Neighbor-Based BGP Path Filtration".

In general, only one or two routes are forwarded to your IGP and become the exterior routes in IGRP or the BGP session sponsor generates a default AS route. When the routes are forwarded from BGP to IGP, only the routes obtained through EBGP can be forwarded. In most cases, your IGP is not redistributed to BGP; the networks in the AS are listed by running the router configuration command **network**; your network, therefore, will be broadcast. The network listed in this way is called as the local network; BGP has the origin attribute of IGP. These routes, such as directly-connected routes, static routes or routes learned from IGP, must be in the main IP routing table and be valid. In BGP routing process, the main IP routing table is scanned periodically

to detect whether local network exists and the BGP routing table is updated afterwards. Be careful when the BGP forwards the routes. Routes in IGP may be forwarded by other routers through BGP. BGP potentially sends information to IGP and IGP then sends the information back to BGP.

4.2.1.6   Configuring BGP Route Weight

BGP route weight is a number that is endowed to BGP route for controlling route choice process. The weight is local for the router. The weight ranges from 0 to 65535. The default weight of the local BGP routes is 32768. The route weight obtained from the neighbor is 0. The administrator can carry out the routing policy by modifying the route weight.

Run the following command to configure the route weight:

Command	Purpose
neighbor {*ip-address* } **weight** *weight*	**Designates a weight for all neighbor's routes.**

You can also modify the route weight through the route map.

4.2.1.7   Configuring Neighbor-Based BGP Routing Filtration

The router software provides the following methods to filter the BGP routes of the designated neighbor:

(1)   Use the Aspath list filter with the commands **ip aspath-list** and **neighbor filter-list**.

Command	Purpose
ip aspath-list *aspaths-list-name* {**permit | deny**} *as-regular-expression*	Defines a BGP-related access table.
router bgp *autonomous-system*	Enters the router configuration mode.
neighbor {*ip-address* } **filter-list** *aspath-list-name* {**in | out** }	Establishes a BGP filter.

(2)   Use the access list with the commands **ip access-list** and **neighbor distribute-list**.

Command	Purpose
ip access-list standard *access-list-name*	Defines an access list.
router bgp *autonomous-system*	Enters the router configuration mode.
neighbor {*ip-address* } **distribute-list** *access-list-name* {**in | out** }	Establishes a BGP filter.

(3)   Use the prefix list with the commands **ip prefix-list** and **neighbor prefix-list**.

Command	Purpose
ip prefix-list *prefixs-list-name* |*sequence number* {**permit |deny**} A.B.C.D/n **ge** x **le** y	Defines a prefix list.

router bgp *autonomous-system*	Enters the router configuration mode.
neighbor {*ip-address* } **prefix-list** *prefix-list-name* {**in	out**}	Establishes a BGP filter.

(4) Use the route mapping with the commands **route-map** and **neighbor route-map**.

Route mapping can filter and change the routing attribute.

For details, refer to the section "Example for Neighbor-Based BGP Path Filtration".

4.2.1.8 Configuring Port-Based BGP Route Filtration

You can use the access list or the prefix list to configure the port-based BGP route filtration. You can filter the network number or the gateway address of the route. You can designate the **access-list** option to use the access list, or designate the **prefix-list** option to use the prefix list to filter the network number of the route. You also can designate the **gateway** option to use the access list to filter the **Nexthop** attribute of the route. The **access-list** option and the **prefix-list** option cannot be used together. The asterisk mark (*) can be designated to filter routes on all ports.

Run the following command in BGP configuration mode to configure the port-based BGP route filtration.

Command	Purpose
filter interface 〔**in	out**〕〔**access-list** access-list-name〕〔**prefix-list** prefix-list-name〕〔**gateway** access-list-name〕	Configures the port-based BGP route filtration.

For details, refer to the section "Example for Port-Based BGP Route Filtration".

4.2.1.9 Cancelling BGP-Updated Next Hop Processing

You can cancel the next hop processing for the neighbor's BGP update. The configuration is useful in the non-broadcast networks such as frame relay or X.25. In frame relay or X.25, BGP neighbors cannot directly access all other neighbors in the same IP subnet. The following methods can cancel the next hop processing:

● The local IP address that uses the BGP connection replaces the next-hop address of the outgoing route.

● Use the route map to designate the next-hop address of the outgoing route or the incoming route.

Run the following command to cancel the next-hop processing:

Command	Purpose
neighbor {*ip-address* } **next-hop-self**	Cancels the next-hop processing when BGP neighbors update.

When the previous command is used, the current router notifies itself to take as the next hop of the route. Therefore, other BGP neighbors will send packets to the current

router. It is useful in the non-broadcast network because a path from the current router to the designated neighbor. However, it is useless in the broadcast network because unnecessary extra hops will occur.

4.2.2 Configuring Senior BGP Characteristics

4.2.2.1 Filtering and Modifying Route Update Through Route Map

The route map can be used on each neighbor to filter the route update and modify the parameter's attributes. The route map can be applied in both the incoming update and the outgoing update. Only the routes that pass the route map are processed when the route update is sent or received.

The route map supports that the incoming update and the outgoing update are based on the AS path, community and network number. The **aspath-list** command requires be used for the AS matching. The community matching requires the **community-list command. The network matching requires the ip access-list command** .

Run the following command to filter and modify the route update through the route map.

Command	Purpose
neighbor {*ip-address* } **route-map** *route-map-name* **{in | out}**	Applies the route map to the incoming or outgoing route.

For details, refer to the section "BGP Route Map Example".

4.2.2.2 Configuring Aggregation Addres

The non-type inter-field route can create the aggregation route (and super network) to minimize the routing table. You can configure the aggregation route by redistributing the aggregation route to BGP or by using the aggregation attribute described in the following table. If the BGP table has at least one more detailed record, add the aggregation address to the BGP table.

Use one or several of the following command to create the aggregation address in the routing table:

Command	Purpose
aggregate network/len	Creates the aggregation address in the routing table.
aggregate network/len **summary-only**	Broadcasts only the summary address.
aggregate network/len **route-map** map-name	Generates the designated aggregation address through the route map.

Refer to the section "BGP Route Aggregation Example".

4.2.2.3 Configuring BGP Community Attribute

The routing policy that BGP supports is based one of the following three values for BGP routing information:

- Routing network number

- Value of the **AS_PATH** attribute

- Value of the **COMMUNITY** attribute

Routes can be classified into the community through the COMMUNITY attribute and the community-based routing policy can be applied to routes. Therefore, the configuration of routing information control is simplified.

Community is a group of routes having the same attributes. Each route may belong to multiple communities. The AS administrator can decide which community a route belongs to.

The COMMUNITY attribute is an optional, transmissible and global, which ranges from 1 to 4,294,967,200. The famous communities that are predefined in the Internet are listed in the following table:

Community	Description
no-export	Does not broadcast the route to the EBGP peers, including the EBGP peers in the autonomous system.
no-advertise	Does not broadcast the route to any peer.
local-as	Does not broadcast the route to the outside of the autonomous system.

When generating, receiving or forwarding the route, the BGP session sponsor can set, add or modify the route community attributes. After the routes are aggregated, the aggregation contains the COMMUNITY attribute from all original routes.

The COMMUNITY attribute is not sent to neighbors by default. Run the following command to send the COMMUNITY attribute to the designated neighbor.

Command	Purpose
neighbor {*ip-address* } **send-community**	Sends the COMMUNITY attribute to the designated neighbor.

Perform the following operations to set the community attribute:

Command	Purpose
route-map *map-name sequence-number* **{deny | permit}**	Configures the route map.
set community *community-value*	Configures the setup regulations.
router bgp *autonomous-system*	Enters the router configuration mode.
neighbor {*ip-address* } **route-map** *access-list-name* {**in | out** }	Applies the route map.

Perform the following operations to configure the community-attribute–based routing information filtration:

Command	Purpose
Command	Purpose

ip community-list standard | expended *community-list-name* {**permit | deny**} *communtiy-expression*	Defines the community list.
route-map *map-name* *sequence-number* **{deny | permit}**	Configures the route map.
match *community-list-name*	Configures the matching regulations.
router bgp *autonomous-system*	Enters the router configuration mode.
neighbor {*ip-address* } **route-map** *route-map-name* {**in | out** }	Applies the route map.

Refer to the section "Example for Route Map Through BGP Community Attribute".

4.2.2.4   Configuring Autonomous System Alliance

The method to reduce IBGP connections is to divide one AS into multiple sub ASs and classify them into an autonomous system alliance. As to the outside, the alliance seems like an AS. As to the inside of the alliance, each sub AS is full-connected and connects other sub ASs in the same alliance. Even if the EBGP session exists in the peers of different sub AS, they still exchange route choice information as IBGP peers do. That is, they save the next hop, MED and local priority information.

To configure a BGP autonomous system alliance, you must designate the alliance identifier. The alliance identifier is an AS number. As to the outside, the AS looks like a single AS which takes the alliance identifier as the AS number.

Run the following command to configure the identifier of the autonomous system alliance:

Command	Purpose
bgp confederation0 identifier *autonomous-system*	Configures the identifier of the autonomous system alliance.

Run the following command to designate the autonomous system number belonging to the autonomous system alliance:

Command	Purpose
bgp confederation peers autonomous-system [autonomous-system ...]	Designates the AS belonging to the autonomous system alliance.

Refer to the section "BGP Autonomous System Alliance Example".

4.2.2.5   Configuring Route Reflector

Another method to reduce IBGP connections is to configure the route reflector.

The peers in the route reflector are divided into two groups: client peers and other routers in the AS (non-client peers). The route reflector reflects the routes between the two groups. The route reflector and the client peers consists of a cluster. The non-client peers must be fully connected. The client peers need not be fully connected. The clients in the cluster do not communicate with the IBGP session sponsors in the different cluster.

When the route reflector receives the routing infotmation, it will perform the following tasks:

● Broadcast the routes from the external BGP session sponsors to all clients and non-client peers.

● Broadcast the routes from the non-client routes to all clients.

● Broadcast the routes from the client to all client peers and non-client peers. The client peers need not be fully connected.

Run the following command to set the local router as the reflector and designate the neighbor as the client:

Command	Purpose
neighbor *ip-address* **route-reflector-client**	Sets the local router to the reflector and designate the neighbor as the client.

One AS has multiple route reflectors. The route reflector handles other route reflectors as it handles IBGP session sponsors. In general, the clients in the same cluster has only one route reflector. The cluster is identified by the router ID of the route reflector. To add redundancy and avoid the failure of the single node, one cluster may have several route reflectors. In this case, all route reflectors in the cluster must be set to a 4-bit cluster ID, enabling the route reflector to identify the update information of other route reflectors in the same cluster. All the route reflectors in the same cluster must be fully connected and have the same client peers and non-client peers.

If several route reflectors exists in a cluster, run the following command to configure the cluster ID:

Command	Purpose
bgp cluster-id *cluster-id*	Configures the cluster ID.

Refer to the section "BGP Route Reflector Configuration Example".

4.2.2.6 Shutting down peers

Run the following command to shut down the BGP neighbors:

Command	Purpose
neighbor {ip-address } **shutdown**	Shuts down the BGP neighbor.

Run the following command to activate the neighbor:

Command	Purpose
no neighbor {ip-address } **shutdown**	Activates the BGP neighbor.

4.2.2.7 Configuring multihop external peers

The external peers must be in the directly-connected networks by default. Run the following command to configure multihop external peers:

Command	Purpose

neighbor {ip-address } **ebgp-multihop** ttl	Sets the BGP neighbor to the multihop external peers.

4.2.2.8   Setting BGP route management distance

The management distance is a unit to measure the priority of routing protocols. BGP uses three kinds of management distance: external distance, internal distance and local distance. The route learned from the external BGP shows the external distance. The route learned from the internal BGP shows the internal distance. The local route shows the local distance. Run the following command to set BGP route management distance:

Command	Purpose
distance bgp external-distance internal-distance local-distance	Sets BGP route management distance.

It is dangerous to modify the management distance of the BGP routes. You are not recommended to do it. The external distance should be shorter than the distance of any dynamic routing protocol. The internal distance should be longer than the distance of any dynamic routing protocol.

4.2.2.9   Modifying BGP timer

Run the following command to modify BGP **keepalive** and **holdtime** timer:

Command	Purpose
neighbor *[ip-address	peer group-name]* **timers** *keepalive holdtime*	Sets the **keepalive** and **holdtime** timer for the designed peers or the peer group (unit: second).

Run the command **no neighbor timers** to resumes the timer of the BGP neighbor or the peer group to the default value.

4.2.2.10   Comparing MED of the routes from different ASs

MED is a parameter that is considered when an optimal route needs to be selected from multiple available paths. The path with comparatively small MED value is first considered.

By default, when the best route is being chosen, the MED compare is performed only among the routes from the same AS. You can configure to allow the MED compare during route choice, no matter which AS the routes come from.

Run the following command to perform the MED compare among routes from different ASs:

Command	Purpose
bgp always-compare-med	Performs the MED compare among routes from different ASs.

4.3 Monitoring and Maintaining BGP

The administrator can browse and delete the content in the routing table or other databases in BGP. The value of the detailed statistics information can be displayed.

4.3.1.1 Clearing BGP routing table and database

Run the following command in management mode to perform relative tasks about clearing high-speed cache, table or BGP database.

Command	Purpose
clear ip bgp *	Resets all BGP connections.
clear ip bgp *as-number*	Resets the BGP connection of the designated autonomous system.
clear ip bgp *address*	Resets the BGP connection of the designated neighbor.
clear ip bgp *address* **soft** ｛**in|out**｝	Clears the incoming or outgoing database of the designated neighbor.
clear ip bgp *aggregates*	Clears the routes generated during route aggregation.
clear ip bgp *networks*	Clears the routes generated by the **network** command.
clear ip bgp *redistribute*	Clears the routes generated in the forwarding process.

4.3.1.2 Displaying routing table and system statistics information

The detailed statistics information such as the BGP routing table and the database content can be displayed. These statistics information helps you to fully use network resources and resolve network problems.

Run the following command to display different kinds of statistics information:

Command	Purpose
show ip bgp	Displays the BGP routing table in the system.
show ip bgp prefix	Displays the routs that match the prefix-matched list.
show ip bgp community	Displays the statistics information about the community attribute.
show ip bgp regexp *regular-expression*	Displays the routes that match the regular expression.
show ip bgp *network*	Displays the designated BGP route.
show ip bgp neighbors *address*	Displays the detailed information about the TCP connection and BGP connection of the designated neighbor.

show ip bgp neighbors *[address]* [received-routes | routes | advertised-routes]	Displays the routes learned from a special BGP neighbor.
show ip bgp paths	Displays all BGP path information in the database.
show ip bgp summary	Displays the state of all BGP connections.

4.3.1.3   Tracking BGP information

To locate the fault and resolve the problem, you need to observe the BGP connection establishment, route receiving and route forwarding by tracking the BGP information. Perform the following operations:

Command	Purpose
debug ip bgp *	Tracks common BGP information.
debug ip bgp all	Tracks all BGP information.
debug ip bgp fsm	Tracks the BGP state machine.
debug ip bgp keepalive	Tracks the BGP **keepalive** message .
debug ip bgp open	Tracks the BGP **Open** message.
debug ip bgp update	Tracks the BGP **Update** message.

4.4   BGP Configuration Example

4.4.1.1   BGP route map example

The following example shows how to modify the attributes of the incoming route from neighbors by using the route map. Set the weight of any route that is received from neighbor 140.222.1.1 and matches the ASPATH access list **aaa** to **200**. Set the local priority to **250**. If the route is declined, other routes are declined.

router bgp 100

!

 neighbor 140.222.1.1 route-map fix-weight in

 neighbor 140.222.1.1 remote-as 1

!

route-map fix-weight permit 10

 match as-path aaa

 set local-preference 250

 set weight 200

!

ip aspath-list aaa permit ^690$

ip aspath-list aaa permit ^1800

In the following example, the first item of route map **freddy** sets the MED attribute of all routes starting from autonomous system 690 to **127**. The second item enables the routes that do not satisfy the previous conditions to be sent to neighbor 1.1.1.1:

router bgp 100

```
 neighbor 1.1.1.1 route-map freddy out
!
ip aspath-list abc permit ^690_
ip aspath-list xyz permit .*
!
route-map freddy permit 10
 match as-path abc
 set metric 127
!
route-map freddy permit 20
match as-path xyz
```

The following example shows how to modify the routes that are generated in route forwarding through the route map:

```
router bgp 100
 redistribute rip route-map rip2bgp
!
 route-map rip2bgp
 match ip address rip
 set local-preference 25
 set metric 127
 set weight 30000
 set next-hop 192.92.68.24
 set origin igp
!
ip access-list standard rip
 permit 131.108.0.0 255.255.0.0
 permit 160.89.0.0 255.255.0.0
 permit 198.112.0.0 255.255.128.0
```

4.4.1.2   BGP neighbor configuration example

In the following example, the BGP router belongs to AS109. AS109 establishes two networks. The router has three neighbors: an external neighbor (in a different AS), an internal neighbor (with the same AS number) and an external neighbor.

```
router bgp 109
 network 131.108.0.0
 network 192.31.7.0
 neighbor 131.108.200.1 remote-as 167
 neighbor 131.108.234.2 remote-as 109
 neighbor 150.136.64.19 remote-as 99
```

4.4.1.3   Example for neighbor-based BGP path filtration

The following is an example for neighbor-based BGP path filtration. The route that gets through the access list **test1** of **as-path** obtains a weight value **100**. Only the route that gets through the access list **test2** of **as-path** can be sent to neighbor 193.1.12.10.

Similarly, the route that gets through the access list **test3** can be accepted by neighbor 193.1.12.10:

```
router bgp 200
 neighbor 193.1.12.10 remote-as 100
 neighbor 193.1.12.10 filter-list test1 weight 100
 neighbor 193.1.12.10 filter-list test2 out
 neighbor 193.1.12.10 filter-list test3 in
ip aspath-list test1 permit _109_
ip aspath-list test2 permit _200$
ip aspath-list test2 permit ^100$
ip aspath-list test3 deny _690$
ip aspath-list test3 permit .*
```

4.4.1.4   Example for port-based BGP route filtration

The following example shows that the routes from port e1/0 are filtered through access list **acl**:

```
router bgp 122
 filter vlan10 in access-list acl
```

The following example shows how to filter the routes from port e1/0 simultaneously using the access list **filter-network** and the access list **filter-gateway** to respectively filter the network number and the gateway address.

```
router bgp 100
 filter vlan100 in access-list filter-network gateway filter-gateway
```

The following example shows how to filter routes from all ports simultaneously using the prefix list **filter-prefix** and the prefix list **filter-gateway** to respectively filter the network number and the gateway address.

```
router bgp 100
filter * in prefix-list filter-prefix gateway filter-gateway
```

4.4.1.5   Example for prefix-list-based route filtration configuration

The following example shows that the default route 0.0.0.0/0 is declined:

```
ip prefix-list abc deny 0.0.0.0/0
```

The following example shows that the route which matches the prefix 35.0.0.0/8 is allowed:

```
ip prefix-list abc permit 35.0.0.0/8
```

In the following example, only the prefixes with the length from /8 to /24 are accepted in the BGP process:

```
router bgp
 network 101.20.20.0
 filter * in prefix max24
 !
 ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
 !
```

In the following example, the router filters all the routes and only accepts the routes whose prefix length ranges from 8 to 24:

router bgp 12

 filter * in prefix-list max24

!

ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24

The following example shows that route whose prefix length is no more than 24 is permitted in network 192/8:

ip prefix-list abc permit 192.0.0.0/8 le 24

The following example shows that route whose prefix length exceeds 25 is permitted in network 192/8:

ip prefix-list abc deny 192.0.0.0/8 ge 25

The following example shows that routes whose prefix length is larger than 8 and smaller than 24 are permitted:

ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24

The following example shows that routes whose prefix length exceeds 25 are denied:

ip prefix-list abc deny 0.0.0.0/0 ge 25

The following example shows that all routes from network 10/8 are denied. If the mask of A-class network 10.0.0.0/8 is less than or equal to 32 bits, all routes are denied:

ip prefix-list abc deny 10.0.0.0/8 le 32

The following example shows that all routes are denied because the mask length of network 204.70.1/24 exceeds 25:

ip prefix-list abc deny 204.70.1.0/24 ge 25

The following example shows that all routes are permitted:

ip prefix-list abc permit any

4.4.1.6   BGP route aggregation example

The following example shows how to create the aggregation route in BGP through route forwarding or the conditional route aggregation function:

In the following example, the command **redistribute static** is used to forward the aggregation route 193. * . * . * :

ip route 193.0.0.0 255.0.0.0 null 0

!

router bgp 100

 redistribute static

If at least one route in the routing table belongs to the designated range, an aggregation route is created in the BGP routing table according to the following configuration. The aggregation route is considered to be from your AS and has the **atomic** attribute which may be lost in the indication information:
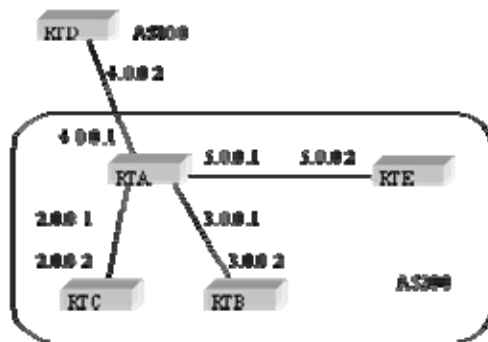
router bgp 100

aggregate 193.0.0.0/8

The following example shows how to create the aggregation route 193.*.*.* and how to constrain more detailed routes from broadcasting to all neighbors:

router bgp 100
aggregate 193.0.0.0/8 summary-only

4.4.1.7   BGP route reflector configuration example

The following is an example for the route reflector configuration. RTA, RTB, RTC and RTE belongs to the same autonomous system AS200. RTA functions as the route reflector, while RTB and RTC function as the clients of the route reflector. RTE is a common IBGP neighbor. RTD belongs to AS100 and establishes an EBGP connection with RTA. The configuration is shown as follows:

RTA configuration:

```
interface vlan110
ip address 2.0.0.1 255.0.0.0
!
interface vlan111
ip address 3.0.0.1 255.0.0.0
!
interface vlan112
ip address 4.0.0.1 255.0.0.0
!
interface vlan113
ip address 5.0.0.1 255.0.0.0
!
router bgp 200
neighbor 2.0.0.1 remote-as 200    /*RTC IBGP*/
neighbor 2.0.0.1 route-reflector-client
neighbor 3.0.0.1 remote-as 200    /*RTB IBGP*/
neighbor 3.0.0.1 route-reflector-client
neighbor 5.0.0.1 remote-as 200    /*RTE IBGP*/
neighbor 4.0.0.2 remote-as 100    /*RTD EBGP*/
network 11.0.0.0/8
!
```

```
ip route 11.0.0.0 255.0.0.0 2.0.0.12
```

RTB configuration:

```
interface vlan110
ip address 3.0.0.2 255.0.0.0
!
router bgp 200
neighbor 3.0.0.1 remote-as 200    /*RTA IBGP*/
network 13.0.0.0/8
!
ip route 13.0.0.0 255.0.0.0 3.0.0.12
```

RTC configuration:

```
interface vlan110
ip address 2.0.0.2 255.0.0.0
!
router bgp 200
neighbor 2.0.0.1 remote-as 200    /*RTA IBGP*/
network 12.0.0.0/8
!
ip route 12.0.0.0 255.0.0.0 2.0.0.12
```
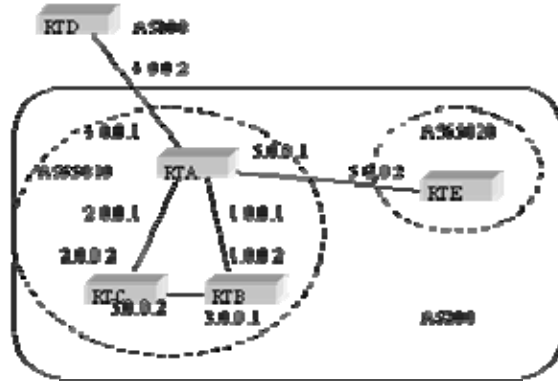
RTD configuration:

```
interface vlan110
ip address 4.0.0.2 255.0.0.0
!
router bgp 100
neighbor 4.0.0.1 remote-as 200    /*RTA EBGP*/
network 14.0.0.0/8
!
ip route 14.0.0.0 255.0.0.0 4.0.0.12
```

RTE configuration:

```
interface vlan110
ip address 5.0.0.2 255.0.0.0
!
router bgp 200
neighbor 5.0.0.1 remote-as 200    /*RTA IBGP*/
network 15.0.0.0/8
!
ip route 15.0.0.0 255.0.0.0 5.0.0.12
```

4.4.1.8 BGP autonomous system alliance example

The following figure shows an autonomous system alliance configuration. RTA, RTB and RTC create the IBGP connection. RTA, RTB and RTC belong to the private autonomous system 65010. RTE belongs to the private autonomous system 65020. RTE and RTA establish the EBGP connection in the autonomous system alliance. AS65010 and AS65020 make up of an autonomous system alliance. The number of the autonomous system alliance is AS200. RTD belongs to AS100. An EBGP connection is established between RTD and AS200 through RTA.

RTA configuration:

```
interface vlan110
ip address 1.0.0.1 255.0.0.0
!
interface vlan111
ip address 2.0.0.1 255.0.0.0
!
interface vlan112
ip address 4.0.0.1 255.0.0.0
!
interface vlan113
ip address 5.0.0.1 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
neighbor 1.0.0.2 remote-as 65010   /*RTB IBGP*/
neighbor 2.0.0.2 remote-as 65010   /*RTC IBGP*/
neighbor 5.0.0.2 remote-as 65020   /*RTE EBGP*/
neighbor 4.0.0.2 remote-as 100     /*RTD EBGP*/
```

RTB configuration:

```
interface vlan110
```

```
ip address 1.0.0.2 255.0.0.0
!
interface vlan111
ip address 3.0.0.1 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
neighbor 1.0.0.1 remote-as 65010   /*RTA IBGP*/
neighbor 3.0.0.2 remote-as 65010   /*RTC IBGP*/
```

RTC configuration:

```
interface vlan110
ip address 2.0.0.2 255.0.0.0
!
interface vlan111
ip address 3.0.0.2 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
neighbor 2.0.0.1 remote-as 65010   /*RTA IBGP*/
neighbor 3.0.0.1 remote-as 65010   /*RTB IBGP*/
```

RTD configuration:

```
interface vlan110
ip address 4.0.0.2 255.0.0.0
!
router bgp 100
neighbor 4.0.0.1 remote-as 200   /*RTA EBGP*/
```

RTE configuration:

```
interface vlan110
ip address 5.0.0.2 255.0.0.0
!
router bgp 65020
bgp confederation identifier 200
bgp confederation peers 65010
neighbor 5.0.0.1 remote-as 65010   /*RTA EBGP*/
```

4.4.1.9   Example for route map using BGP community attribute

In the following example, the command **route map set-community** is used to update the outgoing routes of neighbor 171.69.232.50. The special community attribute value **no-export** can be set through the route of the access list **aaa**. Other routes perform normal broadcast. The special community attribute value automatically prevents the

BGP session sponsor in AS200 from broadcasting the route to the outside of the autonomous system.

router bgp 100

 neighbor 171.69.232.50 remote-as 200

 neighbor 171.69.232.50 send-community

 neighbor 171.69.232.50 route-map set-community out

!

 route-map set-community 10 permit

 match ip address aaa

 set community no-export

!

 route-map set-community 20 permit

In the following example, the command **route map set-community** is used to update the outgoing routes of neighbor 171.69.232. 90. Set the current value to the community attribute value **200**. Other routes performs normal broadcast.

route-map bgp 200

 neighbor 171.69.232.90 remote-as 100

 neighbor 171.69.232.90 send-community

 neighbor 171.69.232.90 route-map set-community out

!

route-map set-community 10 permit

 match as-path test1

 set community-additive 200 200

!

route-map set-community 20 permit

 match as-path test2

!

ip aspath-list test1 permit 70$

ip aspath-list test2 permit .*

In the following example, Set the MED and the local priority of the route from neighbor 171.69.232.55 according to the community attribute value.   Set MED of all routes that match the community list **com1** to **8000**. These routes may contain routes with community value "100 200 300" and "900 901". These routes may have other attribute values.

Set the local priority of the routes which send the community list **com2** to **500**.

Set the local priority of other routes to **50**. Therefore, all the local priority value of all remaining routes of neighbor 171.69.232.55 is **50**.

router bgp 200

neighbor 171.69.232.55 remote-as 100

neighbor 171.69.232.55 route-map filter-on-community in

!

route-map filter-on-community 10 permit

match community com1

set metric 8000

!

```
route-map filter-on-community 20 permit
match community com2
set local-preference 500
!
route-map filter-on-community 30 permit
set local-preference 50
!
ip community-list com1 permit 100 200 300
ip community-list com1 permit 900 901
!
ip community-list com2 permit 88
ip community-list com2 permit 90
!
```