QTECH
МИР ДОСТУПНЕЕ

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Руководство пользователя

QSR-2830

www.qtech.ru

Оглавление

1 CONFIGURING PROTOCOL-INDEPENDENT

1.1 Configuring the IP Routing

1.1.1 Configuring Static Routes

Static routes are manually configured to send the packets to the specified target network. It is essential to configure the static routes when the routes of some target network cannot be learned by the dynamic routing protocols. Usually, a default static route is configured for the packets without the routes.

To configure static routes, execute the following commands in global configuration mode:

Command	Function
Qtech(config)# **ip route** [**vrf** *vrf_name1*] *network mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [**tag** *tag*] [**permanent** | **track** *object-number*] [**weight** *weight*]	Configures the static routes. The vrf vrf-name1 parameter can be used to specify the VRF for the routes.
Qtech(config)# **no ip route** [**vrf** *vrf_name1*] *network mask*	Deletes the static routes.
Qtech(config)# **ip static route-limit** *number*	Specifies the upper limit of the static routes.
Qtech(config)# **no ip static route-limit**	Restores the static routes to the default maximum values.

For examples of configuring the static routes, see the "Example of Replacing the Static Routes with the Dynamic Routes" section.

If the static routes are not deleted, they will be retained on Qtech products permanently. However, you can replace the static routes with the better routes learned by the dynamic routing protocols. Better routes mean that they have smaller management distances. All routes, including the static ones, carry the parameters of the management distance. The following table shows the management distances for various route sources retained on Qtech products:

Route source	Default management distance
Directly connected network	0
Static route	1
OSPF route	110
ISIS route	115
RIP route	120
Unreachable route	255

Note

The static route redistribution shall be configured if the static routes require to be advertised by the dynamic routing protocols such as RIP and OSPF.

When an interface is "down", all routes to the interface disappear from the routing table. In addition, when Qtech products fail to find the forwarding route to the next-hop address for a static route, the static route will also disappear from the routing table.

When the specified VRF static routes are added to the corresponding VRF, if the egress is specified at the same time, the addition fails when the VRF of the egress does not match the specified VRF. If no VRF is specified, it is added to the global routing table by default.

If the specified VRF is a multi-protocol VRF, the static route can be configured only for the multi-protocol VRF that is configured for the IPv4 address family. When the IPv4 address family of the VRF is deleted, the IPv4 static route of the VRF will also be deleted.

If the association of the static route with the track object is specified, and if the track object is advertised to be inactive, the static route also takes no effect.

By default, the weight of the static route is 1. To view the static route of non-default weights, execute the command **show ip route weight**. The weight parameter is used to enable the WCMP function. When there are load-balanced routes to a destination address, the switch assigns data flows by their weights. The higher the weight of a route is, the more data packets the route carries. The WCMP limit is generally 32 for routers. However, the WCMP limit varies depending on switch models because their chipsets support different weights. For the detailed information about the route weight value of specific models, see the product specifications. When the sum of the load-balanced route weights exceeds the WCMP limit, the excessive routes will not take effect. For example, if the WCMP limit on a device is 8, only one of the following static route configurations takes effect:

```
Qtech(config)#ip route 10.0.0.0 255.0.0.0 172.0.1.2 weight 6
Qtech(config)#ip route 10.0.0.0 255.0.0.0 172.0.1.4 weight 6
Qtech(config)#show ip route 10.0.0.0

Routing entry for 10.0.0.0/8
  Distance 1, metric 0
  Routing Descriptor Blocks:
  *172.0.1.2, generated by "static"
Qtech(config)#show ip route weight

------------[distance/metric/weight]-----------
S    10.0.0.0/8 [1/0/6] via 172.0.1.2
```

The maximum number of the static routes is 1024 by default. If the number of the configured static routes exceeds the specified upper limit, they are not be automatically deleted, but the addition fails.

To view the configurations of the IP routing, execute the **show ip route** command to view the IP routing table. For details, see Protocol-independent Command Configuration.

1.1.2 Configuring the Default Route

Not all devices have a complete network-wide routing table. To allow every device to route and forward all packets, it is a common practice that the powerful core device on the network is provided with a complete routing table, while the other devices have a default route to this core device. Default routes can be transmitted by the dynamic routing protocols, and can also be manually configured on every router.

Default routes can be generated in two ways: 1) manually configuring a default static route. For details, see the "Configuring Static Routes" section; 2) manually configuring a default network.

Most internal gateway routing protocols have a mechanism that transmits the default route to the entire routing domain. The device that transmits the default route must have a default route. The transmission of the default route described in this section applies only to the RIP routing protocol. The RIP always notifies the 0.0.0.0/0 network as the default route to the RIP routing domain. For details about how the OSPF routing protocols generate and transmit the default route, see related sections in Guide on Configuring the OSPF Routing Protocols.

For generating the default static route, execute the following commands in global configuration mode:

Command	Function
Qtech(config)# **ip default-network** network	Configures the default network.
Qtech(config)# **no ip default-network** network	Deletes the default network.

Note

Generating the default route by using the **default-network** command must meet the following requirement: The default network is not a directly-connected interface network, but is reachable in the routing table.

Note

Under the same condition, the RIP can also transmit the default route. Alternatively, the RIP can use another way to transmit the default route, that is, by configuring the default static route or learning the 0.0.0.0/0 route by other routing protocols.

If the router has a default route, whether it is learned by the dynamic routing protocol or manually configured, when you execute the show ip route command, the "gateway of last resort" area in the routing table will show information about the last gateway. A routing table may have multiple routes as alternative default routes, but only the best default route is presented in the "gateway of last resort" area.

1.1.3 Configuring the Number of Equivalent Routes

To enable the load-balancing function, configure the number of the equivalent routes for control. An equivalent route is an alternative path to the same destination address. When there is only one equivalent route, one destination address can be configured with only one route, then the load-balancing function is disabled.

To configure the number of the equivalent routes, execute the following command in global configuration mode. Use the no form of this command to restore the default number of the equivalent routes.

This command is valid for both the IPv4 and the IPv6. That is to say, after configuring this command, the maximum numbers of the equivalent paths to the IPv4 and IPv6 destinations are the same as the configured value.

Command	Function
Qtech(config)# **maximum-paths** *number*	Configures the number of the equivalent routes. The maximum number of the equivalent routes configured on different products varies. The maximum number of the equivalent routes configured on routers is 32. However, this number varies depending on the chipsets for switches. Therefore, see the prompts during the configuration.

1.1.4 Configuring the Route-Map

The route-map is a collection of filter policies that are independent from the detailed routing protocols and used in the routing protocols and the policy-based routing. The route-map is used to filter and modify the routing information in the routing protocols, and control the packet forwarding in the policy-based routing.

To define the route-map, execute the following commands in global configuration mode:

Command	Function
Qtech(config)# **route-map** *route-map-name* [[**permit** | **deny**] *sequence*]	Defines the route-map.
Qtech(config)# **no route-map** *route-map-name* [{**permit** | **deny**} *sequence*]	Deletes the route-map.

When configuring the rules for a route-map, you can execute one or more **match** or **set** commands. If there is no **match** command, all routes are matched. If there is no **set** command, no operation is performed.

To define the matching conditions for the rules, execute the following commands in route-map configuration mode:

Command	Function
Qtech(config-route-map)# **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name*} [**exac-match**]…	Matches the community attribute of the BGP route.
Qtech(config-route-map)# **match interface** [*interface-type interface-number*…]	Matches the next-hop interface of the route.
Qtech(config-route-map)# **match ip address** *access-list-number* [*access-list-number*…]	Matches the IP address in the ACL.
Qtech(config-route-map)# **match ip next-hop** *access-list-number* [*access-list-number*…]	Matches the next-hop IP address in the ACL.

Qtech(config-route-map)# **match ip route-source** *access-list-number* [*access-list-number*…]	Matches the route source IP address in the ACL.
Qtech(config-route-map)# **match ipv6 address** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the IPv6 ACL or prefix list.
Qtech(config-route-map)# **match ipv6 next-hop** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the next-hop IP address in the ACL or prefix list.
Qtech(config-route-map)# **match ipv6 route-source** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the route source IP address in the ACL or prefix list.
Qtech(config-route-map)# **match metric** *Metric*	Matches the route metric value. The metric value is in the range from 0 to 4294967295.
Qtech(config-route-map)# **match origin** {**egp** | **igp** | **incomplete**}	Matches the route origin type.
Qtech(config-route-map)# **match route-type** {**local** | **internal** | {{**external** | **nssa-external**} [**type-1**][**type-2**]} | [**level-1** | **level-2**]}	Matches the route type.
Qtech(config-route-map)# **match tag tag**	Matches the route tag value. The tag value is in the range from 0 to 4294967295.

To define operations after matching, execute the following commands in route-map configuration mode:

Command	Function
Qtech(config-route-map)# **set aggregator as** *as-num ip_addr*	Sets the AS attribute value for the route aggregator.
Qtech(config-route-map)# **set as-path prepend** *as-number*	Sets the AS_PATH attribute value.
Qtech(config-route-map)# **set comm-list** *community-list-number* | *community-list-name delete*	Deletes all COMMUNITY attribute values in the COMMUNITY_LIST.
Qtech(config-route-map)# **set community** {*community-number*[*community-number*…] **additive** | **none**}	Sets the COMMUNITY attribute value.
Qtech(config-route-map)# **set dampening** *half-life reuse suppress max-suppress-time*	Sets the route dampening parameters.
Qtech(config-route-map)#**set extcommunity** {**rt** *extend-community-value* | **soo** *extend-community-value*}	Sets the extended community attribute value.
Qtech(config-route-map)# **set interface** *interface-type interface-number*	Sets the interface for forwarding packets.
Qtech(config-route-map)# **set ip default next-hop** *ip-address* [*weight*] [*ip-address* [*weight*] …]	Sets the default next-hop IP address.
Qtech(config-route-map)# **set ipv6 default next-hop** *global-ipv6-address* [*weight*] [*global-ipv6-address* [*weight*]...]	Sets the default next-hop IPv6 address.
Qtech(config-route-map)# **set ip next-hop** *ip-address* [*weight*] [*ip-address* [*weight*]…]	Sets the next-hop IP address.
Qtech(config-route-map)# **set ipv6** [**vrf** *vrf-name* | **global**] **next-hop** *global-ipv6-address* [*weight*] [*global-ipv6-address* [*weight*]]	Sets the next-hop IPv6 address. If the **vrf** *vrf-name* parameter is specified, the VRF is crossed when packets are forwarded. If the **global** parameter is specified, packets are forwarded globally from the VRF. If the [**vrf** *vrf-name* | **global**] parameter is not specified, the IPv6 packets will inherit the VRF during transmission. That is, the next hop belongs to the VRF that receives the IPv6 packets.
Qtech(config-route-map)# **set level** {**stub-area** | **backbone** | **level-1** | **level-1-2** | **level-2**}	Sets the routing area.
Qtech(config-route-map)# **set local-preference** *number*	Sets the LOCAL_PREFERENCE value.
Qtech(config-route-map)# **set metric** *metric*	Sets the metric value for the redistributed route.
Qtech(config-route-map)# **set metric** [+ *metric-value* |- *metric-value* | *metric-value*]	Sets the metric type for the redistributed route.

Qtech(config-route-map)# **set metric-type** {**type-1** | **type-2** | **external** | **internal**}	Sets the metric type for the redistributed route.
Qtech(config-route-map)# **set next-hop** *next-hop*	Sets the next-hop IP address for the redistributed route. next-hop: indicates the next-hop IP address.
Qtech(config-route-map)# **set origin** {**egp** | **igp** | **incomplete**}	Sets the route origin attribute.
Qtech(config-route-map)#**set originator-id** *ip-addr*	Sets the route originator ID.
Qtech(config-route-map)# **set tag** *tag*	Sets the tag value for the redistributed route.
Qtech(config-route-map)# **set weight** *number*	Sets the BGP route weight.

Whether a route-map supports the **match** command and the **set** command depends on applications associated with the route-map. The general instructions are as follows:

- When you configure commands associated with a route-map, the system displays a prompt when the configured match command or the set command is inapplicable to the current applications associated with the route-map.
- When you configure a route-map, the match command, or the set command, the system displays a prompt when any match command or set command is inapplicable to any application associated with the route-map.

For examples of displaying the prompt when the command is not applicable, see the "*Example of Route-Map Configuration*" section.

Caution The two instructions are inapplicable to the association of the policy-based routing with the route-map.

1.2 Redistributing Routes

1.2.1 Configuring Route Redistribution

To enable the device to run multiple routing protocol processes, Qtech products provide the function for redistributing the route information from one routing process to another routing process. For example, you can redistribute the routes in the OSPF routing area to the RIP routing area, and vice versa. All IP routing protocols support the mutual route redistribution.

For route redistribution, route-maps are usually used to control the mutual route redistribution between two routing areas

To redistribute routes from one routing area to another and control the route redistribution, execute the following commands in routing process configuration mode:

Command	Function
Qtech(config-router)# **redistribute** *protocol* [*process-id*] [metric *metric*] [**metric-type** *metric-type*] [**match internal** | **external** *type*| **nssa-external** *type*] [[**tag** *tag*] [**route-map** *route-map-name*] [**subnets**]	Redistributes the routes. Protocol (protocol type): bgp, connected, isis, rip, static
Qtech(config-router)# **default-metric** *metric*	Sets default metric values for all redistributed routes.

The route redistribution may easily cause loops, so be careful when performing the operation.

Note When the route redistribution is configured in the OSPF routing process, the metric value of 20 is allocated to the redistributed routes with the type of Type-2 by default. This type of the routes is the least credible routes to the OSPF.

1.2.2 Configuring Default Route Distribution

To advertise the default route, it is necessary for the routing protocol to introduce the default route to the process, or enforce to generate a default route.

To configure the default route distribution, execute the following commands in routing process configuration mode:

Command	Function
Qtech(config-router)# **default-information originate** [**always**] [**metric** *metric*] [**metric-type** *type*] [**route-map** *map-name*]	Introduces the default route to the routing protocol process and advertises the default route. **always** (optional): always introduces a default route to the process no matter whether the default route exists in the local routing table. **metric** (optional): sets the metric value for the introduced default route. **metric-type** (optional): sets the metric type for the introduced default route. **route-map** (optional): filters and sets the introduced default route.
Qtech(config-router)# no default-information originate [always] [metric metric] [metric-type type] [route-map map-name]	Cancels the operation for introducing the default route to the routing protocol process and advertising the default route.

1.2.3 Configuring Route Filtering

The route filtering is the process to control the inbound/outbound routes so that the device only learns the necessary and predictable routes, and only advertises the necessary and predictable routes to external trusted devices. The divulgence and chaos of the routes may affect the running of the network. Therefore, it is essential to configure the route filtering, especially for telecom operators and on financial service networks.

1.2.3.1 Controlling Route Updating Advertising

To prevent other routers on a local network from learning unnecessary information, you can control the route updating advertisement to prevent the specified route from updating.

To prevent the route updating advertisement, execute the following commands in routing process configuration mode:

Command	Function
Qtech(config-router)# **distribute-list** {[*access-list-number* | *access-list-name*] **prefix** *prefix-list-name*} **out** [*interface-type interface-number*| *protocol*]	According to the ACL rules, permits or denies some routes. **prefix**: This keyword specifies the prefix list for filtering the routes. The prefix list should be separately configured by using the ip prefix-list command.
Qtech(config-router)# **no distribute-list** {[*access-list-number* | *access-list-name*] **prefix** *prefix-list-name*} **out** [*interface-type interface-number*| *protocol*]	Cancels the operation for preventing the route updating advertising.

When you configure the OSPF, you cannot specify the interface. This feature is only applicable to the external routes in the OSPF routing area.

1.2.3.2 Controlling the Process of Route Updating

To avoid processing some specified routes of the inbound route updating packets, you can configure this feature, which does not apply to the OSPF routing protocols.

To control the route updating processing, execute the following commands in routing process configuration mode:

Command	Function
Qtech(config-router)# **distribute-list** {[*access-list-number* | *access-list-name*] | **prefix** *prefix-list-name* [**gateway** *prefix-list-name*]| **gateway** *prefix-list-name*} **in** [*interface-type interface-number*]	According to the ACL rules, permits or denies receiving the specified inbound routes. **prefix**: This keyword specifies the prefix list for filtering the routes. The prefix list should be separately configured by using the ip prefix-list command. **gateway**: Uses the prefix list to filter the inbound routes according to the route sources.
Qtech(config-router)# **no distribute-list** {[*access-list-number* | *access-list-name*] | **prefix** *prefix-list-name* [**gateway** *prefix-list-name*]| **gateway** *prefix-list-name*} **in** [*interface-type interface-number*]	Cancels the operation for controlling the process of the route updating.

1.3 Configuring Fast Reroute

When a link or router fails, the packets that need to forwarded through this link or router will be lost or a loop will be generated to cause service suspension. This problem can be avoided by configuring the router with the static fast reroute function.

Please refer to *Configuring OSPF* for OSPF fast rerouting configuration details.

Run the following commands to enable a static fast reroute in global configuration mode.

Command	Function
Qtech(config)# route-map fast-reroute	Enters the route map configuration mode.
Qtech(config-route-map)# set fast-reroute backup-nexthop GigabitEthernet 0/1 192.168.1.2	Configures a backup interface and backup next hop of the fast reroute.
Qtech(config-route-map)# exit	Returns to the global configuration mode.
Qtech(config)# ip fast-reroute route-map fast-reroute	Configures a static fast reroute.

1.4 Configuring the Key Chain

The key chain is used to manage authentication keys, assign key IDs, and specify the authentication string and the lifetime in the sending and receiving directions. Each key is identified and stored using a unique ID.

To manage the authentication keys, run the following commands in global configuration mode.

Command	Function
Qtech(config)# **key chain** *key-chain-name*	Configures the key chain.
Qtech(config-keychain)# **key** *key-id*	Configures the key ID.
Qtech(config-keychain-key)# **key-string** [**0**|**7**] *text*	Configures the authentication string.
Qtech(config-keychain-key)# **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	Configures the lifetime in the receiving direction.
Qtech(config-keychain-key)# **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	Configures the lifetime in the sending direction.
Qtech(config-keychain-key)# **end**	Exits key chain configuration mode.
Qtech# **show key chain**	Shows configuration information about key chains.

1.5 Configuration Examples

1.5.1 Example of Route-Map Configuration

The route-map can be configured very flexibly and applies to the route redistribution and the policy-based routing configuration. No matter how the route-map is used, the configuration principle is the same, except that different command sets are used. Even if the route-map is used for the route redistribution, different routing protocols can use different commands.

The following examples show that the RIP routes are redistributed based on the OSPF routing protocols. It is required that only the RIP routes whose hops are 4 be redistributed. The type of these routes is the external route type-1, the default metric value is 40, and the route tag is set to 40 in the OSPF routing area.

# Configure the OSPF.

```
Qtech(config)# router ospf 1
Qtech(config-router)# redistribute rip subnets route-map redrip
Qtech(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

# Configure the access control list.

```
Qtech(config)# access-list 10 permit 200.168.23.0 0.0.0.255
```

# Configure the route-map.

```
Qtech(config)# route-map redrip permit 10
Qtech(config-route-map)# match metric 4
Qtech(config-route-map)# set metric 40
Qtech(config-route-map)# set metric-type type-1
Qtech(config-route-map)# set tag 40
```

The following examples show that the OSPF routes are redistributed based on the RIP routing protocols. It is required that only the OSPF routes whose tag is 10 be redistributed. The default metric value of these routes is set to 10.

# Configure the RIP.

```
Qtech(config)# router rip
Qtech(config-router)# version 2
Qtech(config-router)# redistribute ospf 1 route-map redospf
Qtech(config-router)# network 200.168.23.0
```

# Configure the route-map.

```
Qtech(config)# route-map redospf permit 10
Qtech(config-route-map)# match tag 10
Qtech(config-route-map)# set metric 10
```

The following examples show that the OSPF routes are redistributed based on the RIP routing protocols. Since rules that are not supported in the route-map application have been configured, when the route redistribution is associated with the route-map, a prompt displays indicating that the application does not support the rules.

# Configure the route-map.

```
Qtech(config)# route-map redrip permit 10
Qtech(config-route-map)# match length 1 3
Qtech(config-route-map)# match route-type external
Qtech(config-route-map)# set level backbone
```

# Configure the OSPF.

```
Qtech(config)# router ospf 1
Qtech(config-router)# redistribute rip subnets route-map redrip
% ospf redistribute rip not support match length
% ospf redistribute rip not support match route-type
% ospf redistribute rip not support set level backbone
```

1.5.2 Example of the Static Route Redistribution

- Configuration Requirements

A device exchanges route information with other devices through the RIP. In addition, there are three static routes that require the route redistribution based on the RIP. The RIP only allows to advertise two routes 172.16.1.0/24 and 192.168.1.0/24.

- Specific Configurations of the Routers

This is a common configuration example in practice for route filtering according to the distribution list. Note that the metric value is not specified for the routes to be redistributed. Since a static route is redistributed, the RIP automatically assigns the metric value. During the RIP configuration, the version must be specified and the route aggregation must be disabled because the access list allows the 172.16.1.0/24 route. To advertise the route externally, the RIP protocol must first support the classless route, and the route cannot be aggregated to the 172.16.0.0/16 network.

# Configure the static route.

```
Qtech(config)# ip route 172.16.1.0 255.255.255.0 172.200.1.2
Qtech(config)# ip route 192.168.1.0 255.255.255.0 172.200.1.2
Qtech(config)# ip route 192.168.2.0 255.255.255.0 172.200.1.4
```

# Configure the RIP.

```
Qtech(config)# router rip
Qtech(config-router)# version 2
Qtech(config-router)# redistribute static
Qtech(config-router)# network 192.168.34.0
Qtech(config-router)# distribute-list 10 out static
Qtech(config-router)# no auto-summary
```

# Configure the extended ACL.

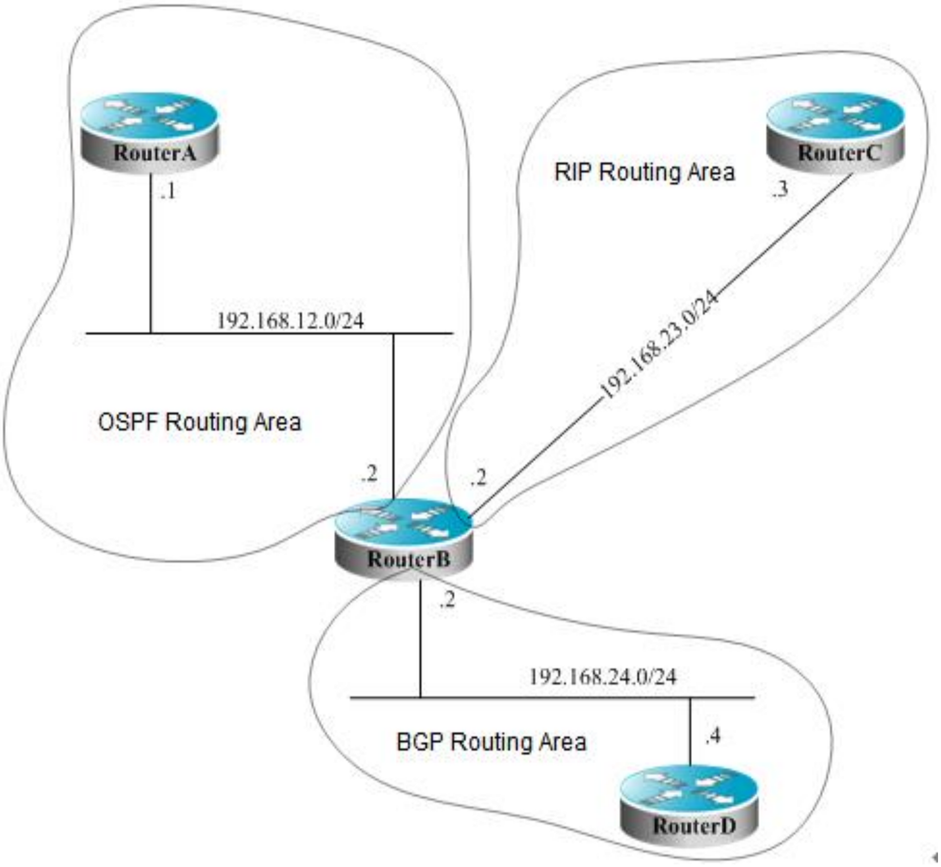
```
Qtech(config)# ip access-list extended EXT_ACL
Qtech(config-ext-nacl)#10 permit ip 192.168.1.0 0.0.0.255
any
Qtech(config-ext-nacl)#10 permit ip 172.16.1.0 0.0.0.255 any
```

1.5.3 Example of Dynamic Routing Protocol Redistribution

- Configuration Requirements

The connection among four routers is shown in Figure 1. Router A belongs to the OSPF routing area. Router C belongs to the RIP routing area. Router D belongs to the BGP routing area. Router B is connected to the three routing areas. Router A advertises the two routes 192.168.10.0/24 and 192.168.100.1/32. Router C advertises the two routes 200.168.3.0/24 and 200.168.30.0/24. Router D advertises the two routes 192.168.4.0/24 and 192.168.40.0/24.

Figure 1 Dynamic routing protocol redistribution



On Router B, the OSPF redistributes the RIP routes with the route type of Type-1, redistributes the BGP routes whose community attribute is 11:11 in the BGP routing area. The RIP redistributes the 192.168.10.0/24 route whose metric value is set to 3 in the OSPF routing area, and advertises a default route to the RIP routing area.

- Specific Configurations of the Routers

When the routing protocols redistribute the routes among each other, the simple route filtering can be controlled by using the distribution list. However, different attributes must be set for different routes, which cannot be implemented by using the distribution list. In this case, a route-map must be used for control. The route-map provides more control functions than the distribution list, but the router configuration is more complex. Therefore, do not use the route-map if possible. The following examples use the route-map to match the community attribute of the BGP routes.

Configurations of Router A:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.10.1 255.255.255.0
Qtech(config)# interface loopback 1
Qtech(config-if)# ip address 192.168.100.1 255.255.255.255
Qtech(config-if)# no ip directed-broadcast
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF.

```
Qtech(config)# router ospf 12
Qtech(config-router)# network 192.168.10.0 0.0.0.255 area 0
Qtech(config-router)# network 192.168.12.0 0.0.0.255 area 0
Qtech(config-router)# network 192.168.100.0 0.0.0.255 area 0
```

Configurations of Router B:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.12.2 255.255.255.0
Qtech(config)# interface Serial 1/0
Qtech(config-if)# ip address 192.168.23.2 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ip address 192.168.24.2 255.255.255.0
```

#Configure the OSPF and specify the redistribution route type.

```
Qtech(config)# router ospf 12
Qtech(config-router)# redistribute rip metric 100 metric-type 1 subnets
Qtech(config-router)# redistribute bgp route-map ospfrm subnets
Qtech(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

#Configure the RIP and use the distribution list to filter the redistributed routes.

```
Qtech(config)# router rip
Qtech(config-router)# redistribute ospf 12 metric 3
Qtech(config-router)# network 192.168.23.0
Qtech(config-router)# distribute-list 10 out ospf
Qtech(config-router)# default-information originate always
Qtech(config-router)# no auto-summary
```

# Configure the BGP.

```
Qtech(config)# router bgp 2
Qtech(config-router)# neighbor 192.168.24.4 remote-as 4
Qtech(config-router)# address-family ipv4
Qtech(config-router-af)# neighbor 192.168.24.4 activate
Qtech(config-router-af)# neighbor 192.168.24.4 send-community
```

# Configure the route-map.

```
Qtech(config)# route-map ospfrm
Qtech(config-route-map)# match community cl_110
```

# Define the access list.

```
Qtech(config)# access-list 10 permit 192.168.10.0
```

# Define the community list.

```
Qtech(config)# ip community-list standard cl_110 permit 11:11
```

Configurations of Router C:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.30.1 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ip address 192.168.3.1 255.255.255.0
Qtech(config)# interface serial 1/0
Qtech(config-if)# ip address 192.168.23.3 255.255.255.0
```

# Configure the RIP.

```
Qtech(config)# router rip
Qtech(config-router)# network 192.168.23.0
Qtech(config-router)# network 192.168.3.0
Qtech(config-router)# network 192.168.30.0
```

Configurations of Router D:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.40.1 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ip address 192.168.4.1 255.255.255.0
Qtech(config)# interface serial 1/0
Qtech(config-if)# ip address 192.168.24.4 255.255.255.0
```

# Configure the BGP.

```
Qtech(config)# router bgp 4
Qtech(config-router)# neighbor 192.168.24.2 remote-as 2
Qtech(config-router)# redistribute connected route-map bgprm
Qtech(config-router)# address-family ipv4
Qtech(config-router-af)# neighbor 192.168.24.2 activate
Qtech(config-router-af)# neighbor 192.168.24.2 send-community
```

# Configure the route-map.

```
Qtech(config)# route-map bgprm
Qtech(config-route-map)# match community 22:22
```

The OSPF routes found on Router A:

```
O E1 192.168.30.0/24 [110/101] via 192.168.12.2, 00:04:07, FastEthernet0/1
O E1 192.168.3.0/24 [110/101] via 192.168.12.2, 00:04:07, FastEthernet0/1
```

The RIP routes found on Router C:

```
R    0.0.0.0/0 [120/1] via 192.168.23.2, 00:00:00, Serial1/0
R    192.168.10.0/24 [120/2] via 192.168.23.2, 00:00:00, Serial1/0
```

1.5.4 Example of Fast Reroute Configuration

- Configuration Requirements

Three routers A, B and C are connected with each other via static routes. It is required that when the link between B and C fails, the service will be fast removed to the link between A and C and reach B after passing through C and A.

- Specific Configurations

Router A:

# Configure an Ethernet interface

```
Qtech(config)interface GigabitEthernet 0/1
Qtech(config-if)ip address 192.168.1.1 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
Qtech(config)interface GigabitEthernet 0/2
Qtech(config-if)ip address 192.168.2.1 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
```

# Configure a static route

```
Qtech(config)ip route 1.1.1.1 255.255.255.255 192.168.2.2
```

Router B:

# Configure an Ethernet interface

```
Qtech(config)interface GigabitEthernet 0/1
Qtech(config-if)ip address 192.168.2.2 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
Qtech(config)interface GigabitEthernet 0/2
Qtech(config-if)ip address 192.168.3.1 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
Qtech(config)interface loopback 1
Qtech(config-if)ip address 1.1.1.1 255.255.255.255
```

Router C:

# Configure an Ethernet interface

```
Qtech(config)interface GigabitEthernet 0/1
Qtech(config-if)ip address 192.168.1.2 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if)carrier-delay 0
Qtech(config)interface GigabitEthernet 0/2
Qtech(config-if)ip address 192.168.3.2 255.255.255.0
Qtech(config-if)bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if)carrier-delay 0
```

# Configure a route map

```
Qtech(config)access-list 1 permit host 1.1.1.1
Qtech(config)route-map frr
Qtech(config-route-map)match ip address 1
Qtech(config-route-map)set fast-reroute backup-nexthop GigabitEthernet 0/1 192.168.1.1
```

# Configure a static route and a fast reroute

```
Qtech(config)ip fast-reroute route-map frr
Qtech(config)ip route 1.1.1.1 255.255.255.255 192.168.3.1
```

2 CONFIGURING POLICY-BASED ROUTING

2.1 Understanding Policy-based Routing

2.1.1 Overview

Policy-based Routing offers a more flexible packet routing forwarding mechanism than destination address-based routing forwarding, which enables you to route IPv4/IPv6 packets by elements like source address, destination address, port number and packet length.

In general, user networks apply different bandwidths from different ISPs. Meanwhile, to ensure resources for important users in the same user environment, the system needs to selectively forward packets rather than forwarding packets by the general routing table. In this case, policy-based routing takes full advantages of ISP resources and satisfy these flexible and diversified applications.

IP/IPv6 policy-based routing takes effect only on the packets received on interfaces, without any control on the packets sent from interfaces. Applying policy-based routing on an interface will check all the packets received on the interface. The packets not matching any policy of the routing map are forwarded by the general routing table, but the ones matching some policy of the routing map are forwarded by the policy.

Generally, policy-based routing takes preference over general routing and forwards IP/IPv6 packets in accord with defined policies. In other words, packets are forwarded by IP/IPv6 policy-based routing. If no rule of the PBR is matched, the packets are forwarded by general routings. Certainly, users can configure policy-based routing with the priority lower than general routing. Namely, the packets received on an interface are forwarded by general routing or policy-based routing in cast of no matching.

Users can configure forwarding mode like load balance or redundant backup according to real circumstances. Load balance or redundant backup is enabled on more than one next hop. The proportion of load balance can be also set. In redundant backup mode, multiple next hops are applied, that is the previous next hop has priority to take effect and the latter one takes effect only when the previous one fails. You can configure multiple next hops at the same time.

Policy-based routing falls into two types:

Policy-based routing enabled for the IP packets received on an interface. It performs PBR only on packets received on the interface instead of controlling the packets sending from the interface.

Policy-based routing enabled for the IP packets that the local device sends out. It controls IP packets sent from the local device to other devices, not the packets sent from external devices to the local device.

2.1.2 Basic Concepts and Features

2.1.2.1 Application Process

To use the policy-based routing, you must create a routing map for it and then apply the routing map on the interface. A routing map consists of many policies with corresponding sequence. Smaller sequence means higher priority.

Each policy consists of one or more match statements and corresponding one or more set statements. The match statement defines the matching rule of IP/IPv6 packets, and the set statement defines the processing rules of matched IP/IPv6 packets. In the course of policy-based routing, packets are matched by priorities in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing.

Policy-based routing for IPv4 packets uses standard or extended ACL as matching rule. Policy-based routing for IPv6 packets, however, uses extended ACL as matching rule. For IPv6 packets, only one match ipv6 address can be configured for a policy at most.

2.1.2.2 Routing Map Policy Matching Mode

When you configure the routing map, you can specify the match mode of a policy as permit or deny, which is described as below:

- Permit: Specify the matching mode as permit, that is to apply the corresponding set rule to the IPv4/v6 packets meeting the match rules of the policy. If no match rule is met, the system applies the next policy to packets.
- Deny: Specify the matching mode as deny, that is if IPv4/v6 packets meet all match statements; the system performs common routing rather than policy-based routing.

IP/IPv6 packets are matched by the priority of every policy of the routing map in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing. If the packets do not match any policy of the routing map, the system performs common routing.

2.1.2.3 Next Hop Rules

Policy-based routing offers two forwarding rules-set {ip | ipv6} next-hop and set {ip | ipv6} default next-hop, which set the next hop and the egress, respectively. These two rules are described as follows:

- set {ip | ipv6} next-hop: Configure the policy-based routing's next hop IPv4/IPv6 address, which takes precedence over common routes. The IPv4/v6 packets meeting the match rule received on the interface are first forwarded to the next hop specified by the set {ip | ipv6} next-hop command, no matter whether the real routing of the packets in the routing table and the next hop specified by the policy-based route is valid or not.
- set {ip | ipv6} default next-hop: The policy-based routing specified by this command is of the priority lower than common routes but higher than default route. For the packets meeting the match rule received on the interface, if routing in the routing table is failed or the default route is used, these packets will be forwarded to the next hop specified by this command.

The next hops specified by these two rules must be direct or otherwise the configuration does not take effect.

The priority is subject to the order of set {ip | ipv6} next-hop > network route/host route > set {ip | ipv6} default next-hop > default route. These two commands can be configured simultaneously, but only the one of higher priority takes effect.

2.1.2.4 Load Balancing Mode for Policy-based Routing Next Hop

More than one next hop can be configured in the sequence of a route map, and one of the following load balancing modes can be configured among them.

- Redundant backup: Only one next hop takes effect at a time if there are many next hops. Once the active next hop failed, another next hop will take over its works immediately.
  - When R1 of the active next hop fails, the system automatically hands over to R2 of the next next hop. When R1 recovers, the system will automatically hand over back to R1.
  - When there are many next hops in the order, for example, R1/R2/R3, R2 takes effect after you deleting and then adding R1 in the order of R2/R3/R1.
- Load balancing: Load balancing is enabled among next hops by traffic. This function is not available for the next hop in egress type.

⚠ Caution

1 Only one route map can be configured on a port. Configuring route maps repeatedly on a port will overlap the previous configurations, namely that the latest configuration takes effect.

⚠ Caution

2 Only one IPv6 ACL can be configured in the sequence of a route map in sub route map.

⚠ Caution

3 If the sub route map is configured with next hop but not ACL, all packets are matched; if the sub route map is configured with ACL but not next hop, the matched packets are forwarding by common routes; if the sub route map is not configured with ACL and next hop, all packets are forwarded by common routes.

⚠ Caution

4 The deny rule of ACE forwards packets by common routes. To meet the matching rule of policy-based routing, the deny any any command matches packets starting from the next IPv6 ACL

⚠ Caution

5 Enabling PBR will apply to incoming packets at the same time. If you do not need to apply PBR to a specific incoming IPv4/v6 packet, add “deny the specific IPv4/v6 address” in the ACL manually

⚠ Caution

6.In redundant backup mode, the IP packets matching the policy of the sub route map are forwarded to the next hop firstly resolved in the sequence. If all next hops are not resolved, the IP packets matching the policy are discarded. If the first next hop is resolved later, the IP packets matching the policy are forwarded to the first next hop.

Note

For details on the next hop of PBR set actions, refer to Rns&track Configuration Guide and Rns&track Command Reference or Link Detection Configuration Guide and DLDP Command Reference (for routers). IPv6 PBR is not supported at present

Note

For linkup of PBR and BFD, refer to BFD Configuration Guide and BFD Command Reference

2.1.3 Enabling Track Function

Track function can increase the insight of policy-based routing in the change of networks. When the device perceives that the next hop for forwarding failed, policy-based routing will rapidly hand the traffic over to the next valid next hop (in redundant backup mode) or all other valid next hops (in load balancing mode).

For track configuration, refer to Rns&track Configuration Guide. IPv6 PBR does not support linkup with track.

2.1.4 Enabling BFD Function

Linkup between policy-based routing and BFD avoids setting the policy-based routing as forwarding path when it is not reachable. If the backup forwarding path is available, the system rapidly hands over to this path.

2.1.5 VRF Selection Using Policy-based Routing

The PBR implementation of the VRF selection feature allows the ports that apply PBR to filter the packets based on the matching rule. If the packet matches this rule, route selection is performed in the specified VRF. Matching rule is defined in an IP access list or based on packet length. Users can balance traffic on different VRF instances as required.

In general, the packets received on an interface of a VRF are routed and forwarded through this VRF. The packets received on an interface of the global routing table are routed and forward through the global routing table. VRF selection using policy based routing can remove this limit. This feature supports VRF successor route, the route across VRFs and the route from VRF to the global routing table. In VRF successor route mode, the packets received on an interface of a VRF are routed and forwarded by the routing table of this VRF. In route across VRFs mode, the packets received on an interface of a VRF are routed and forwarded by the routing table of another specified VRF. In route from VRF to the global routing table mode, the packets received on an interface of a VRF are routed and forwarded by the global routing table.

Version 10.4(3) introduces multi-protocol VRF, which supports VRF selection using IPv6 PBR. If a single-protocol IPv4 VRF is specified, it does not take effect on IPv6 PBR. When a multi-protocol VRF is specified, if it does not configure the IPv4 address family, the multi-protocol VRF does not take effect on IPv4 PBR. Similarly, if the multi-protocol VRF does not configure the IPv6 address family, it does not take effect on IPv6 PBR. If the multi-protocol VRF configures IPv4 and IPv6 address families concurrently, the rules of the set vrf command take effect on both IPv4 and IPv6 PBRs.

2.1.6 Working Principles

For policy-based routing, first of all, you need to define a route map used to specify the policy of packet forwarding. The route map consists of a set of statements with permit or deny action.

Secondly, define a set of set statements in the route map to forward and control packets in order. Each statement does not refer to the previous or latter statements.

Finally, apply the policy-based routing at the inbound direction. If the policy-based routing is applied at the outbound direction, packets are forwarded by common routes.

For routers, outgoing packets can be processed by the specific policy-based routing, not the common routing table.

2.1.7 Protocol Specifications

None

2.2 Default Configurations

The default configurations of policy-based routing are described as follows:

Function	Default value
Load balance of many next hops	Redundance (redundant backup mode)
Next hop WCMP weight	1

2.3 Configuring Policy-based Routing

The following sections configure the basic functions of IP/IPV6 PBR.

2.3.1 Configuring IPv4 Policy-based Routing

You must specify a route map for the policy-based routing and create the route map before applying the policy-based routing. A route map consists of many policies with corresponding sequences. The smaller the sequence, the higher the priority is. Each policy consists of one or more match statements and corresponding one or more set statements. The match statement defines the matching rule of IPv4/IPv6 packets, and the set statement defines the processing rules of matched IPv4/IPv6 packets. In the course of policy-based routing, packets are matched by priorities in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing.

There are two kinds of match statements; match length and match ip address. The former statement matches packets by packet length and the latter matches packets by ACL. For a policy, you can configure only one match length statement but many match ip address statements. If both statements are configured at the same time, the action specified by the set rule of the policy is executed only when the packets match both.

Similarly, there are two types of set statements. Type 1 modifies the QoS field of IP packet, including set ip tos, set ip precedence and set ip dscp. Type 2 controls IP packet, for example, set vrf, set ip nexthop, set ip default nexthop, set interface and set default interface. Once all match rules are met, Type 1 set statements must be executed and Type 2 set statements are executed by priority in the following order:

- set vrf: Set policy-based routing as the VRF instance for IP packet routing with the priority higher than common route. The command is mutually exclusive with the set ip [default] nexthop and set [default]interface command. The IPv4 packets received on the interface that meet match rules will be routed by the routing table of the VRF instance specified by this command, no matter whether the VRF is the same as the one the interface belongs to.
- set ip nexthop: Set next hop of policy-based routing with the priority higher than common route and the one set by the set interface command. This command takes precedence over one of the following three commands. The IPv4 packets received on the interface that meet match rules will be firstly forwarded to the next hop specified by the set ip nexthop command, no matter whether the real routing of IPv4 packets in the routing table is the same as the one specified by the policy-based routing.
- set interface: Set the egress of policy-based routing with the priority higher than common route. This command takes precedence over set default interface and set ip default nexthop. The IPv4 packets received on the

interface that meet match rules will be firstly forwarded through the egress specified by the set interface command, no matter whether the real routing of IPv4 packets in the routing table is the same as the egress specified by the policy-based routing.

- set default interface: Set the default interface with the priority higher than default route and the one specified by the set ip default nexthop command but lower than common route. The IPv4 packets received on the interface that meet match rules will be forwarded through the interface specified by this command in case of routing failure or the default route is used.
- set ip default nexthop: Set the policy-based routing with the priority higher than the default route but lower than common route. The IPv4 packets received on the interface that meet match rules will be forwarded to the next hop specified by this command in case of routing failure or the default route is used.

When you configure the routing map, you can specify the match mode of a policy as permit or deny, which is described as below:

- Permit: Specify the matching mode as permit, that is to apply the corresponding set rule to the IPv4/v6 packets meeting all match rules of the policy. If not all match rules are met, the system applies the next policy of the routing map to match packets.
- Deny: Specify the matching mode as permit, that is if IPv4/v6 packets meet all match statements of this packet's node, the system performs common routing rather than policy-based routing.

IPv4/IPv6 packets are matched by the priority of every policy of the routing map in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing. If the packets do not match any policy of the routing map, the system performs common routing.

The next hop specified by the set ip nexthop command is used for forwarding only when its tracking object is active. Track function greatly increases the insight of policy-based routing in the change of network environments, enabling PBR to adapt to dynamic changed networking topologies.

To configure a policy-based routing, perform the following steps:

1 Define an ACL as the matching rule of IP packets.

Command	Function
Qtech(config)# ip access-list {extended | standard} {id | name}	Defines an ACL as the matching rule of IP packets.

2 Define a route map, which consists of many policies in sequence order. When a policy is matched, the system quits the execution of the route map.

Use the following command in global configuration mode to define a route map.

Command	Function
Qtech(config)# route-map route-map-name [permit | deny] sequence	Defines a route map.
Qtech(config)# no route-map route-map-name {[permit | deny] sequence}	Deletes a route map.

3 Define the match rule of every policy of the route map.

Use the following command in route map configuration mode to define the match rule of a policy.

Command	Function
Qtech(config-route-map)# match ip address {access-list-number | access-list-name}	Matches the address in the ACL.
Or : Qtech(config-route-map)# match length min max	Matches packet length.

4 Define the action after meeting match rule.

Use the following command in route map configuration mode to define actions after rules are matched.

Command	Function
Qtech(config-route-map)# set vrf name	Routes the packets matching PBR by the routing table of the specific VRF instance.
Qtech(config-route-map)# set ip next-hop ip-address [weight][ip-address[weight]]	Sets the next hop IP address of packets.
Qtech(config-route-map)# set interface intf_name	Sets the egress of packets.
Qtech(config-route-map)# set ip default next-hop ip-address[weight] [ip-address[weight]]	Sets the next hop IP address for the packets without route.
Qtech(config-route-map)# set default interface intf_name	Sets the default egress of IP packets.
Qtech(config-route-map)# set ip precedence	Modifies the priority of IP packet.
Qtech(config-route-map)# set ip tos	Modifies the ToS value of IP packet.
Qtech(config-route-map)# set ip dscp	Modifies the DSCP value of IP packet.

⚠ Caution

The set vrf, set ip [default] nexthop and set [default] interface commands cannot be configured concurrently for a policy. But the set vrf command can be configured with other set statements. The VRF must exist when you configure the VRF of policy-based routing otherwise the system prompts configuration failure.

⚠ Caution

The set ip dscp, set ip tos and set ip precedence commands cannot be configured concurrently for a policy or otherwise the corresponding domains of IP packet may be different from the expectation

⚠ Caution

The priorities of the set vrf, set ip nexthop, and set interface commands take precedence over that of common routes. IP packets matching policy-based routing are forwarded by policy-based routing, but the IP packets not matching the policy-based routing are forwarded by common routes.

⚠ Caution

The set default ip nexthop and set default interface commands are lower than common route in terms of priority. IP packets are routed and forwarded by policy-based routing only after common route failed

For details on route map configuration, refer to Protocol-independent Configuration Commands.

1 Apply the route map on the specified interface.

Use the following command in interface configuration mode to apply the policy-based routing on the specified interface.

Command	Function
Qtech(config-if)# ip policy route-map name	Applies policy-based routing on the interface.
Qtech(config-if)# no ip policy route-map	Removes the configuration.

2. Apply the policy-based routing to the packets sent locally.

Command	Function
Qtech(config)# ip local policy route-map [name]	Applies the policy-based routing to the packets sent locally.
Qtech(config)# no ip local policy route-map	Removes the configuration.

For example:

Configure policy-based routing on Fastethernet 0/0 so that all incoming packets are forwarded to the device whose next hop is 192.168.5.5.

```
Qtech(config)# access-list 1 permit any
Qtech(config)# route-map name
Qtech(config-route-map)# match ip address 1
Qtech(config-route-map)# set ip next-hop 192.168.5.5
Qtech(config-route-map)# int fastethernet 0/0
Qtech(config-if)# ip policy route-map name
```

3 Configure load balancing mode for policy-based routing

In redundant backup mode, the policy-based routing will automatically hand over the next valid next hop when the active next hop fails. In load balancing mode, on contrary, the traffic will be balanced on other valid next hop when the active next hop fails.

Use the following command in global configuration mode to configure load balance or redundant backup:

Command	**Function**
Qtech(config)# ip policy {load-balance | redundance}	Configures load balance or redundant backup for policy-based routing forwarding.
Qtech(config)# no ip policy	Removes the configuration.

⚠ Caution

In load balancing mode, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

⚠ Caution

In load balancing mode, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

⚠ Caution

For default policy-based routing, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

⚠ Caution

In redundant backup mode, the first resolved next hop takes effect. If all next hops are not resolved, the packets matching policy-based routing are dropped. If the originally unresolved next hop of higher priority than active next hop is resolved, the system hands over to this next hop.

2.3.2 Configuring IPv6 Policy-based Routing

Command	Function
Qtech#configure terminal	Enters global configuration mode.
Qtech(config)#ipv6 access-list access-list-name	Creates an IPv6 ACL.
Qtech (config)#route-map route-map-name [permit | deny] sequence	Creates a route map.
Qtech (config-route-map)#match ipv6 address access-list-name	Matches the IPv6 address in ACL.
Qtech (config-route-map)#set ipv6 [vrf vrf-name | global] next-hop global-ipv6-address [weight][global-ipv6-address [weight]] [global-ipv6-address...]	Sets the next hop IPv6 address of packets. The [vrf vrf-name | global] parameter is supported since version 10.4(3), that is, cross VRFs and from VRF to global modes are supported. If the vrf vrf-name parameter is specified, the next hop belongs to the VRF, while if the global parameter is specified, the next hop belongs to the global. Note that the specified VRF must be a multi-protocol VRF whose IPv6 address family has been configured. VRF will be inherited when forwarding IPv6 packets if the set ipv6 next-hop command is configured. The next hop belongs to the VRF that receive the IPv6 packets and forward them internally.
Or: Qtech (config-route-map)#set ipv6 default next-hop global-ipv6-address [weight][global-ipv6-address [weight]] [global-ipv6-address...]	Specifies the next hop IPv6 address for the packets without obvious routes in the routing table.
Qtech (config)#interface interface-type interface-number	Enters the interface require applying PBR.
Qtech (config-if- interface-type interface-number)#ipv6 policy route-map route-map-name	Applies policy-based routing on the interface.
Or: Qtech (config-if- interface-type interface-number)#no pv6 policy route-map	Removes the policy-based routing applied on the interface.
Qtech#show ipv6 policy	Shows the configuration of policy-based routing.
Or: Qtech#show route-map	Shows the configuration of route map.

For route map configuration, refer to *Configuring Protocol-independent*.

2.3.3 Configuring Load Balancing Mode

Command	Function
Qtech#configure terminal	Enters global configuration mode.
Qtech(config)#Ipv6 policy [load-balance | redundance]	Configures load balance mode.
Qtech(config)#no Ipv6 policy	Restores the setting to the default value.

2.3.4 Displaying Configuration and States

Command	Function
Qtech#**show** { **ip** | **ipv6** } **policy**	Shows the configuration of policy-based routing.
Qtech#**show route-map**	Shows the configuration of route map.
Qtech#**show access-lists**	Shows the configuration of ACL.

2.4 Typical Configuration Examples

2.4.1 Example 1: Source address based PBR

2.4.1.1 Networking Requirements

There are two egresses of a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses. All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1 and all streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2. If GigabitEthernet 0/1 is disconnected, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

2.4.1.2 Networking Topology

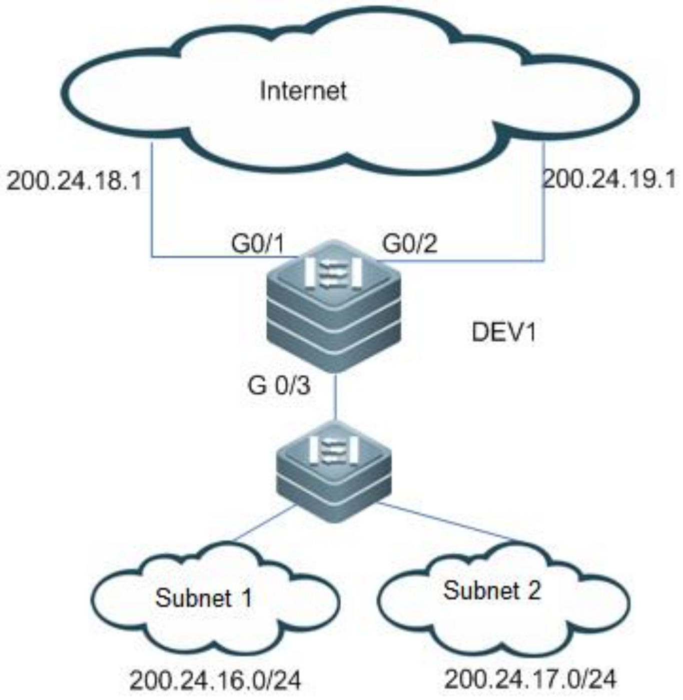


Figure 1 Network topology

As shown in the Figure-1, Layer-3 device DEV1 connects to subnets 1 and 2 through G0/3, and connects to the Internet through G0/1 and G0/2 with the next hop of 200.24.18.1 and 200.24.19.1, respectively. Subnet 1's segment is 200.24.16.0/24 and subnet 2's segment is 200.24.17.0/24.

2.4.1.3 Configuration Steps

# Create ACLs for subnet 1 and subnet 2, respectively.

```
Qtech(config)#access-list 1 permit 200.24.16.0 0.0.0.255
Qtech(config)#access-list 2 permit 200.24.17.0 0.0.0.255
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer.

```
Qtech(config)#route-map RM_FOR_PBR 10
Qtech(config-route-map)#match ip address 1
Qtech(config-route-map)#set ip nexthop 200.24.18.1
Qtech(config-route-map)#set ip nexthop 200.24.19.1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Qtech(config)#route-map RM_FOR_PBR 20
Qtech(config-route-map)#match ip address 2
Qtech(config-route-map)#set ip nexthop 200.24.19.1
Qtech(config-route-map)#set ip nexthop 200.24.18.1
```

# Configure redundant backup.

```
Qtech(config)#ip policy redundance
```

# Apply policy-based routing on GigabitEthernet 0/3.

```
Qtech(config)#interface GigabitEthernet 0/3
Qtech(config-if)#ip policy route-map RM_FOR_PBR
```

2.4.2 Example 2: Enabling Track function.

2.4.2.1 Networking Requirements

There are two egresses on a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses. All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1 and all streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2. If the next hop 200.24.18.1 fails, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

2.4.2.2 Networking Topology

As shown in Figure 1.

2.4.2.3 Configuration steps

# Track the egress GigabitEthernet 0/1’s next hop 200.24.18.1.

```
Qtech(config)#ip rns 1
Qtech(config-ip-rns)#icmp-echo 200.24.18.1
Qtech(config)#track 1 rns 1
```

# Track the egress GigabitEthernet 0/2’s next hop 200.24.19.1.

```
Qtech(config)#ip rns 2
Qtech(config-ip-rns)#icmp-echo 200.24.19.1
Qtech(config)#track 2 rns 2
```

# Enable the routing map to use this track object..

```
Qtech(config)#route-map RM_FOR_PBR 10
Qtech(config-route-map)#match ip address 1
Qtech(config-route-map)#set ip nexthop verify-availability 200.24.18.1 track 1
Qtech(config-route-map)#set ip nexthop verify-availability 200.24.19.1 track 2
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Qtech(config)#route-map RM_FOR_PBR 20
Qtech(config-route-map)#match ip address 1
Qtech(config-route-map)#set ip nexthop verify-availability 200.24.19.1 track 2
Qtech(config-route-map)#set ip nexthop verify-availability 200.24.18.1 track 1
```

# Configure redundant backup.

```
Qtech(config)#ip policy redundance
```

# Apply policy-based routing on GigabitEthernet 0/3.

```
Qtech(config)#interface GigabitEthernet 0/3
Qtech(config-if)#ip policy route-map RM_FOR_PBR
```

2.4.3 Example 3: Configuring VRF Selection Using PBR

2.4.3.1 Networking Requirements

A provider edge (PE) requires applying policy-based routing to the packets received from FastEthernet 0/1. It routes the IP packets from subnet 1 by VRF1, the IP packets from subnet 2 by VRF2, and the IP packets from subnet 3 by VRF3. Other packets are routed in the public network.

2.4.3.2 Configuration Steps

# Create VRF instances.

```
Qtech(config)#ip vrf VRF1
Qtech(config)#ip vrf VRF2
Qtech(config)#ip vrf VRF3
```

# Create ACLs as matching rules of the routing map.

```
Qtech(config)#access-list 1 permit 192.168.195.0 0.0.0.255
Qtech(config)#access-list 2 permit 192.168.196.0 0.0.0.255
Qtech(config)#access-list 3 permit 192.168.197.0 0.0.0.255
```

# Create route maps.

```
Qtech(config)#route-map PBR-VRF-Selection permit 10
Qtech(config-route-map)#match ip address 1
Qtech(config-route-map)#set vrf VRF1

Qtech(config)#route-map PBR-VRF-Selection permit 20
Qtech(config-route-map)#match ip address 2
Qtech(config-route-map)#set vrf VRF2

Qtech(config)#route-map PBR-VRF-Selection permit 30
Qtech(config-route-map)#match ip address 3
Qtech(config-route-map)#set vrf VRF3
```

# Import IP address of the interface to VRFs 1 to 3.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ip address 192.168.195.1 255.255.255.0
Qtech(config-if)#ip vrf receive VRF1
Qtech(config-if)#ip vrf receive VRF2
Qtech(config-if)#ip vrf receive VRF3
```

# Apply the policy-based routing on the interface.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ip policy route-map PBR-VRF-Selection
```

The configurations are as follows if the single protocol IPv4 VRF is replaced with a multi-protocol VRF:

# Create a VRF instance.

```
Qtech(config)#vrf definition VRF1
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#vrf definition VRF2
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#vrf definition VRF3
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#exit
```

# Configure an ACL as the matching rule of the routing map.

```
Qtech(config)#access-list 1 permit 192.168.195.0 0.0.0.255
Qtech(config)#access-list 2 permit 192.168.196.0 0.0.0.255
Qtech(config)#access-list 3 permit 192.168.197.0 0.0.0.255
```

# Configure the routing map.

```
Qtech(config)#route-map PBR-VRF-Selection permit 10
Qtech(config-route-map)#match ip address 1
Qtech(config-route-map)#set vrf VRF1

Qtech(config)#route-map PBR-VRF-Selection permit 20
Qtech(config-route-map)#match ip address 2
Qtech(config-route-map)#set vrf VRF2

Qtech(config)#route-map PBR-VRF-Selection permit 30
Qtech(config-route-map)#match ip address 3
Qtech(config-route-map)#set vrf VRF3
```

# Import IP address of the interface to VRFs 1 to 3.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ip address 192.168.195.1 255.255.255.0
Qtech(config-if)#vrf receive VRF1
Qtech(config-if)#vrf receive VRF2
Qtech(config-if)#vrf receive VRF3
```

# Apply the policy-based routing to the interface.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ip policy route-map PBR-VRF-Selection
```

2.4.4 Example 4: Appling IPv6 Policy-based Routing on the Interface

2.4.4.1 Networking Requirements

There are two egresses on a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses.

Specific requirements are as follows:

- All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1.
- All streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2.
- If GigabitEthernet 0/1 is disconnected, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

2.4.4.2 Networking Topology

As shown in Figure 2, Lay-3 device Device1 connects to subnets 1 and 2 through G0/3 (routed port), and connects to the Internet through G0/1 and G0/2 with the next hop of 2001::1/64 and 2002::1/64 respectively. Subnet 1's segment is 2003::/64 and subnet 2's segment is 2004::/64.

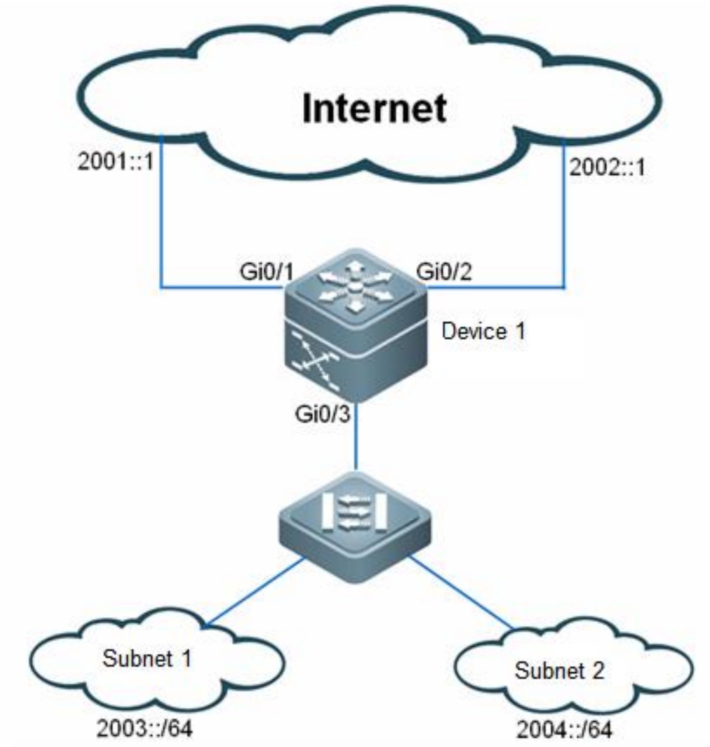


Figure 2 IPV6 PBR topology

2.4.4.3 Configuration Tips

2.4.4.4 Configuration Steps

# Create ACLs for subnet 1 and subnet 2 respectively.

```
Qtech(config)#ipv6 access-list net1
Qtech(config-ipv6-acl)#permit ipv6 2003::/64 any
Qtech(config)#ipv6 access-list net2
Qtech(config-ipv6-acl)#permit ipv6 2004::/64 any
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer.

```
Qtech(config)#route-map RM_FOR_PBR 10
Qtech(config-route-map)#match ipv6 address net1
Qtech(config-route-map)#set ipv6 next-hop 2001::1
Qtech(config-route-map)#set ipv6 next-hop 2002::1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Qtech(config)#route-map RM_FOR_PBR 20
Qtech(config-route-map)#match ipv6 address net2
Qtech(config-route-map)#set ipv6 next-hop 2002::1
Qtech(config-route-map)#set ipv6 next-hop 2001::1
```

# Configure redundant backup.

```
Qtech(config)#ipv6 policy redundance
```

# Apply the policy-based routing on the interface GigabitEthernet 0/3.

```
Qtech(config)#interface GigabitEthernet 0/3
Qtech(config-if-GigabitEthernet 0/3)#ipv6 policy route-map RM_FOR_PBR
```

2.4.4.5 Verification

# Show the configuration of route map.

```
Qtech#show route-map
route-map RM_FOR_PBR, permit, sequence 10
  Match clauses:
    ipv6 address net1
  Set clauses:
    ipv6 next-hop 2001::1 2002::1
route-map RM_FOR_PBR, permit, sequence 20
  Match clauses:
    ipv6 address net2
  Set clauses:
ipv6 next-hop 2002::1 2001::1
```

# Show the configuration of IPv6 policy-based routing.

```
Qtech#show ipv6 policy
Interface                                Route map
GigabitEthernet 0/3                     RM_FOR_PBR
```

# Show the configuration of ACL.

```
Qtech#show access-lists
ipv6 access-list net1
 10 permit ipv6 2003::/64 any
 (0 packets matched)
ipv6 access-list net2
 10 permit ipv6 2004::/64 any
 (0 packets matched)
```

2.4.5 Example 5: Configuring IPv4/IPv6 PBRs Concurrently

2.4.5.1 Networking Requirements

There are two egresses on a LAN connecting to the Internet, one of which is the egress of education network. In general, load balance and backup should be enabled for these two egresses.

Specific requirements are as follows:

- IPv4/IPv6 dual stacks are used in the networks. IPv4 and IPv6 PBRs should be enabled on an interface at the same time.
- All streams from the IPv4 education network of subnet 1 to the Internet are transmitted through the egress of education network.
- All streams from the IPv4 education network of subnet 2 to the Internet are transmitted through the egress of the Internet.
- All streams from the IPv6 education network of subnet 1 to the Internet are transmitted through GigabitEthernet 0/1.
- All streams from the IPv6 education network of subnet 2 to the Internet are transmitted through GigabitEthernet 0/2.
- Internal interactive data, for example, the data from subnet 1 to subnet 2, is transmitted using the internal dynamic route rather than policy-based routing.
- By default, data streams are transmitted through the egress of the Internet by the default route.
- If GigabitEthernet 0/1 fails, the data streams on the interface are switched over to GigabitEthernet 0/2, and vice versa.

2.4.5.2 Networking Topology

As shown in Figure 3, Device 1 connects to subnets 1 and 2 through G0/3 (routed port), and connects to the Internet through G0/1 and G0/2 with the next hop of 2001::1/64 (210.82.12.1) and 2002::1/64 (59.78.184.1) respectively. Subnet 1’s segment is 2003::/64 (202.112.144.0/25) and subnet 2’s segment is 2004::/64(218.62.95.0/24).

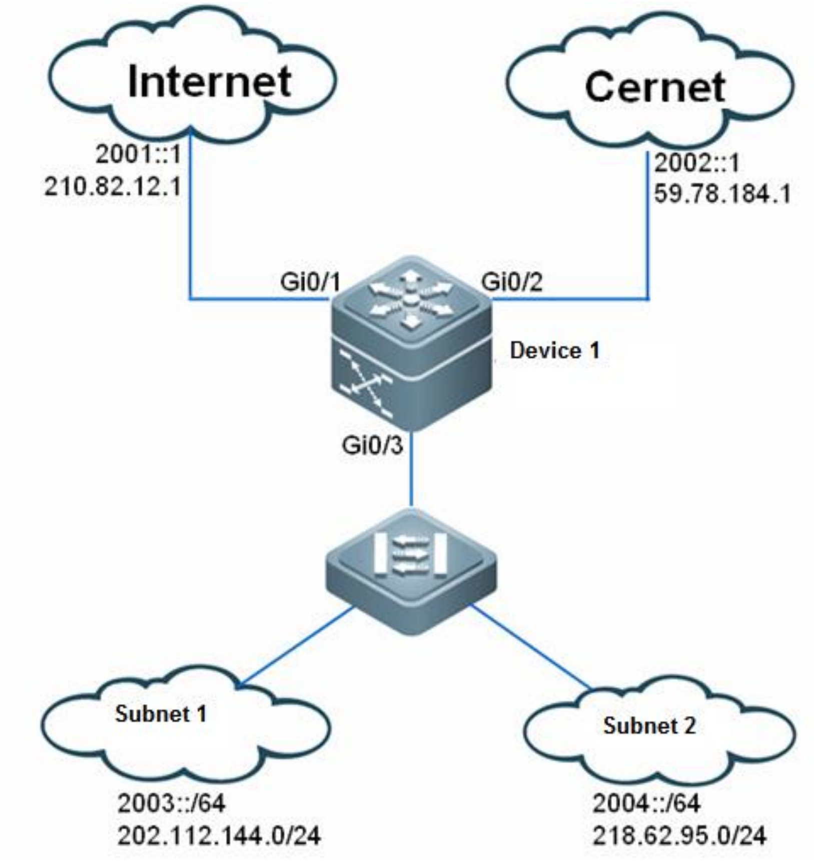


Figure 3 IPv4/IPv6 PBR topology

2.4.5.3 Configuration Tips

2.4.5.4 Configuration Steps

# Create IPv4 ACLs for subnet 1 and subnet 2, respectively.

```
Qtech(config)#ip access-list extended 101
Qtech(config-ip-acl)#permit ip 202.112.144.0 0.0.0.255 any
Qtech(config)#ip access-list extended 102
Qtech(config-ip-acl)#permit ip 218.62.95.0 0.0.0.255 any
```

# Create IPv6 ACLs for subnet 1 and subnet 2, respectively.

```
Qtech(config)#ipv6 access-list net1
Qtech(config-ipv6-acl)#permit ipv6 2003::/64 any
Qtech(config)#ipv6 access-list net2
Qtech(config-ipv6-acl)#permit ipv6 2004::/64 any
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer. The default parameter is included in attributes of the IPv4 next hop.

```
Qtech(config)#route-map RM_FOR_PBR 10
Qtech(config-route-map)#match ip address 101
Qtech(config-route-map)#set ip default next-hop 59.78.184.1
Qtech(config-route-map)#set ip default next-hop 210.82.12.1

Qtech(config-route-map)#match ipv6 address net1
Qtech(config-route-map)#set ipv6 next-hop 2001::1
Qtech(config-route-map)#set ipv6 next-hop 2002::1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer. The default parameter is included in attributes of the IPv4 next hop.

```
Qtech(config)#route-map RM_FOR_PBR 20
Qtech(config-route-map)#match ip address 102
Qtech(config-route-map)#set ip default next-hop 210.82.12.1
Qtech(config-route-map)#set ip default next-hop 59.78.184.1

Qtech(config)#route-map RM_FOR_PBR 20
Qtech(config-route-map)#match ipv6 address net2
Qtech(config-route-map)#set ipv6 next-hop 2002::1
Qtech(config-route-map)#set ipv6 next-hop 2001::1
```

# Configure redundant backup.

```
Qtech(config)#ipv6 policy redundance
```

# Apply the IPv4/IPv6 policy-based routing on the interface GigabitEthernet 0/3.

```
Qtech(config)#interface GigabitEthernet 0/3
Qtech(config-if-GigabitEthernet 0/3)#ip policy route-map RM_FOR_PBR
Qtech(config-if-GigabitEthernet 0/3)#ipv6 policy route-map RM_FOR_PBR
```

2.4.5.5 Verification

# Show the configuration of the route map.

```
Qtech#show route-map
route-map RM_FOR_PBR, permit, sequence 10
  Match clauses:
    ip address 101
    ipv6 address net1
  Set clauses:
    ipv6 next-hop 2001::1 2002::1
    ip default next-hop 59.78.184.1 210.82.12.1
route-map RM_FOR_PBR, permit, sequence 20
  Match clauses:
    ip address  102
    ipv6 address net2
  Set clauses:
    ipv6 next-hop 2002::1 2001::1
    ip default next-hop 210.82.12.1 59.78.184.1
```

# Show the application of IPv6 policy-based routing.

```
Qtech#show ipv6 policy
Interface                               Route map
GigabitEthernet 0/3                    RM_FOR_PBR
```

# Show the application of IPv4 policy-based routing.

```
Qtech#show ip policy
Interface                               Route map
GigabitEthernet 0/3                    RM_FOR_PBR
```

# Show the configuration of ACLs.

```
Qtech#show access-lists
Extended IP access list 101
    10 permit ip 202.112.144.0 0.0.0.255 any
Extended IP access list 102
    10 permit ip 218.62.95.0 0.0.0.255 any
IPv6 access list net1
    permit ipv6 2003::/64 any sequence 10
IPv6 access list net2
    permit ipv6 2004::/64 any sequence 10
```

2.4.6 Example 6: VRF Election Using IPv6 PBR

2.4.6.1 Networking Requirements

A provider edge (PE) requires applying policy-based routing to the packets received from FastEthernet 0/1. It routes the IP packets from subnet 1 by VRF1, the IP packets from subnet 2 by VRF2, and the IP packets from subnet 3 by VRF3. Other packets are routed in the public network.

2.4.6.2 Configuration Steps

# Create a VRF instance.

```
Qtech(config)#vrf definition VRF1
Qtech(config-vrf)#address-family ipv6
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#vrf definition VRF2
Qtech(config-vrf)#address-family ipv6
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#vrf definition VRF3
Qtech(config-vrf)#address-family ipv6
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#exit
```

# Configure an ACL as the matching rule of the routing map.

```
Qtech(config)#ipv6 access-list acl1
Qtech(config-ipv6-acl)#permit ipv6 1000::/64 any
Qtech(config-ipv6-acl)#ipv6 access-list acl2
Qtech(config-ipv6-acl)#permit ipv6 2000::/64 any
Qtech(config-ipv6-acl)#ipv6 access-list acl3
Qtech(config-ipv6-acl)#permit ipv6 3000::/64 any
```

# Configure the routing map.

```
Qtech(config-ipv6-acl)#route-map PBR-VRF-Selection permit 10
Qtech(config-route-map)#match ipv6 address acl1
Qtech(config-route-map)#set vrf VRF1

Qtech(config)#route-map PBR-VRF-Selection permit 20
Qtech(config-route-map)#match ipv6 address acl2
Qtech(config-route-map)#set vrf VRF2

Qtech(config)#route-map PBR-VRF-Selection permit 30
Qtech(config-route-map)#match ipv6 address acl3
Qtech(config-route-map)#set vrf VRF3
```

# Import IPv6 address of the interface to VRFs 1 to 3.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ipv6 address 1000::1/64
Qtech(config-if)#vrf receive VRF1
Qtech(config-if)#vrf receive VRF2
Qtech(config-if)#vrf receive VRF3
```

# Apply the IPv6 PBR to the interface.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if)#ipv6 policy route-map PBR-VRF-Selection
```

3 CONFIGURING RIP

3.1 Overview

The Routing Information Protocol (RIP) is a relatively old routing protocol, which is widely used in small or homogeneous networks. The RIP uses the distance-vector algorithm, and so is a distance-vector protocol. The RIPv1 is defined in RFC 1058 and the RIPv2 is defined in RFC 2453. Qtech RGOS supports both two versions.

The RIP exchanges the routing information by using UDP packets with UDP port number 520. Usually, RIPv1 packets are broadcast packets, while RIPv2 packets are multicast packets with the multicast address of 224.0.0.9. The RIP sends an update packet at the interval of 30 seconds. If a device fails to receive the route update packets from the peer within 180 seconds, it will mark all the routes from the device unreachable. After that, the device will delete these routes from its routing table if it still fails to receive any update packets within 120s.

The RIP measures the distance to the destination in hop, known as route metric. In the RIP, zero hop exists when the device directly connects to the network. One hop exists when the destination is reachable through one device and so on. If the destination is unreachable, the hop count is 16.

The RIP-enabled device can learn the default routes from the neighbors or generate its own default route. When any of the following conditions is met, Qtech products will introduce the default route and advertise it to its neighbor devices by using the **default-information originate** command:

- IP Default-network is configured.
- Other RIPs learn the default routes or are configured with static default routes.

The RIP will send update packets to a specified network interface. If the network is not associated with the RIP routing progress, the interface will not advertise any update packets. The RIP is available in two versions: RIPv1 and RIPv2. The RIPv2 supports plain-text authentication, MD5 cryptographic text authentication, and variable length subnet masks.

Qtech RIP offers Split Horizon to avoid a loop.

3.2 Configuring RIP

The RIP configuration task list contains:

- Create the RIP routing process (mandatory).
- Configure RIP packets in unicast mode (optional).
- Configure Split Horizon (optional).
- Define the RIP Version (optional).
- Configure the Route Aggregation function(optional).
- Configure RIP Authentication.
- Adjust the RIP clock (optional).
- Configure the RIP Route Source Address Validation (optional).
- Control RIP interface status (optional).
- Advertise the default route through the RIP interface (optional).
- Advertise the supernet route through the RIP interface (optional).
- Configure RIP VRF (optional).
- Configure RIP BFD (optional).
- Configure RIP triggered expansion (optional).
- Configure RIP graceful restart (GR) (optional).

For the following topics, see the "IP Routing Protocol Independent Feature Configuration" chapter.

- Filter RIP route information
- Redistribute routes
- Configure default route distribution

The following table describes the default configuration of RIP.

Feature	Default Setting

Feature	Default Setting
Network interface	After an interface joins the RIP, it receives RIPv1 and RIPv2 packets and advertises RIPv1 packets by default. By default, when advertising RIPv1 packets: ■ The interface sends the packets in broadcast mode. ■ The interface does not advertise supernet routes. By default, when advertising RIPv2 packets: ■ The interface sends the packets in multicast mode. ■ The interface automatically converges routes into classified routes. ■ The interface advertises supernet routes. The interface enables split horizon.
RIP neighbor	Undefined
Verification of the source IP address of a packet	Enabled
Timer	By default: ■ The update time is 30 seconds. ■ The expiry time is 180 seconds. ■ The clearing time is 120 seconds.
Offset list	Undefined
Automatic convergence	Enabled
Redistribution	By default, routing redistribution is disabled. If it is enabled: ■ Redistribute OSPF, that is, redistribute all sub-type routes of this instance. ■ Redistribute ISIS, that is, redistribute level-2 sub-type routes of this instance. ■ In other cases, redistribute all routes of this type. ■ The metric value for route distribution is the default one.
Default route distribution	By default, default route distribution is disabled. If it is enabled: the metric value for route distribution is the default one.
Default metric	Redistribute metric values used by routes of other protocols. The default value is 1.
Administrative distance	120
RIP triggered expansion	By default, RIP triggered expansion is disabled. If it is enabled: The default interval is 5 seconds for retransmitting update request and update response packets. By default, a maximum of 36 times are allowed for retransmitting update request and update response packets.
RIP GR	By default, RIP GR is disabled. The default RIP GR period is the smaller value between twice the update time and 60 seconds.

3.2.1 Creating the RIP Routing Process

To run RIP for a device, create the RIP routing process and define networks associated with the RIP routing process.

Use the following commands to create the RIP routing process in global configuration mode.

Command	Function
Qtech(config)# **router rip**	Creates the RIP routing process.
Qtech(config-router)# **network** *network-number wildcard*	Defines associated networks.

You can configure the *network-number* and *wildcard* parameters at the same time to enable the network segments of the interface IP address within the IP address range to run RIP.

If the *wildcard* parameter is not configured, by default, RGOS will enable the network segments of the interface IP address within the classified IP address range to run RIP.

Note There are two meanings for associated networks defined by the network command:

Note

The RIP only advertises the route information of associated networks.

Note

The RIP only advertises and receives route update messages through the interfaces of associated networks.

3.2.2 Configuring RIP Packets in Unicast Mode

The RIP is usually a broadcast or multicast protocol. If the RIP route information needs to be transmitted through non-broadcast networks, a device needs to be configured to support that the RIP advertises route information update packets in unicast mode.

Use the following commands to advertise RIP information update messages in unicast mode in RIP routing process configuration mode.

Command	Function
Qtech(conf-router)# neighbor ip-address	Configures RIP packet advertising in unicast mode.

This command enables you to control an interface about whether to advertise RIP route update packets and forbid advertising route update packets in broadcast mode through an interface. You need to configure the **passive-interface** command in routing progress configuration mode. For related descriptions on the restriction of route message advertisements, see the "Route Filtering Configuration" of *Configuring Protocol Independent*.

Note

During the configuration of FR and X.25, if the broadcast keyword is specified for IP address mapping, the neighbor command is not required because the command is mainly used for reducing broadcast packets and filtering routes.

3.2.3 Configuring Split Horizon

Split horizon can be used to avoid loop when multiple devices running distance-vectortype routing protocols connect to a network in which IP packets are broadcasted. Split horizon can prevent devices from advertising certain route information through an interface from which the devices learn such information. This optimizes route information exchange among multiple devices.

However, split horizon may cause the failure of some device to learn all the route information in a non-broadcast multi-access network (frame relay or X.25). In this case, you may need to disable split horizon. If an interface is configured with the secondary IP address, you need to pay attention to split horizon.

If the **poisoned-reverse** parameter is configured, split horizon with poisoned reverse is enabled. The device will advertise the route information from the interface where it learns the information, and configure the metric value of the route information as unreachable.

Use the following commands to enable or disable split horizon in interface configuration mode.

Command	Function
Qtech(config-if)# **no ip split-horizon**	Disables split horizon.
Qtech(config-if)# **ip split-horizon**	Enables split horizon.

Use the following commands to enable or disable split horizon with poisoned reverse in interface configuration mode.

Command	Function
Qtech(config-if)# **no ip rip split-horizon poisoned-reverse**	Disable split horizon with poisoned reverse.
Qtech(config-if)# **ip rip split-horizon poisoned-reverse**	Enable split horizon with poisoned reverse.

By default, all interfaces are configured as enabling split horizon without poisoned reverse.

3.2.4 Defining the RIP Version

Qtech products support RIP version 1 and version 2, where RIPv2 supports authentication, key management, route convergence, CIDR, and VLSMs. For the information about key management and VLSMs, see the "*IP Routing Protocol Independent Feature Configuration*" chapter.

By default, Qtech products can receive RIPv1 and RIPv2 packets, but they can send only RIPv1 packets. You can configure them to receive and send only RIPv1 or RIPv2 packets.

Use the following command to enable software to receive and send only the packets of a specific version in routing process configuration mode.

Command	Function
Qtech(config-router)# **version** {**1** | **2**}	Defines the RIP version.

The above command allows the software to receive or send only the packets of a specific version by default. If necessary, you can modify the default setting of each interface.

Use the following commands to enable an interface to send only the packets of a specific version in interface configuration mode.

Command	Function
Qtech(config-if)# **ip rip send version 1**	Specifies to send only RIPv1 packets.
Qtech(config-if)# **ip rip send version 2**	Specifies to send only RIPv2 packets.
Qtech(config-if)# **ip rip send version 1 2**	Specifies to send only RIPv1 and RIPv2 packets.

Use the following commands to configure an interface to receive only the packets of a specific version in interface configuration mode.

Command	Function
Qtech(config-if)# **ip rip receive version 1**	Specifies to receive only RIPv1 packets.
Qtech(config-if)# **ip rip receive version 2**	Specifies to receive only RIPv2 packets.
Qtech(config-if)# **ip rip receive version 1 2**	Specifies to receive only RIPv1 and RIPv2 packets.

3.2.5 Configuring Route Convergence

Automatic RIP route convergence means that the routes of subnets are automatically converged into the routes of a classful network when they pass through the border of the classful network. By default, RIPv2 will automatically perform route convergence, while the RIPv1 does not support this function.

The automatic route convergence function of the RIPv2 improves the scalability and effectiveness of the network. If there are any converged routes, the sub-routes contained in them cannot be seen in the routing table. This greatly reduces the size of the routing table.

It is more efficient to advertise converged routes than the separated routes. Factors are as follows:

- Converged routes will be handled first when you search the RIP database.
- Any sub-routes will be ignored when you search the RIP database, and thus reducing the handling time.

Sometimes, you want to learn the specific sub-net routes rather than the converged network routes. In this case, you need to disable the automatic route convergence function.

Use the following commands to configure automatic route convergence in RIP routing progress mode.

Command	Function
Qtech(config-router)# **no auto-summary**	Disables automatic route convergence.
Qtech(config-router)# **auto-summary**	Enables automatic route convergence.

Use the following commands to configure interface-level convergence in interface mode. Then, configure route convergence within the specified classified subnet range on an interface.

Command	Function
Qtech(config-if)# **ip summary-address** *rip ip-address ip-network-mask*	Enables route convergence on the interface.
Qtech(config-if)# **no ip summary-address** *rip ip-address ip-network-mask*	Disables route convergence on the interface.

3.2.6 Configuring RIP Authentication

RIPv1 does not support authentication. If a device is configured with the RIPv2, you can configure authentication on an appropriate interface.

RIPv2 for Qtech products supports two RIP authentication modes: plain-text authentication and MD5 authentication. The default authentication mode is plain-text authentication.

In plain-text authentication mode, you can run the **ip rip authentication text-password** command to configure the plain-text authentication password or obtain the plain-text authentication password through an associated key chain. The latter takes precedence over the former.

In MD5 authenticaiton mode, you must implement MD5 authentication through an associated key chain.

For plain-text authentication, no authentication occurs if no plain-text authentication password or associated key chain is configured. Similarly, for MD5 authentication, no authentication occurs if no associated key chain is configured.

If a key chain is specified in interface configuration mode, you need to use the **key chain** command in global configuration mode to define the key chain. Otherwise, authentication of RIP data packets may fail.

Use the following commands to configure RIP authentication in interface configuration mode.

Command	Function
Qtech(config-if)# **ip rip authentication mode** {**text** | **md5**}	Uses the key chain, enables RIP authentication, and configures RIP authentication through the interface. text: indicates plain-text authentication. md5: indicates MD5 authentication.
Qtech(config-if)# **ip rip authentication text-password** [**0**|**7**] *password-string*	Configures the plain-text authentication password in the length of 1 – 16 bytes. 0 Displays a key in plain text manner. 7 Displays a key in encrypted manner.
Qtech(config-if)# **ip rip authentication key-chain** *key-chain-name*	Configures authentication using a key chain.

3.2.7 Configuring RIP Clock Adjustment

The RIP provides the clock adjustment function, which allows you to adjust a clock based on network conditions so that the RIP can run in a better way. You can adjust the following clocks:

Route update time: It defines the period in seconds for a device to send route update packets;

Route expiry time: It defines the time in seconds after which the routes in the routing table will become invalid if not updated;

Route clearing timer: It defines the time in seconds after which the routes in the routing table will be cleared;

By adjusting above clocks, the convergence and fault recovery of the routing protocol may be accelerated. Use the following command to adjust an RIP clock in RIP routing process configuration mode.

Command	Function
Qtech(config-router)# **timers basic** *update invalid flush*	Adjusts the RIP clock.

By default, the update time is 30 seconds, the expiry time is 180 seconds, and the clearing time is 120 seconds.

Note For devices connected on the same network, the values of the RIP clocks must be the same.

3.2.8 Configuring Verification of the Source IP Address of an RIP Route

By default, the RIP will verify the source IP address of a received route update packet. The RIP will discard the packet if the source IP address is invalid. Judging whether the source IP address is valid, that is, judging whether the source IP address is in the same network as the IP address of the interface. No validation authentication will be performed on the interface of no numbered IP address.

Use the following commands to configure verification of route source IP address in RIP routing process configuration mode.

Command	Function
Qtech(config-router)# **no validate-update-source**	Disables the source IP address validation.
Qtech(config-router)# **validate-update-source**	Enables the source IP address validation.

3.2.9 Configuring Control of the RIP Interface Status

In some case, it is necessary to configure the RIP flexibly. If you only need to enable a device to learn RIP routes rather than advertising RIP routes, you can configure a passive interface. Or, if you need to configure the status of a certain interface individually, you can use a command to control the sending or receiving of the RIP packets on a specific interface.

Use the following commands to configure an interface as the passive interface in RIP routing process configuration mode.

Command	Function
Qtech(config-router)# **passive-interface** {**default** | *interface-type interface-num*}	Configures a passive interface.
Qtech(config-router)#**no passive-interface** {**default** | *interface-type interface-num*}	Cancels the configuration.

Caution

A passive interface responds the non-RIP requests (such as the route diagnosis program) rather than the RIP requests because these request programs hope to learn about the routes of all devices.

Use the following commands to disable or allow an interface to receive RIP packets in interface configuration mode.

Command	Function
Qtech(config-if)# **no ip rip receive enable**	Forbids the interface to receive the RIP packets.
Qtech(config-if)# **ip rip receive enable**	Allows the interface to receive the RIP packets.

Use the following commands to disable or allow an interface to send RIP packets in interface configuration mode.

Command	Function
Qtech(config-if)# **no ip rip send enable**	Forbids the interface to send the RIP packets.
Qtech(config-if)# **ip rip send enable**	Allows the interface to send the RIP packets.

3.2.10 Configuring Default Route Advertisement through an Interface

Use the following command to generate a default route (0.0.0.0/0) in the update packet through a specified interface in interface configuration mode.

Command	Function
Qtech(config-if)# **ip rip default-information originate** [**metric** *metric-value*]	Advertises the default route and other routes.
Qtech(config-if)# **no ip rip default-information**	Cancels default route advertising through the interface.

In interface configuration mode, use the following commands to generate a default route (0.0.0.0/0) in the update route through a specified interface, and advertise only this default route instead of other RIP routes through this interface.

Command	Function
Qtech(config-if)# **ip rip default-information only** [**metric** *metric-value*]	Advertises the default route only.
Qtech(config-if)# **no ip rip default-information**	Cancels default route advertising through the interface.

If both the **ip rip default-information** command on the interface and the **default-information originate** command in the RIP progress are configured, only the default route configured on the interface is advertised.

3.2.11 Configuring Supernet Route Advertisement through the RIP Interface

A supernet route (for example, 80.0.0.0/6) is defined when the mask length is less than its natural mask length. According to IP address classification, 80.0.0.0 belongs to class-A network and its natural mask legnth is 8. Therefore, 80.0.0.0/6 is a supernet route.

When an RIPv1-enabled device monitors RIPv2 route response packets, it will learn incorrect routes because RIPv1 ignores the subnet masks of the routes in the packets if information about the supernet routes is received. In this case, an RIPv2-enabled device needs to disable advertising super network route on its interface.

Use the following command to configure whether to advertise the supernet route through an interfacein interface configuration mode.

Command	Function
Qtech(config-if)#**no ip rip send supernet-routes**	Disables advertising the supernet route through the interface.
Qtech(config-if)# **ip rip send supernet-routes**	Enables advertising the supernetwork route through the interface.

Note

1. When only RIPv1 packets rather than RIPv2 packets are received through the interface, no supernet route is received.
2. Supernet routes can be received when RIPv2 packets are allowed to be received through the interface.
3. No supernet route is sent when RIPv1 packets are sent through the interface.
4. Supernet routes are permitted to be sent by default when RIPv2 packets are sent through the interface.
5. The **no rip rip send supernet-routes** command prohibits sending supernet routes.
The **auto-summary** command takes no effect for supernet routes, that is, supernet routes are not converged.
The **ip rip summary** command does not support configuration of supernet routes.

3.2.12 Configuring RIP VRF

The RIP supports VRFs. Multiple RIP instances can be created to manage the corresponding VRFs in the RIP process. By default, there is only one RIP instance in the RIP process, which is used to manage the global routing table. After a VRF is created, you can manage the routing table of the VRF by creating a new RIP instance.

Run the **address-family** command to enable a router device to enter the address family configuration mode (with the prompt (config-router-af)#). When you specify the VRF associated with the sub mode at the first time, the RIP will create a RIP instance corresponding to the VRF. Under this mode, you can configure the RIP instance of the VRF in the same way as that in global route configuration mode.

To exit the address family configuration sub mode and return to the route configuration mode, run the **exit-address-family** command.

Use the following commands to configure a RIP instance managng the VRF in RIP routing process configuration mode.

Command	Function
Qtech(config-router)# **address-family ipv4 vrf** *vrf-name*	Creates the RIP instance managing the VRF.
Qtech(config-router)#**no address-family ipv4 vrf** *vrf-name*	Removes the RIP instance managing the VRF.

3.2.13 Configuring RIP BFD

For details on RIP BFD configuration, see *BFD Configuration Guide*.

3.2.14 Configuring TRIP

Triggered RIP (TRIP) is a RIP extension on a wide area network(WAN), and is mainly used on the on-demand link.

When TRIP is enabled, RIP protocol will no longer periodically send route updates but only send route updates to WAN interfaces in the following cases:

- When route update request packets are received.
- When RIP routing information has changed.
- When interface state changes.
- When routers start.

Since the periodic RIP update is canceled, an acknowledgement and retransmission mechanism is required to guarantee successful update packet transmission and receiving on the WAN. RIP uses three new types of packets which are identified by the value of the command field in the RIP header:

- Update request (Type-9): requests the peer to send the routing information needed.
- Update response (Type-10): contains the route updates requested by the peer.
- Update Acknowledge (Type-11): acknowledges the received update responses, indicating that the route updates sent by peer have been received.

Caution

1. This function can be used in the following cases: (1) The interface is connected to only one neighbor; (2) The interface is connected to multiple neighbors using unicast communication mode. You are advised to enable this feature on PPP, frame relay, X.25, and similar link layer protocols.
2. You are advised to enable split horizon with poisoned reverse on TRIP-enabled interface. Otherwise, there may be residual invalid routing information.
3. It shall be guaranteed that the feature is enabled on all routers on the same link. Otherwise, the function may fail and routing information cannot be exchanged properly.
4. This function cannot be used together with BFD for RIP;
5. When this function is enabled, make sure that the RIP configurations on both ends of the link are identical, such as RIP authentication and version of RIP protocol supported by the interface and etc.
6. With this function enabled on the interface, the valid-update-source will be performed for the packets of this interface no matter whether the valid-update-source function is enabled.

Use the following commands to enable or disable this function in interface configuration mode.

Command	Function
Qtech(config-if)# **ip rip triggered**	Enables Triggered RIP.
Qtech(config-if)# **no ip rip triggered**	Disables Triggered RIP.

3.2.15 Configuring RIP GR

RIP graceful restart (GR) guarantees non-stop data forwarding during the process of protocol restart. When RIP GR is enabled on the router, the forwarding table will be maintained during the process of RIP restart, and request packets will be sent to neighbors to re-learn routes in order to complete route re-convergence within the period of graceful restart. Upon expiration of the GR period, GR will exit and forwarding table entries will be updated and advertised to neighbors.

The GR period is the maximum duration from RIP GR execution to RIP GR completion. During this period, the forwarding table will be maintained and RIP route recovery will be implemented in order to restore RIP to the state before GR. Upon expiration of grace period, RIP will exit from the GR state and perform common RIP operations.

graceful-restart grace-period allows users to explicitly change the restart period. Please note that GR must be completed within the RIP expiration time and one RIP route update cycle is completed. If this value is not properly configured, non-stop data forwarding cannot be guaranteed during the GR process. For example, if the GR period is longer than the expiration time of neighbor routers and GR is not completed within such expiration time, the neighbor's routes will not be sent upon expiration of the expiration time, thus causing interruption of data forwarding. Therefore, unless otherwise specified, it is not allowed to adjust the GR period. If the GR period is adjusted, please refer to the configuration of the **timers basic** command and make sure the GR period is longer than the update time and smaller than the expiration time.

Use the following commands to enable or disable this function in RIP routing process configuration mode.

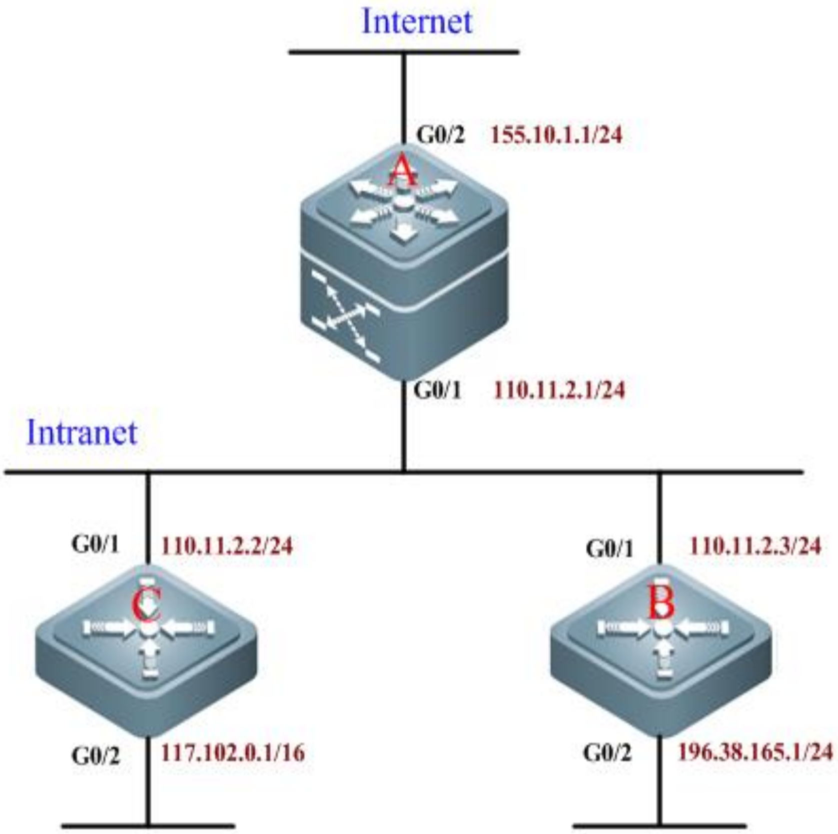
Command	Function
Qtech(config-router)# **graceful-restart** [**grace-period** *grace-period*]	Enables RIP GR.
Qtech(config-router)# **no graceful-restart** [**grace-period**]	Disables RIP GR.

3.3 RIP Configuration Examples

3.3.1 Configuring RIP Routes and Defining RIP Versions

3.3.1.1 Networking Topology

Figure 1 Configuring RIP routes and defining RIP versions



3.3.1.2 Networking Requirements

A small-sized company runs on a small office network, and requires network layer intercommunication between any two nodes. Networking requirements are as follows:

- Devices shall be able to adapt to the changes in the network topology, in order to reduce the workload of manual maintenance;
- Route updates can carry subnet masks;
- Device A only receives the routing information from external networks, but will not advertise routing information of internal network.
- RIP information can be exchanged between devices A, B, and C, so that internal hosts can access Internet.

3.3.1.3 Configuration Tips

- According to user's requirements and network environment, the RIPv2 routing protocol is selected to achieve user network intercommunication;
- To allow device A to receive routing information sent from external network without advertising the routing information of internal network, the G0/2 port of device A shall be configured as a passive interface.

3.3.1.4 Configuration Steps

# Configure device A

! Configure the IP address of the corresponding port on device A.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 110.11.2.1 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface gigabitEthernet 0/2
Qtech(config-if)#ip address 155.10.1.1 255.255.255.0
```

! Create the RIP routing progress.

```
Qtech(config)#router rip
```

! Configure RIP version as version 2.

```
Qtech(config-router)#version 2
```

! Configure G0/2 as a passive interface.

```
Qtech(config-router)#passive-interface gigabitEthernet 0/2
```

! Disable automatic route convergence.

```
Qtech(config-router)#no auto-summary
```

! Specify the associated network.

```
Qtech(config-router)#network 110.11.2.0 255.255.255.0
Qtech(config-router)#network 155.10.1.0
```

# Configure device B

! Configure the IP address of the corresponding port on device B.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 110.11.2.2 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface gigabitEthernet 0/2
Qtech(config-if)#ip address 196.38.165.1 255.255.255.0
Qtech(config-if)#exit
```

! Create RIP routing progress.

```
Qtech(config)#router rip
```

! Configure the RIP version as version 2.

```
Qtech(config-router)#version 2
```

! Disable automatic route convergence.

```
Qtech(config-router)#no auto-summary
```

! Specify the associated network.

```
Qtech(config-router)#network 110.11.2.0
Qtech(config-router)#network 196.38.165.0
```

# Configure device C

! Configure the IP address of the corresponding port on device C.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 110.11.2.3 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface gigabitEthernet 0/2
Qtech(config-if)#ip address 117.102.0.1 255.255.0.0
Qtech(config-if)#exit
```

! Create RIP routing progress.

```
Qtech(config)#router rip
```

! Configure RIP version as version 2.

```
Qtech(config-router)#version 2
```

! Disable automatic route convergence.

```
Qtech(config-router)#no auto-summary
```

! Specify the associated network.

```
Qtech(config-router)#network 110.11.2.0
Qtech(config-router)#network 117.102.0.0
```

3.3.1.5 Verification

View the routing table of each device;

View the routing table on A, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.1/32 is local host.
R 117.102.0.0/16 [120/1] via 110.11.2.2, 00:00:47, GigabitEthernet 0/1
C    155.10.1.0/24 is directly connected, GigabitEthernet 0/2
C    155.10.1.1/32 is local host.
C    192.168.217.0/24 is directly connected, VLAN 1
C    192.168.217.233/32 is local host.
R  196.38.165.0/24 [120/1] via 110.11.2.3, 00:19:18, GigabitEthernet 0/1
```

View the routing table on B, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.2/32 is local host.
R 155.10.1.0/24 [120/1] via 110.11.2.1, 00:15:21, GigabitEthernet 0/1
C    196.38.165.0/24 is directly connected, GigabitEthernet 0/2
C    196.38.165.1/32 is local host.
R 117.102.0.0/16 [120/1] via 110.11.2.2, 00:00:47, GigabitEthernet 0/1
```

View the routing table on C, as shown below (the bold figures are the routing information learned through RIP):

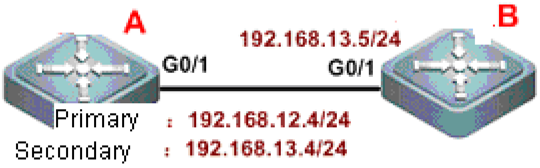
```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
```

```
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.3/32 is local host.
C    117.102.0.0/16 is directly connected, GigabitEthernet 0/2
C    117.102.0.1/32 is local host.
R  155.10.1.0/24 [120/1] via 110.11.2.1, 00:20:55, GigabitEthernet 0/1
R  196.38.165.0/24 [120/1] via 110.11.2.3, 00:19:18, GigabitEthernet 0/1
```

3.3.2 RIP Split Horizon

3.3.2.1 Networking Topology

Figure 2 Topology for RIP split horizon



3.3.2.2 Networking Requirements

There are two devices on the network. Device A is configured with a secondary IP address.

The following requirements shall be met:

- The RIP routing protocol runs on both devices;
- Device B can learn the routes of network segment 192.168.12.0/24.

3.3.2.3 Configuration Tips

To meet the above requirements, the following configurations are required:

- RIPv2 routing protocol is run on both devices;
- Split horizon shall be disabled on device A (by default, split horizon is enabled on all interfaces), or else device A won't advertise network segment 192.168.12.0 to device B.

3.3.2.4 Configuration Steps

# Configure device A

! Configure Ethernet ports.

```
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)#ip address 192.168.12.4 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)#ip address 192.168.13.4 255.255.255.0 secondary
```

! Disable split horizon.

```
Qtech(config-if-GigabitEthernet 0/1)#no ip rip split-horizon
```

! Configure the RIP routing protocol.

```
Qtech(config)#route rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.12.0
Qtech(config-router)#network 192.168.13.0
```

! Disable automatic route convergence.

```
Qtech(config-router)#no auto-summary
```

# Configure device B

! Configure Ethernet ports.

```
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)#ip address 192.168.13.5 255.255.255.0
```

! Configure the RIP routing protocol.

```
Qtech(config)#route rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.13.0
```

3.3.2.5 Verification

View the routing table on device B before and after disabling split horizon.

Before split horizon is disabled, view the routing table on device B, as shown below:

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    192.168.13.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.13.5/32 is local host.
```

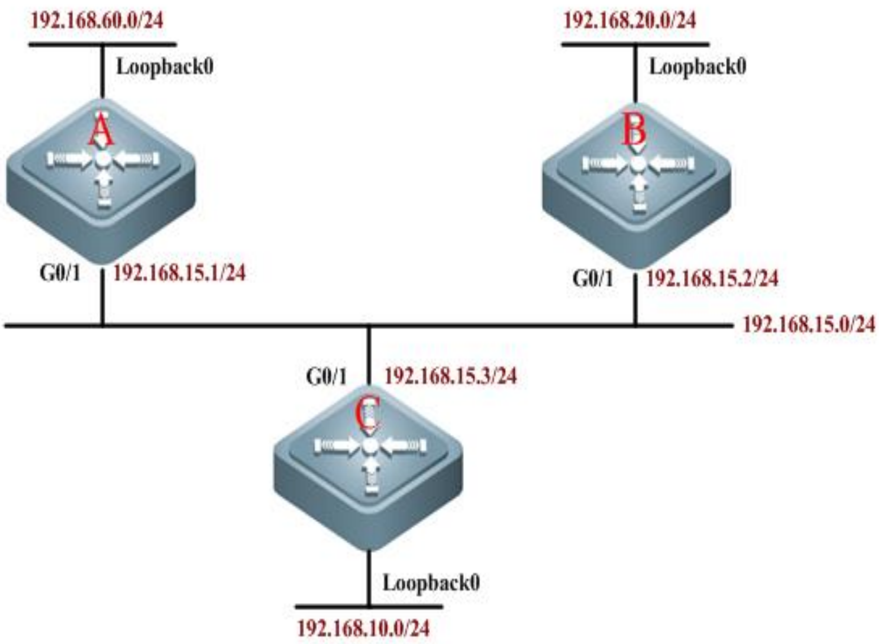
After split horizon is disabled, view the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.12.0/24 [120/1] via 192.168.13.4, 00:00:10, GigabitEthernet 0/1
C    192.168.13.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.13.5/32 is local host.
```

3.3.3 RIP Unicast Update

3.3.3.1 Networking Topology

Figure 3 Topology for RIP unicast update



3.3.3.2 Networking Requirements

As shown below, three devices are connected to the LAN and run the RIP routing protocol.

- Device A can learn the routes advertised by devices B and C;
- Device C can learn the routes advertised by devices A and B;
- Device B cannot learn the routes advertised by device C.

3.3.3.3 Configuration Tips

To meet the above configuration requirements, RIP unicast packets must be configured on device C. Add the command of **neighbor** during the RIP configuration of device C, so that the RIP protocol can send advertisements to the interface of device A in unicast mode. Configure the **passive-interface** command on G0/1 of device C to avoid broadcast update on this link.

3.3.3.4 Configuration Steps

# Configure device A

! Configure the IP address of corresponding interface.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 192.168.15.1 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface Loopback 0
Qtech(config-if)#ip address 192.168.60.1 255.255.255.0
Qtech(config-if)#exit
```

! Create RIP routing progress.

```
Qtech(config)#router rip
```

! Specify the associated network.

```
Qtech(config-router)#network 192.168.60.0
Qtech(config-router)#network 192.168.15.0
```

# Configure device B

! Configure the IP address of corresponding interface.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 192.168.15.2 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface Loopback 0
Qtech(config-if)#ip address 192.168.20.1 255.255.255.0
Qtech(config-if)#exit
```

! Create RIP routing progress.

```
Qtech(config)#router rip
```

! Specify the associated network.

```
Qtech(config-router)#network 192.168.20.0
Qtech(config-router)#network 192.168.15.0
```

# Configure device C

! Configure the IP address of corresponding interface.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 192.168.15.3 255.255.255.0
Qtech(config-if)#exit
Qtech(config)#interface Loopback 0
Qtech(config-if)#ip address 192.168.10.1 255.255.255.0
Qtech(config-if)#exit
```

! Create RIP routing progress.

```
Qtech(config)#router rip
```

! Specify the associated network.

```
Qtech(config-router)#network 192.168.15.0
Qtech(config-router)#network 192.168.10.0
```

! Configure G0/1 as a passive interface.

```
Qtech(config-router)#passive-interface gigabitEthernet 0/1
```

! Enable unicast update.

```
Qtech(config-router)#neighbor 192.168.15.1
```

3.3.3.5 Verification

View the routing table of each device (mainly the routing information on devices C and B):

View the routing table on device B, as shown in the following figure:

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.20.0/24 is directly connected, Loopback 0
C    192.168.20.1/32 is local host.
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/1
```

```
C    192.168.15.2/32 is local host.
R 192.168.60.0/24 [120/1] via 192.168.15.1, 00:15:21, GigabitEthernet 0/1
```

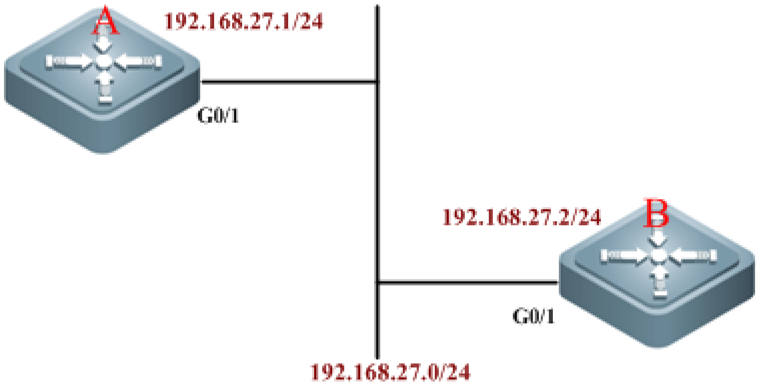
View the routing table on device C, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.10.0 is directly connected, Loopback 0
C    192.168.10.1/32 is local host.
R 192.168.60.0/24 [120/1] via 192.168.15.1, 00:15:21, GigabitEthernet 0/1
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.15.3/32 is local host.
R 192.168.20.0 [120/1] via 192.168.15.2, 00:00:47, GigabitEthernet 0/1
```

3.3.4 RIP Authentication

3.3.4.1 Networking Topology

Figure 4 Topology for RIP authentication



3.3.4.2 Networking Requirements

Interconnected through Ethernet, two devices run the RIP routing protocol and use MD5 authentication. The requirements are as follows:

- The authentication key for device A to send RIP packets is "Hello", and device A can receive RIP packets with authentication keys being "Hello" and "World";
- The authentication key for device B to send RIP packets is "World", and device B can receive RIP packets with authentication keys being "Hello" and "World";
- The first key is used from 4:30pm October 1st, 2010 for 12 hours (43200s)
- The second key becomes permanently valid from 4:00 am October 2, 2010.

3.3.4.3 Configuration Tips

Authentication is not supported in RIPv1. If the RIPv2 routing protocol is configured on the device, authentication can then be configured on the corresponding interface.

The key string specifies the key set that can be used by this interface. If the key string is not configured and even if the interface uses the key chain, no authentication occurs. Therefore, before configuring authentication, the key chain and the associated key string must be configured first.

RGOS supports two RIP authentication modes: plain text and MD5, while plain text is the default authentication mode.

- The authentication key for sending RIP packets must be configured with the first key on keychain;
- When configuring the authentication key that can be received, configure any key on the keychain.

3.3.4.4 Configuration Steps

# Configure device A:

! Configure the IP address of Ethernet interface.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 192.168.27.1 255.255.255.0
Qtech(config-if)#exit
```

! Configure the key chain named "ripchain".

```
Qtech(config)#key chain ripchain
```

! Configure the first key of "Key 1", which contains the key-string of "Hello", and configure the corresponding period needed.

```
Qtech(config-keychain)#key 1
Qtech(config-keychain-key)#key-string Hello
Qtech(config-keychain-key)#accept-lifetime 16:30:00 Oct 1 2010 duration 43200
Qtech(config-keychain-key)#send-lifetime 16:30:00 Oct 1 2010 duration 43200
Qtech(config-keychain-key)#exit
```

! Configure the second key of "Key 2", which contains the key-string of "World", and configure the corresponding period needed.

```
Qtech(config-keychain)#key 2
Qtech(config-keychain-key)#key-string World
Qtech(config-keychain-key)#accept-lifetime 04:00:00 Oct 2 2010 infinite    //Beginning
time that the key is valid to be received
Qtech(config-keychain-key)#send-lifetime 04:00:00 Oct 2 2010 infinite
//Beginning time that the key is valid to be sent
Qtech(config-keychain-key)#end
```

! Configure G0/1 to use the MD5 authentication key to authenticate the update messages sent from device B.

```
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip rip authentication key-chain ripchain
Qtech(config-if)#ip rip authentication mode md5
Qtech(config-if)#exit
```

! Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.27.0
```

#Configure device B:

! Configure the IP address of Ethernet interface.

```
Qtech>enable
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip address 192.168.27.2 255.255.255.0
Qtech(config-if)#exit
```

! Configure the key chain.

```
Qtech(config)#key chain ripchain //The name of key chain is only valid on the local
device. You can also use other names.
```

! Configure the first key of "Key 1", which contains the key-string of "Hello", and configure the corresponding period needed.

```
Qtech(config-keychain)#key 1
Qtech(config-keychain-key)#key-string Hello
Qtech(config-keychain-key)#accept-lifetime 16:30:00 Oct 1 2010 duration 43200
Qtech(config-keychain-key)#send-lifetime 16:30:00 Oct 1 2010 duration 43200
Qtech(config-keychain-key)#exit
```

! Configure the second key of "Key 2", which contains the key-string of "World", and configure the corresponding period needed.

```
Qtech(config-keychain)#key 2
Qtech(config-keychain-key)#key-string World
Qtech(config-keychain-key)#accept-lifetime 04:00:00 Oct 2 010 infinite
Qtech(config-keychain-key)#send-lifetime 04:00:00 Oct 2 2010 infinite
Qtech(config-keychain-key)#end
```

! Configure G0/1 to use the MD5 authentication key to authenticate the update messages sent from device A.

```
Qtech#configure terminal
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-if)#ip rip authentication key-chain ripchain
Qtech(config-if)#ip rip authentication mode md5
Qtech(config-if)#exit
```

! Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.27.0
```

3.3.4.5 Verification

Run the show run command to verify the correctness of configurations (taking device A as the example):

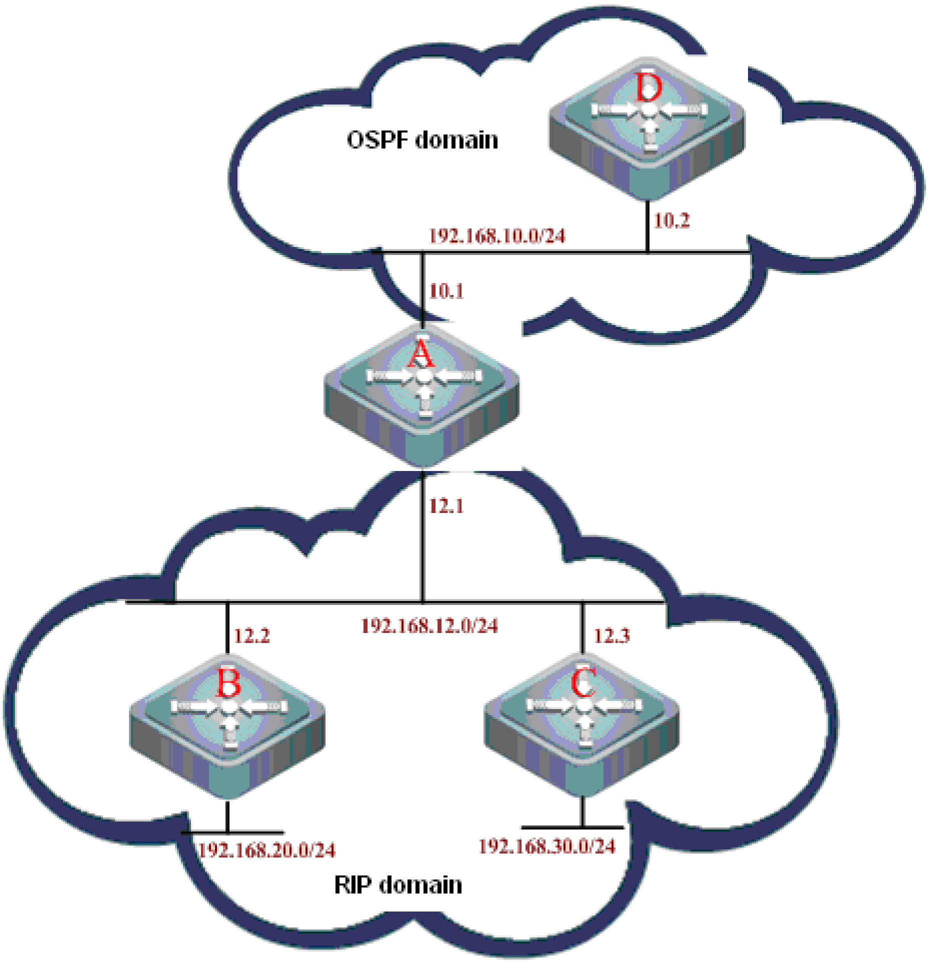
```
Qtech#show run

Building configuration...
Current configuration : 1561 bytes

!
vlan 1
!
!
key chain ripchain
 key 1
  key-string Hello
     accept-lifetime 16:30:00 Oct 01 2010 duration 43200
  send-lifetime 16:30:00 Oct 01 2010 duration 43200
 key 2
  key-string World
     accept-lifetime 04:00:00 Oct 02 2010 infinite
     send-lifetime 04:00:00 Oct 02 2010 infinite
!
no service password-encryption
!
interface GigabitEthernet 0/1
      ip rip authentication mode md5
 ip rip authentication key-chain ripchain
 no ip proxy-arp
     ip address 192.168.27.1 255.255.255.0
```

```
!
interface GigabitEthernet 0/2
!
interface GigabitEthernet 0/3
!
interface GigabitEthernet 0/4
...
!
!
!
router rip
 version 2
 network 192.168.27.0
!
!
!
line con 0
line vty 0 4
 login
!
!
end
```

3.3.5 RIP Redistribution and Default Route

3.3.5.1 Networking Topology

Figure 5 Topology for RIP redistribution and default route



3.3.5.2 Networking Requirements

Devices A, B, and C are interconnected in the same network segment and run the RIP routing protocol. Devices A and D are interconnected in the same network segment and run the OSPF routing protocol. Configure these four devices to achieve the following goals:

- Device A can learn the OSPF routes advertised by device D;
- Device A can redistribute OSPF routes to RIP;
- Device A advertises the redistributed routes to devices B and C;
- Device C advertises the default routing to devices A and B.

3.3.5.3 Configuration Tips

- Configure to redistribute OSPF routes to RIP in the RIP process of device A;
- Configure to advertise the default routing on the corresponding interface of device C;

3.3.5.4 Configuration Steps

#Configure device A:

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.12.1 255.255.255.0
Qtech(config-if-FastEthernet 0/1)#exit
Qtech(config)#interface FastEthernet0/2
Qtech(config-if-FastEthernet 0/2)#ip address 192.168.10.1 255.255.255.0
```

# Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.12.0
Qtech(config-router)#redistribute ospf 10 metric 3
```

//Redistribute the OSPF routing progress in the RIP progress, with metric value being 3

# Configure the OSPF routing protocol.

```
Qtech(config)#router ospf 10
Qtech(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

#Configure device B:

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.12.2 255.255.255.0
Qtech(config-if-FastEthernet 0/1)#exit
```

! Configure loopback ports.

```
Qtech(config)#interface Loopback 0
Qtech(config-if-Loopback 0)#ip address 192.168.20.1 255.255.255.0
```

! Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.12.0
Qtech(config-router)#network 192.168.20.0
```

# Configure device C:

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.12.3 255.255.255.0
Qtech(config-if-FastEthernet 0/1)#ip rip default-information originate metric 5
```

//Advertise default route, with metric value being 5

! Configure loopback ports.

```
Qtech(config)#interface Loopback 0
Qtech(config-if-Loopback 0)#ip address 192.168.30.1 255.255.255.0
```

# Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.12.0
Qtech(config-router)#network 192.168.30.0
```

# Configure device D:

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.10.2 255.255.255.0
```

! Configure the OSPF routing protocol.

```
Qtech(config)#router ospf 10
Qtech(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

3.3.5.5 Verification

View the routing table of each device (mainly the routing information on devices A, B, and C):

View the routing table on device A, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R*   0.0.0.0/0 [120/5] via 192.168.12.3, 00:00:23, FastEthernet 0/1
C    192.168.10.0/24 is directly connected, FastEthernet 0/2
C    192.168.10.1/32 is local host.
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.1/32 is local host.
R    192.168.20.0/24 [120/1] via 192.168.12.2, 00:07:09, FastEthernet 0/1
R    192.168.30.0/24 [120/1] via 192.168.12.3, 00:00:23, FastEthernet 0/1
```

View the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.10.0/24 [120/3] via 192.168.12.1, 00:00:06, FastEthernet 0/1
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.2/32 is local host.
C    192.168.20.0/24 is directly connected, Loopback 0
C    192.168.20.1/32 is local host.
R    192.168.30.0/24 [120/3] via 192.168.12.3, 00:00:06, FastEthernet 0/1
```

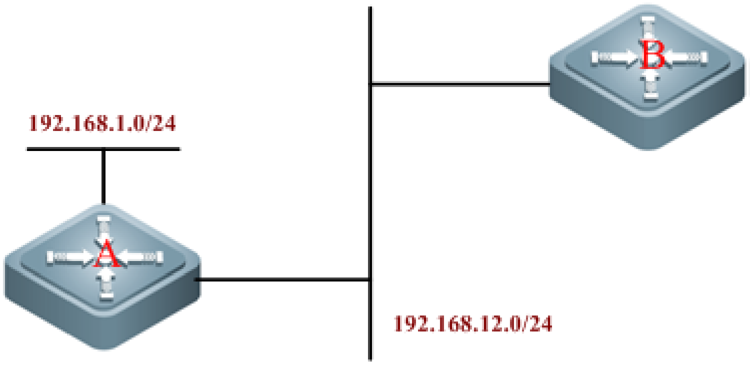
View the routing table on device C, as shown below (the bold figures are the routing information learned through RIP):

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.10.0/24 [120/3] via 192.168.12.1, 00:01:49, FastEthernet 0/1
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.3/32 is local host.
C    192.168.30.0/24 is directly connected, Loopback 0
C    192.168.30.1/32 is local host.
R    192.168.20.0/24 [120/3] via 192.168.12.2, 00:01:49, FastEthernet 0/1
```

3.3.6 RIP Supernet Route

3.3.6.1 Networking Topology

Figure6 Topology for the RIP supernet route



3.3.6.2 Networking Requirements

Two devices are interconnected through Ethernet. Device A runs RIPv2, and device B only supports the RIPv1 protocol and is unable to learn supernet routes.

Requirements:

- Configure supernet route 80.0.0.0/6 on device A, with next hop pointing to interface loopback 1 (192.168.1.0);
- Redistribute the aforementioned static route to RIP;
- Prohibit advertising supernet routes on device A.

3.3.6.3 Configuration Tips

Device B supports only the RIPv1 protocol. According to RFC 1058, such device is able to receive update packets of higher-version RIP, but such fields as subnet mask and next hop in the packets must be neglected. Therefore, route 80.0.0.0/6 received by device B will be treated as 80.0.0.0/8. To prevent device B from learning incorrect routes, device A must be configured to prohibit supernet route advertisement.

3.3.6.4 Configuration Steps

# Configure device A:

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.12.1 255.255.255.0
Qtech(config-if-FastEthernet 0/1)#no ip rip send supernet-routes
```

//Prohibit supernet route advertisement

! Configure loopback ports.

```
Qtech(config)#interface loopback 1
Qtech(config-if-Loopback 1)#ip address 192.168.1.1 255.255.255.0
```

! Configure static routes.

```
Qtech(config)#ip route 80.0.0.0 252.0.0.0 loopback 1
```

! Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#version 2
Qtech(config-router)#network 192.168.12.0
Qtech(config-router)#network 192.168.1.0
Qtech(config-router)#redistribute static
```

//Redistribute static route

# Configure device B (supporting RIPv1 only):

! Configure Ethernet ports.

```
Qtech(config)#interface FastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)#ip address 192.168.12.3 255.255.255.0
```

! Configure the RIP routing protocol.

```
Qtech(config)#router rip
Qtech(config-router)#network 192.168.12.0
```

3.3.6.5 Verification

View the routing table of each device;

View the routing table on device A, as shown below:

```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
S     80.0.0.0/6 is directly connected, Loopback 1
C     192.168.1.0/24 is directly connected, Loopback 1
C     192.168.1.1/32 is local host.
C     192.168.12.0/24 is directly connected, FastEthernet 0/1
C     192.168.12.1/32 is local host.
```

View the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

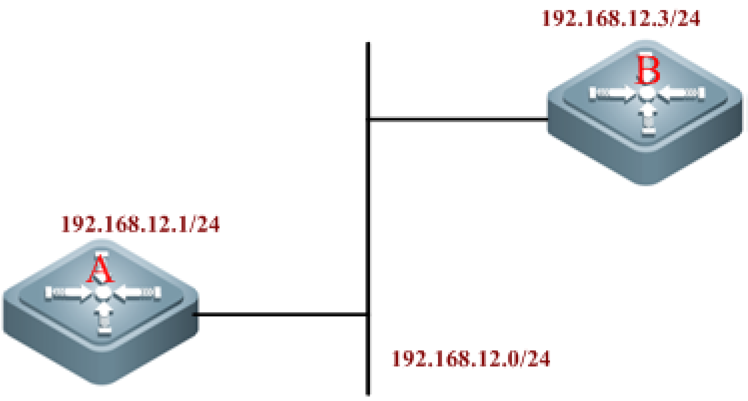
```
Qtech#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R     80.0.0.0/6 [120/1] via 192.168.12.1, 00:00:46, GigabitEthernet 0/1
R     192.168.1.0/24 [120/1] via 192.168.12.1, 00:38:17, FastEthernet 0/1
C     192.168.12.0/24 is directly connected, FastEthernet 0/1
C     192.168.12.2/32 is local host.
```

3.3.7 RIP VRF Configuration Examples

3.3.7.1 Networking Requirements

Two routing devices are interconnected through Ethernet and run the RIP routing protocol. The connection layout and IP address distribution are shown in Figure 7.

Figure7 Example of RIP VRF configuration



Through RIP, routing information is exchanged between VRF "redvpn" of device A and VRF "bluevpn" of device B.

By enabling RIP GR on device A and setting the GR period to 90 seconds, non-stop data forwarding can be realized during hot standby switchover between main and slave management boards on device A. Meanwhile, since the GR period has been changed, timers basic shall be configured to a reasonable value.

3.3.7.2 Detailed Configurations

Configure device A:

# Create VRF.

```
ip vrf redvpn
```

# Bind the interface to VRF and configure the interface IP address.

```
interface fastEthernet 0/1
ip vrf forwarding redvpn
ip address 192.168.12.1 255.255.255.0
```

# Configure the RIP routing protocol and create a RIP instance.

```
router rip
address-family ipv4 vrf redvpn
network 192.168.12.0
graceful-restart grace-period 90
timers basic 45 270 180
exit-address-family
```

Configure device B:

# Create VRF.

```
ip vrf  bluevpn
```

# Bind the interface to VRF and configure the interface IP address.

```
interface fastEthernet 0/1
ip vrf forwarding bluevpn
ip address 192.168.12.3 255.255.255.0
```

# Configure the RIP routing protocol and create a RIP instance.

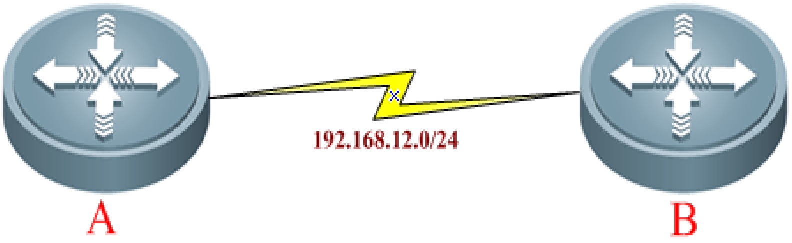
```
router rip
address-family ipv4 vrf bluevpn
network 192.168.12.0
timers basic 45 270 180
exit-address-family
```

3.3.8 TRIP Configuration Examples

3.3.8.1 Networking Requirements

Two routers are interconnected through the PPP link and run the RIP routing protocol. The connection layout and IP address distribution are shown in Figure 8.

Figure 8 Example of TRIP configuration



By configuring TRIP, routing information can be exchanged between devices A and B on the WAN link, and split horizon with poisoned reverse shall be enabled.

3.3.8.2 Detailed Configurations

Configure device A:

# Enable the PPP link protocol on the interface and configure the interface IP address; Enable TRIP and split horizon with poisoned reverse.

```
interface Serial 0/0
encapsulation ppp
ip address 192.168.12.1 255.255.255.0
ip rip triggered
ip rip split-horizon poisoned-reverse
```

# Configure the RIP routing protocol.

```
router rip
network 192.168.12.0
```

Configure device B:

# Enable the PPP link protocol on the interface and configure the interface address; Enable TRIP and split horizon with poisoned reverse.

```
interface Serial 0/0
encapsulation ppp
ip address 192.168.12.2 255.255.255.0
ip rip triggered
ip rip split-horizon poisoned-reverse
```

# Configure RIP routing protocol

```
router rip
network 192.168.12.0
```

4 CONFIGURING RIPNG

4.1 Overview

Similar to RIP, RIPng is a distance-vector routing protocol using hop count as the routing metric. RIPng is an interior gateway protocol applicable to small- and medium-sized networks. RIPng is the necessary extension of RIP to address the routing requirements of IPv6. Therefore, RIPng and RIP have basic working principles in common. The main differences rise from the format of their address and packet. Based on IPv6, RIPng supports and uses the multicast group address of FF02::9 for update messages. Security authentication used in RIP is also cancelled. Instead, RIPng enables security authentication by the security mechanism of IPv6. 521 port is used. Packet format, mask and maximum packet length are all different. Please refer to RFC2080 and RFC2081 for details. Given the difference from RIP, the corresponding CLI commands are also lesser.

4.2 RIPng Configuration List

- Creating RIPng routing process (Required)
- Enabling RIPng on the interface (Required)
- Adjusting RIPng timer (Optional)
- Configuring Split Horizon (Optional)
- Configuring default metric for redustribution
- Adjusting interface metric
- Configuring the advertisement default route on the interface
- Configuring passive interface
- Configuring RIPng route filtering
- Showing RIPng configuration

Please refer to the chapter of “Protocol independent Configuration” for the configuration of route redistribution.

The default configurations of RIPng are given below:

Function	Default setting
Network interface	The default metric-offset is 1.
Redistribution	The route redistribution is disabled by default. If enabled: ■ All sub-routes of the routing process are redistributed. ■ The metric of redistributed route is the default metric.
Split horizon	Enabled
Poisoned reverse	Disabled
Timer	By default: ■ Update time is 30 seconds ■ Invalid time is 180 second ■ Clearing time is 120 seconds
Default metric	Redistribute the metric used by the routes of other protocols，1 by default.
Administrative distance	120

4.2.1 Creating RIPng Routing Process

In order to run RIPng routing protocol, the routing device first needs to create the RIPng routing process and define the network or interface address associated with RIPng routing process.

To create RIPng routing process, input the following command in the global configuration mode:

Command	Function
Qtech(config)# **ipv6 router rip**	Create RIPng routing process.

4.2.2 Enabling RIPng on the Interface

To enable RIPng on the interface, input the following command in the interface configuration mode:

Command	Function
Qtech(config-if)# **ipv6 rip enable**	Enable RIPng on the interface.

In the following example, enable RIPng on ethernet 0/0. Routes in the range of 2001:db8:6::/64 are converged into 2001:db8:6::/64 for advertisement.

```
Qtech(config)# interface ethernet 0/0
Qtech(config-if)# ipv6 address 2001:db8:6::1/64
Qtech(config-if)# ipv6 rip enable
```

Note

Different from RIP, this command enables RIPng on the interface directly without configuring Network command.

4.2.3 Adjusting RIPng Timer

RIPng provides the feature of timer adjustment. You can adjust the timer according to the physical circumstances of the network, so that the RIPng routing protocol can run better. The following timers can be adjusted:

- Route update time (second): defines the interval by which the routing device will send the route update message;
- Route invalid time (second): defines the time by which the route in the routing table becomes invalid upon no update;
- Route clearing time (second): upon expiration of this time, the route will be removed from the routing table.
- By adjusting the aforementioned times, the convergence time and failure recovery time of routing protocol can be accelerated. To adjust RIPng timer, input the following command in the RIPng routing process configuration mode:

Command	Function
Qtech(config-router)# **times** *update invalid garbage-collection*	Adjust RIPng times.

The following example adjusts the values of three RIPng times:

```
Qtech(config-router)# timers 10 30 90
```

⚠ Caution Consistency of RIPng times is mandatory for routing devices in the same network.

4.2.4 Configuring Split Horizon

When multiple routing devices are linked to the IP broadcast network and running distance-vector routing protocol, it is necessary to adopt the mechanism of split horizon to avoid loop. Split horizon can prevent routing device from sending routing information to the port from which such routing information was learned. Such mechanism optimizes the routing information exchange between multiple routing devices.

However, for non-broadcast multiple-access network (such as frame relay, X.25 network), split horizon may cause disable certain routing devices from learning all routing information. In such a case, the split horizon will need to be disabled.

To disable or enable split horizon, input the following command in the routing process configuration mode:

Command	Function
Qtech(config-router)# **no split-horizon**	Disable split horizon.
Qtech(config-router)# **split-horizon**	Enable split horizon.

By default, split horizon is enabled on all RIPng ports.

Note The current version can only support split horizon configuration in RIPng routing process, i.e., this command will be applied to all RIPng interfaces. The subsequent RGOS software will realize interfaced-based split horizon configuration.

Different from split horizon, when poison reverse is enabled, the routing device will advertise certain route information from the interface from which such route information was learned. Just set the corresponding metric to infinity (16).

To enable or disable poison reverse, input the following command in the routing process configuration mode:

Command	Function
Qtech(config-router)#**split-horizon poisoned-reverse**	Enable split horizon with poison reverse
Qtech(config-router)#**no split-horizon poisoned-reverse**	Disable split horizon with poison reverse

⚠ Caution Enabling poison reverse will consume considerable bandwidth.

4.2.5 Configuring Default Metric for Redistribution

When a protocol redistributes routes of other protocols, you need to configure the metric for such redistribution for metric varies by protocols. The default metric of RIPng is 1.

To define the default RIPng metric during the redistribution of other routing protocols, please use the routing process configuration command of "**default-metric**". Use "**no default-metric**" command to reset the default value to 1.

Command	Function
Qtech(config-router)# **default-metric** *metric*	The metric of RIPng is set to *metric*
Qtech(config-router)# **no default-metric**	Reset the default value to 1

4.2.6 Adjusting Interface Metric

Before adding the learned routes into the routing table, you need to add the metric set for the interface to the ones of learned routes. Therefore, you can control the use of routes by configuring the interface metric.

To configure the interface metric, input the following command in the interface configuration mode:

Command	Function
Qtech(config-if)# **ipv6 rip metric-offset** *value*	Configure interface metric within the scope of 1-16.

The following example sets the metric of ethernet 0/0 to 6:

```
Qtech(config)# interface ethernet 0/0
Qtech(config-if)# ipv6 rip metric-offset 6
```

4.2.7 Configuring the Advertisement Default Route on the Interface

To generate an IPv6 default route in the update message of this RIPng process (::/0), input the following command in the interface configuration mode:

Command	Function
Qtech(config-if)# **ipv6 rip default-informaton originate**	Generate a default route to RIPng on the interface and advertise it with other routes.

To generate an IPv6 default route in the update message of this RIPng process (::/0), and advertise only this default route on this interface, input the following command in the interface configuration mode:

Command	Function
Qtech(config-if)# **ipv6 rip default-informaton only**	Generate a default route to RIPng on the interface and advertise only this default route.

4.2.8 Configuring Passive Interface

To prevent other routing devices in the local network from learning the routing information sent by the routing device, configure passive interface to disable sending routing update message from this network interface.

To disable sending update messages from an interface, input the following command in the routing process configuration mode:

Command	Function
Qtech(config-router)# **passive-interface** {**default** | *interface-type interface-num*}	Disable sending update messages from the interface.

Note

When applying the **default** option, all interfaces will be set to passive mode; when applying the **interface** option, the corresponding interface will set to passive mode.

4.2.9 Configuring RIPng Route Filtering

4.2.9.1 Controlling Route Update Advertisement (RIPng)

To prevent other routing devices in the local network from learning unnecessary routing information, disable the update of specific routes by controlling RIPng route update advertisement.

To disable route update advertisement, input the following command in the routing process configuration mode:

Command	Function
Qtech(config-router)# **distribute-list prefix-list** *prefix-list-name* **out** [*interface-type interface-name*]	According to the rule of prefix list, enable or disable the advertisement of certain routes.
Qtech(config-router)# **no distribute-list prefix-list** *prefix-list-name* **out** [*interface-type interface-name*]	Remove the configuration.

In the following example, filtering is only applied to update messages sent from interface eth0, and only update routes included in the prefix-list *outlist* will be sent out.

```
Qtech(config)# ipv6 router rip
Qtech(config-router)# distribute-list prefix-list outlist out eth0
```

4.2.9.2 Controlling Route Update Processing (RIPng)

This feature can be configured to avoid receiving certain routes in the route update message.

To control route update processing, input the following command in the routing process configuration mode:

Command	Function
Qtech(config-router)# **distribute-list prefix-list** *prefix-list-name* **in** [*interface-type interface-name*]	According to the rule of access list, enable or disable the receipt of certain routes in the route update.
Qtech(config-router)# **no distribute-list prefix-list** *prefix-list-name* **in** [*interface-type interface-name*]	Remove the configuration.

In the following example, filtering is only applied to update messages received by interface eth0, and only update routes included in the prefix-list *inlist* will be received.

```
Qtech(config)# ipv6 router rip
Qtech(config-router)# distribute-list prefix-list inlist in eth0
```

4.2.10 Showing RIPng Configuration

4.2.10.1 RIPng Debugging Switch

To show the debugging information of RIPng and observe the route processing behaviors of RIPng, input the following command in the privilege configuration mode:

Command	Function
Qtech# **debug ipv6 rip** [{*interface-type interface-num* | **nsm**}]	Turn on RIPng debugging switch.
Qtech# **no debug ipv6 rip** [{*interface-type interface-num* | **nsm**}]	Turn off RIPng debugging switch.

4.2.10.2 Showing RIPng Routing Table

To show RIPng routing table, input the following command in user mode or privileged EXEC mode:

Command	Function
Qtech# **show ipv6 rip database**	Display RIPng routing table information.

4.2.10.3 Showing RIPng Routing Process

To display the parameters and various statistical data of RIPng routing process, input the following command in user mode or privilege mode:

Command	Function
Qtech# **show ipv6 rip**	Display RIPng routing process information.

4.2.10.4 Showing RIPng Debugging Information

To display the debugging information of RIPng routing process, input the following command in the privileged EXEC mode or global configuration mode:

Command	Function
Qtech# **show debugging**	Display the debugging information of RIPng routing process.

4.3 Configuration Examples

4.3.1 Configuring Default Route Advertisement

Configuration Requirements

There are three devices (see Fig 1 for device connection) running RIPng. The gateway device of Router A advertises the default route to Router B and Router C, with metric being 3. All RIPng interfaces of Router B and Router C are configured to passive mode, so as not to send out RIPng update messages.

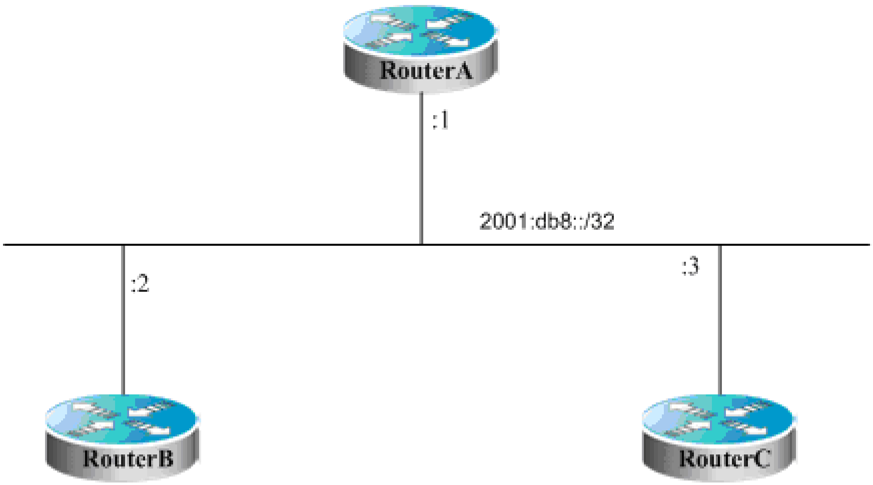


Figure 2 Configuration of default route advertisement

Detailed Configuration of Routing Devices

Router A:

# Configure network interface

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8::1/32
Qtech(config-if)# ipv6 rip enable
Qtech(config-if)# ipv6 rip default-information originate
metric 3
# Configure RIPng
Qtech(config)# ipv6 router rip
Qtech(config-router)# exit
Router B:
# Configure network interface.
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8::2/32
Qtech(config-if)# ipv6 rip enable
# Configure RIPng.
Qtech(config)# ipv6 router rip
Qtech(config-router)# passive-interface default
Qtech(config-router)# exit
Router C:
# Configure network interface.
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8::3/32
Qtech(config-if)# ipv6 rip enable
# Configure RIPng.
Qtech(config)# ipv6 router rip
Qtech(config-router)# passive-interface default
Qtech(config-router)# exit
```

4.3.2 Redistribution Configuration

Configuration Requirements

There are three devices (see Fig 1 for device connection). Router A runs RIPng; Router C runs BGP and introduces static routes; Router B needs to redistribute the static routes redistributed by Router C to RIPng domain.

In order to meet such requirements, we can configure the specified community attribute for static routes redistributed to BGP on Router C, while Router B can redistribute BGP routes with specified community attribute to the RIPng domain.

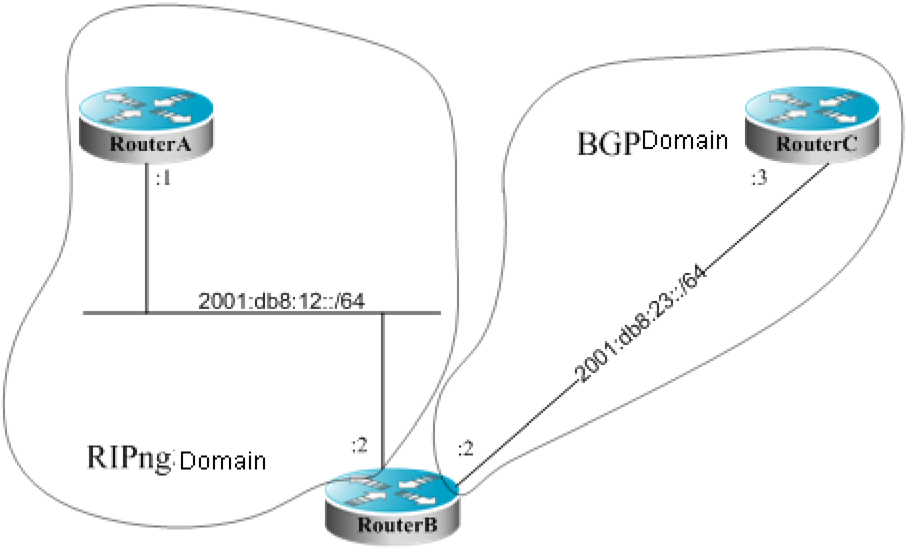


Figure 3 Redistribution configuration

Detailed Configuration of Routing Devices

Router A:

```
# Configure network interface.
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:12::1/64
Qtech(config-if)# ipv6 rip enable
# Configure RIPng.
Qtech(config)# ipv6 router rip
Qtech(config-router)# exit
Router B:
# Configure network interface.
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:12::2/64
Qtech(config-if)# ipv6 rip enable
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:23::2/64
# Configure RIPng.
Qtech(config)# ipv6 router rip
Qtech(config-router)# redistribute bgp route-map riprm
Qtech(config-router)# exit
# Configure BGP.
Qtech(config)# router bgp 2
Qtech(config-router)# neighbor 2001:db8:23::3 remote-as 3
Qtech(config-router)# address-family ipv6
Qtech(config-router-af)# neighbor 2001:db8:23::3 activate
# Configure route map.
Qtech(config)# route-map riprm
Qtech(config-route-map)# match community cl_110
# Define community list.
Qtech(config)# ip community-list standard cl_110 permit 22:22
Router C:
# Configure network interface.
```

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:23::3/64
# Configure BGP.
Qtech(config)# router bgp 3
Qtech(config-router)# neighbor 2001:db8:23::2 remote-as 2
Qtech(config-router)# address-family ipv6
Qtech(config-router-af)# redistribute static route-map bgprm
Qtech(config-router-af)# neighbor 2001:db8:23::2 activate
Qtech(config-router-af)# neighbor 2001:db8:23::2 send-community
# Configure static route.
Qtech(config)# ipv6 route 2001:db8:88::/64 null 0
Qtech(config)# ipv6 route 2001:db8:99::/64 null 0
# Configure route map.
Qtech(config)# route-map bgprm
Qtech(config-route-map)# set community 22:22
```

5 CONFIGURING OSPF

5.1 Overview

An open shortest path first (OSPF) routing protocol is an internal gateway routing protocol based on link status developed by the IETF OSPF work group It is designed for the IP environment and directly runs on the IP layer. With the protocol number being 89, this routing protocol exchanges OSPF packets in a manner of multicast by using the multicast address 224.0.0.5 (for all OSPF routers) and 224.0.0.6 (for specified routers).

The link status algorithm is an algorithm totally different from the Huffman vector algorithm (distance vector algorithm). The traditional RIP routing protocol uses the Huffman vector algorithm, while the OSPF routing protocol uses the link status algorithm. Compared with the RIP routing protocol, the OSPF routing protocol uses a different algorithm and introduces new concepts such as route update authentication, VLSMs, and route aggregation. Even if the RIPv2 has been improved greatly and also supports the features such as route update authentication and VLSM, the RIP routing protocol still has the following fatal weaknesses: 1) Slow convergence; 2) Limited network scale (with the maximum number of hops counting less than 16). The OSPF routing protocol overcomes these weaknesses, enabling the IGP protocol to be used in large and complicated network environments.

The OSPF routing protocol establishes and calculates the shortest path to each target network by using this complicated link status algorithm. Brief information about how the link status algorithm works is as follows:

- In the initialization stage, a router generates a link status notification that contains all link status of its own.
- All routers exchange the link status information in the multicast way. Upon receiving a link status update packet, each router copies the packet into the local database and then transmits the packet to other routers.
- After every router has a complete link status database, a router uses the Dijkstra algorithm to calculate the shortest path trees to all target networks. The results include the target network, next-hop address, and cost, which are the key parts of an IP routing table.

When there is no link cost or network change, the OSPF is inactive. When any changes occur on the network, the OSPF advertises the link status changes of only the changed links. The routers involved in the changes will run the Dijkstra algorithm again to generate new shortest path trees.

A group of routers running the OSPF routing protocol form the autonomous system of the OSPF routing area. An autonomous system consists of all the routers that are controlled and managed by one organization. Within the autonomous system, only one IGP routing protocol is run. However, between multiple autonomous systems, the BGP routing protocol is used to exchange routing information. Different autonomous area systems may use the same IGP routing protocol. Every autonomous system needs to request the related organization for the autonomous system number to connect to the Internet.

When the OSPF routing area is large, the hierarchical structure can be used. In other words, the OSPF routing area can be divided into several areas, which are connected via a backbone area. Every non-backbone area must be directly connected to the backbone area.

There are three roles for the routers in the OSPF routing area based on their deployment positions:

1) Area internal router: All interface networks of this router belong to a same area.

2) Area border router (ABR): The interface network of this router belongs at least to two areas, one of which must be the backbone area.

3) Autonomous system boundary router (ASBR): It is the router through which routes are exchanged between the OSPF route area and the external route area.

Qtech products use the OSPF by fully complying with the OSPFv2 defined in RFC 2328. The main features are described as follows:

- Support multiple OSPF processes.
- Support the VRF. You can run the OSPF routing protocol based on different VRFs.
- Support the definition of the stubby area.
- Support route redistribution of static routes, directly-connected routes, dynamic routes, and the routing information among dynamic routing protocols such as RIP and BGP.
- Support plain-text or MD5 authentication between neighbors.
- Support virtual links.
- Support VLSMs.

- Support area division.
- Support the not so stubby area (NSSA) feature, as defined in RFC 3101.
- Support the graceful restart feature, as defined in RFC 3623.

☑ Qtech products do not support the following functions now: OSPF line support on demand, as defined in RFC 1793; OSPF fast convergence.

5.2 Configuration Task List

The configuration of OSPF should be cooperated with various routers, including internal routers, area border routers, and autonomous system boundary routers. When no configuration is performed on routers, default parameters are used. In this case, packets are sent and received without authentication, and the interface does not belong to any area of an autonomous system. When changing the default parameters, you must ensure that the routers have the same configurations.

To configure an OSPF routing protocol, you must perform the following tasks. Among these tasks, creating the OSPF routing process is mandatory. Other tasks may be optional or mandatory in particular applications. The detailed tasks to configure the OSPF routing protocol are described as follows:

- Creating an OSPF routing process (mandatory)
- Configuring OSPF interface parameters (optional)
- Configuring the OSPF used on different physical networks (optional)
- Configuring OSPF area parameters (optional)
- Configuring an OSPF NSSA (optional)
- Configuring OSPF route aggregation (optional)
- Creating a virtual link (optional)
- Generating a default route (optional)
- Using the loopback interface address as the router ID (optional)
- Changing the default OSPF management distance (optional)
- Configuring the route calculation timer (optional)
- Configuring the link status advertisement (LSA) group pacing timer (optional)
- Configuring the cost for the OSPF interface (optional)
- Configuring an OSPF stub router (optional)
- Configuring whether to perform the MTU check on an interface (optional)
- Disabling an Interface to send the OSPF packets (optional)
- Configuring whether to perform the source address check (optional)
- Configuring the OSPF fast convergence function (optional)
- Configuring the OSPF capacity protection function (optional)
- Configuring the OSPF network management function(optional)
- Configuring the OSPF GR function (optional)
- Configuring the OSPF BFD function (optional)
- Configuring the OSPF VPN function (optional)
- Monitoring and maintaining the OSPF

For the configuration information about the following topics, see related sections in *Configuring Protocol-Independent Information*.

- Filtering the routing information
- Redistributing routes

Default OSPF configurations are described as follows:

Feature	Default Setting
Interface parameters	Interface metric: Not preset. LSA retransmission interval: 5 seconds. LSA transmission delay: 1 second. Interval for transmitting Hello packets: 10 seconds (30 seconds for non-broadcast networks) Failure time of adjacent routers: 4 times the interval for transmitting the Hello packets. Fast Hello: Disabled. Priority: 1. Authentication type: 0 (No authentication). Authentication password: None.
Area	Authentication type: 0 (No authentication).

Feature	Default Setting
	Default metric of aggregated routes to a Stub or NSSA area: 1. Inter-area aggregation scope: Undefined. Stub area: Undefined. NSSA: Undefined. Translator for translating Type-7 LSAs to Type-5 LSAs: Alternative. Interval of stabilizing the translation from Type-7 LSAs to Type-5 LSAs: 40 seconds.
Virtual links	No virtual link is defined. The default parameters of the virtual link are as follows: LSA retransmission interval: 5 seconds. LSA transmission delay: 1 second. Interval for transmitting Hello packets: 10 seconds. Failure time of adjacent routers: 4 times the interval for transmitting the Hello packets. Fast Hello: Disabled. Authentication type: No authentication. Authentication password: None.
Automatic cost calculation	Enabled. Default value automatically calculated is 100 Mbit/s.
Default route generation	Disabled. The default metric is 1 and the type is type-2 if enabled.
Default metric (Default metric)	Default metric used to redistribute other routing protocols
Management distance	Intra-area routing information: 110 Inter-area routing information: 110 External routing information: 110
Database filter	Disabled. All interfaces can receive the status update information (LSA).
Neighbor change log	Enabled.
Neighbor	N/A
Neighbor database filter	Disabled. All output LSAs are sent to all neighbors.
Network area (network area)	N/A
Router ID	Undefined. The OSPF protocol does not run on a router by default.
External route aggregation (summary-address)	Undefined.
Changing time of the status update information	240 seconds
Shortest path first (SPF) calculation timer	The time delay between the time for receiving information about topology changes and the next time for invoking the SPF calculation: 1000 milliseconds. The minimum interval between two calculating operations using the SPF algorithm: 5000 milliseconds. The maximum interval between two calculating operations using the SPF algorithm: 10000 milliseconds.
Optimal path rule used to calculate the external routes	Rules defined in RFC1583
OSPF stub router	Disabled.
OSPF two-way maintenance	Enabled.
Sending LSA packet updates	Time interval for sending data packets: 40 milliseconds. Number of LS-UPD packets in each data packet: 10.
OSPF overflow	Enter the overflow state when the memory lacks.
OSPF GR	GR restarter: Disabled. GR helper: Enabled.
OSPFv2 MIB binding	OSPFv2 process with the smallest process number
OSPFv2 TRAP sending	Disabled.

5.2.1 Creating an OSPF Routing Process

You can create an OSPF routing process and define the range of the IP addresses associated with the OSPF routing process and the OSPF area to which these IP addresses belong. The OSPF routing process only sends and receives the OSPF packets at the interface within the IP address range and advertises the link status of the interface to external routers.

Use the following commands to create the OSPF routing process.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **ip routing**	Enables the IP routing function (if disabled).
Qtech (config)# **router ospf** [*process_id* [**vrf** *vrf-name*]]	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **network** *address wildcard-mask* **area** *area-id*	Defines an IP address range for an area.
Qtech (config-router)# end	Returns to privileged EXEC mode.
Qtech # show ip protocols	Displays the routing protocol that is running currently.
Qtech # write	Saves the configurations.

Note

You can use the parameter *vrf vrf-name* to specify the VRF which the OSPF belongs to. If you do not specify this parameter when creating the OSPF routing process, the default VRF is used. For the **network** command, 32 bit wildcards are opposed to the mask. The value 1 indicates that the bit is not compared, and the value 0 indicates that the bit is compared. However, if you configure the command with masks, Qtech products automatically translate the masks into bit wildcards. An interface belongs to the specific area as long as the address of the interface is within the IP address range defined in the **network** command. When the address of an interface is within more than one IP address ranges defined in the **network** command of multiple OSPF processes, the OSPF process that the interface involves in is determined based on the optimal mapping.

To disable the OSPF protocol, use the **no router ospf** *process-id* command. The following example shows how to enable the OSPF protocol:

```
Qtech (config)# router ospf 1
Qtech (config-router)# network 192.168.0.0 255.255.255.0 area 0
Qtech (config-router)# end
```

5.2.2 Configuring OSPF Interface Parameters

You are allowed to change some particular interface parameters and configure the interface parameters on demand. It should be noted that some parameters must match those of the adjacent router of the interface. These parameters are set by using the **ip ospf hello-interval**, **ip ospf dead-interval**, **ip ospf authentication**, **ip ospf authentication-key**, and **ip ospf message-digest-key** commands. When you use these commands, make sure that the adjacent routers have the same configurations.

Use the following commands to configure the OSPF interface parameters in interface configuration mode.

Command	Function
Qtech# **configure** terminal	Enters global configuration mode.
Qtech(config)# **ip routing**	Enables the IP routing function (if disabled).
Qtech(config)# **interface** *interface-id*	Enters interface configuration mode.
Qtech(config-if)# **ip ospf cost** *cost-value*	(Optional) Defines the cost value for the interface.
Qtech(config-if)# **ip ospf retransmit-interval** *seconds*	(Optional) Sets the link status retransmission interval.
Qtech(config-if)# **ip ospf transmit-delay** *seconds*	(Optional) Sets the transmission delay for the link status update packets.
Qtech(config-if)# **ip ospf hello-interval** *seconds*	(Optional) Sets the interval for sending the Hello packets, which must be same for all the nodes of the entire network.
Qtech(config-if)# **ip ospf dead-interval** *seconds*	(Optional) Sets the dead interval for the adjacent routers, which must be same for all the nodes of the entire network.
Qtech(config-if)# **ip ospf priority** *number*	(Optional) Priority, used to select the dispatched routers (DR) and backup dispatched routers (BDR).
Qtech(config-if)# **ip ospf authentication [message-digest | null]**	(Optional) Sets the authentication type on the interface.

Command	Function
Qtech(config-if)# **ip ospf authentication-key [0|7]** *key*	(Optional) Configures the text authentication key on the interface.
Qtech(config-if)# **ip ospf message-digest-key** ***keyid*** **md5 [0|7]** *key*	(Optional) Configures the key for the MD5 authentication on the interface.
Qtech (config-if)# **ip ospf database-filter all out**	(Optional) Prevents the interface from flooding the link status update packets. By default, the OSPF floods the LSA information over all interfaces in the same area except the interface on which the LSA information is received.
Qtech (config-if)# **end**	Returns to privileged EXEC mode.
Qtech # show **ip ospf interface** [*interface-id*]	Displays the OSPF interface information.
Qtech # **write**	(Optional) Saves the configurations.

To restore the default value, use the **no** form of the above commands.

5.2.3 Configuring the OSPF Used on Different Physical Networks

According to the transmission features of different media, the networks are classified into three types according to the OSPF protocol:

- Broadcast network (Ethernet, token network, and FDDI)
- Non-broadcast network (frame relay, X.25)
- Point-to-point network (HDLC, PPP, and SLIP)

The non-broadcast networks include two types of networks according to the operation modes of the OSPF:

1) Non-broadcast multi-access (NBMA) network: The NBMA network requires direct communication for all interconnected routers. Only fully meshed networks can meet this requirement. If the SVC (for example, X.25) networking is used, this requirement can be met. However, it is difficult to use the PVC (for example, frame relay) networking to meet this requirement. The operation of the OSPF on the NBMA network is similar to that on the broadcast network. That is, a designated router must be specified to advertise the link status on the NBMA network.

2) Point-to-multipoint network: If the network is not a fully meshed non-broadcast network, you need to set the network type of the interface to the point-to-multipoint network type according to the OSPF routing protocol. In a point-to-multipoint network, the connections between all routers are treated as point-to-point links according to the OSPF routing protocol, so you do not need to specify the designated router.

Whatever the default network type of the interface is, you can set it to the broadcast network type. For example, you can set the non-broadcast multi-access network (frame relay, X.25) to a broadcast network. The step to configure the neighbor routers can be omitted during the OSPF routing process configuration. By using the **X.25 map** and **Frame-relay map** commands, you can enable the broadcast function on the X.25 and frame relay networks. In this case, the X.25 and frame relay networks are treated as the broadcast networks.

The point-to-multipoint network interface can be seen as the marked point-to-point interface of one or more neighbors. When the network type of is configured to the point-to-multipoint network type according to the OSPF routing protocol, multiple host routes are generated. Compared with the NBMA network, the point-to-multipoint network has the following advantages:

- Easy configuration without configuring the neighbors or specifying the designated router.
- Low cost without requiring fully meshed topology

Use the following command to configure the network type in interface configuration mode.

Command	Function
Qtech(config-if)# **ip ospf network** {**broadcast** | **non-broadcast** | **point-to-point** | {**point-to-multipoint** [**non-broadcast**]}}	Configures the OSPF network type.

For different link encapsulation types, the default network types described as follows:

- Point-to-point network type:

For PPP, SLIP, frame relay point-to-point subinterface, and X.25 point-to-point subinterface encapsulation

- NBMA (non-broadcast) network type:

For frame relay, X.25 encapsulation (except the point-to-point subinterface)

- Broadcast network type:

For Ethernet encapsulation

- The point-to-multipoint network type has no default.

It should be noted that the network type should be consistent at both sides. Otherwise, exceptions occur. For instance, the neighbor is Full and the routing calculation is incorrect.

5.2.3.1 Configuring a Point-to-multipoint Broadcast Network

When routers are interconnected using the X.25 and frame relay networks, if the network is not a fully meshed network or you do not want to specify the designated router, you can set the network type of the OSPF interface to the point-to-multipoint type. Since the links are treated as point-to-point links on the point-to-multipoint network, multiple host routes are generated. In addition, all neighbors have the same cost value on the point-to-multiple network. If you want to enable different neighbors to have different cost values, you can set the cost by using the **neighbor** command.

Use the following commands to configure the point-to-multipoint network type in interface configuration mode.

Command	Function
Qtech(config-if)# **ip ospf network point-to-multipoint**	Sets the broadcast network type for an interface to point-to-multipoint.
Qtech(config-if)# **exit**	Returns to global configuration mode.
Qtech(config)# **router ospf *1***	Enters routing process configuration mode.
Qtech(config-router)# **neighbor ip-address cost** *cost*	(Optional). Specifies the cost of the neighbor router.

Note

Although the OSPF point-to-multipoint network is a non-broadcast network, the non-broadcast network is allowed to have the broadcast capability by manual configuration or self-learning according to the frame relay and X.25 mapping. Therefore, you do not need to specify neighbors when configuring the point-to-multipoint network.

5.2.3.2 Configuring a Non-broadcast Network

When the OSPF routing protocol works on a non-broadcast network, you can set the network type to the NBMA or point-to-multipoint non-broadcast type. Since a non-broadcast network do not have the broadcast capability and cannot dynamically discover neighbors, you must manually configure neighbors for the non-broadcast network when the OSPF routing protocol is used.

Set the network type as the NBMA type in the following conditions:

- When a non-broadcast network has the fully meshed topology;
- When a broadcast network is configured as the NBMA network type to reduce the generation of the broadcast packets, save the network bandwidth, and avoid some arbitrary reception and transmission of routers. During the configuration of the NBMA network, you must specify neighbors and the designated router. Therefore, you must configure the priorities for the routers. It is more possible for the route with a higher priority to be specifies as the designated router.

Use the following commands to set the network type to the NBMA type in interface configuration mode.

Command	Function
Qtech (config-if)# **ip ospf network non-broadcast**	Specifies the network type of the interface to NBMA.
Qtech (config-if)# **exit**	Returns to global configuration mode.
Qtech (config)# **router ospf** *1*	Enters routing process configuration mode.
Qtech(config-router)# **neighbor** *ip-address* [*priority number*] [*poll-interval seconds*]	Specifies the neighbor, its priority, and polling interval of Hello packets.

The best solution is to set the network where the OSPF is used to the point-to-multipoint non-broadcast network when you cannot make sure whether any two routers on a non-broadcast network are reachable directly.

All neighbors on the point-to-multipoint broadcast or non-broadcast network have the same cost value which is configured by using the **ip ospf cost** command. However, the bandwidth of each neighbor may be different, so the cost is different. You can specify the cost for each neighbor by using the **neighbor** command. However, this only applies to the interface used on the point-to-multipoint type (broadcast or non-broadcast) network.

Use the following commands to set the type of the interface as the point-to-multipoint type on a non-broadcast network in interface configuration mode.

Command	Function
Qtech (config-if)# **ip ospf network point-to-multipoint non-broadcast**	Specifies the network type of the interface to be the point-to-multipoint non-broadcast network type.
Qtech (config-if)# **exit**	Returns to global configuration mode.
Qtech (config)# **router ospf** *1*	Enters routing process configuration mode.
Qtech(config-router)# **neighbor** *ip-address* [*cost number*]	Specifies the neighbor and the cost to the neighbor.

Pay attention to step 4. If you have not specified the cost for the neighbor, the cost referenced in the **ip ospf cost** command in interface configuration mode is used.

5.2.3.3 Configuring the Broadcast Network Type

It is necessary to specify the designated router (DR) and backup designated router (BDR) for the OSPF broadcast network. The DR advertises the link status of this network to external devices. All routers keep the neighbor relationship with each another and only the adjacent relationship with the designated router and backup designated router. That is to say, each router only exchanges the link status packets with the designated router and backup designated router. Then the designated router advertises the link status information to all other routers. As a result, each router can store a consistent link status database.

You can control the specifying result of the designated router by configuring the OSPF priority. This parameter does not take effect immediately until the new round for specifying the designated router. The new round for specifying the designated router occurs only when the OSPF neighbors do not receive the Hello packets from the designated router within the specified time and judge that the DR is down.

Use the following commands to configure the broadcast network type in interface configuration mode.

Command	Function
Qtech(config-if)# **ip ospf network broadcast**	Specifies the type of the interface to be the broadcast network type.
Qtech(config-if)# **ip ospf priority** *priority*	(Optional) Specifies the priority of the interface.

5.2.4 Configuring OSPF Area Parameters

To configure area authentication, stub area, and default route summary cost, you need to configure area commands.

The area authentication is used to prevent from learning non-authenticated and invalid routes and advertising valid routes to non-authenticated routers. In a broadcast network, the area authentication can also prevent the non-authenticated routers from becoming the designated routers, therefore improving the stability and intrusion prevention capability of the routing system.

When an area is an OSPF leaf area, that is, the area neither acts as a transit area nor injects external routes to the OSPF area, you can configure the area as a stub area. The routers in a stub area can only learn about three routes: 1) Routes in the stub area, 2) Routers in other areas, and 3) Default routes advertised by the stub area border router. There are few external routes, so the size of the routing tables of the routers in the stub area is small, and fewer router resources are used. The routers in the stub area may be low- or middle-level routers. To further reduce the number of the LSAs sent to the stub area, you can configure the area as a totally stub area (configured with the **no-summary** option). The routers in the totally stub area can learn two types of routes: 1) Routes in the totally stub area; 2) Default routes advertised by the border router in the totally stub area. After the totally stub area is configured, the router resources occupied by the OSPF are minimized, therefore improving the network transmission efficiency.

If the routers in a stub area can learn multiple default routes, you need to set the costs for these default routes by using the **area default-cost** command, so that the routers in the stub area use the specified default routes with priority.

Pay attention to the following aspects when configuring a stub area:

- A backbone area cannot be configured as a stub area, and the stub area cannot be used as the transmission area of virtual links.
- There is no ASBR in the stub area. In other words, the routes outside an autonomous system cannot be transmitted in this area.
- To set an area as the STUB area, configure all routers in this area with the same attribute.

Use the **no area *area-id*** command to remove the configurations of the specified OSPF area and delete the area, including deleting the area-based configuration commands such as **area authentication**, **area default-cost**, **area filter-list**, **area stub**, and **area nssa**. However, the user cannot remove the OSPF area configurations in the following circumstances:

3) The user needs to remove all configurations of the backbone area, however, configurations of virtual links exist. In this case, the user can delete the backbone area only when the configurations of virtual links are removed.

4) The corresponding **network area** command exists in any area. In this case, the user can delete the area only when all network segment commands added in this area are removed.

Use the following commands to configure the OSPF area parameters in routing process configuration mode.

Command	**Function**
Qtech (config-router)#**area** *area-id* **authentication**	Sets plain-text authentication for the area.
Qtech (config-router)#**area** *area-id* **authentication message-digest**	Sets MD5 authentication for the area.
Qtech (config-router)#area *area-id* **stub** [**no-summary**]	Sets the area as a stubby area. **no-summary**: Sets the area as a stubby area to prevent the ABR in a stub area from sending summary-LSA information to the stub area.
Qtech (config-router)#**area** *area-id* **default-cost** *cost*	Configures the cost of the default route sent to the stub area.

Note

When configuring the authentication, you need to configure the authentication parameters on an interface. For more details, see the "Configuring OSPF Interface Parameters" section. You must configure all routers in the area to have the same configuration with the stub area. To configure a totally stub area, you also have to configure the totally stub area parameters on the border routers in the stub area in addition to the basic configuration of the stub area, and you do not need to change the configurations of other routers.

5.2.5 Configuring an OSPF NSSA

The NSSA is an expansion of the OSPF stub area. In the NSSA, the consumption of router resources is reduced by preventing the type-5 LSAs (AS-external-LSA) from flooding to the NSSA. However, unlike the stub area, the NSSA can inject some routing information outside the autonomous system to the OSPF routing area.

Through the route redistribution, the external AS routes (type-7) are allowed to import to the NSSA. These external type-7 LSAs are converted into the type-5 LSAs on an area border router in the NSSA and flooded to the entire autonomous system. In this process, the external routes are aggregated and filtered.

Pay attention to the following aspects when configuring the NSSA:

- A backbone area cannot be configured as an NSSA, and the NSSA cannot be used as the transmission area of virtual links.
- To set an area as the NSSA, configure all routers connected to the NSSA with the NSSA attribute by using the **area nssa** command.

Use the following commands to configure an area as the NSSA area in routing process configuration mode.

Command	Function
Qtech (config-router)# area *area-id* **nssa** [**no-redistribution**] [**no-summary**] [**default-information-originate[metric** *metric*][**metric-type** [**1** | **2**]]] [**translator** [**stability-interval** *seconds* | **always**]]	(Optional) Defines an NSSA area.
Qtech (config-router)# **area** *area-id* **default-cost** *cost*	Configures the cost of the default route sent to the NSSA area.

Use the *default-information-originate* parameter to generate the default Type-7 LSA. This option varies slightly between the ARR and ASBR in the NSSA. On the ABR, whether the routing table contains a default route or not, the default Type-7 LSA route is generated. On the ASBR, the default Type-7 LSA route is generated only when the routing table of the ASBR contains a default route.

If the *no-redistribution* parameter is configured on the ASBR, other external routes introduced by using the **redistribute** commands are not allowed to be distributed to the NSSA. This option is usually used when the router in the NSSA is both an ASBR and an ABR. This option can also prevent the external routing information from entering the NSSA.

To further reduce the number of the LSAs sent to the NSSA, you can configure the *no-summary* parameter on the ABR to prevent the ABR from sending the aggregated LSAs (Type-3 LSAs) to the NSSA.

In addition, the *area default-cost* parameter is used on the ABR/ASBR connected to the NSSA. This option is used to configure the cost of the default route sent by the ABR/ASBR to the NSSA. By default, the cost value of this default route is 1.

If two or more than two ABRs exist in an NSSA area, the ABR with the largest ID is selected as the translator to translate the Type-7 LSAs to the Type-5 LSAs by default. You can use the *translator always* parameter to configure the current router as the permanent translator ABR.

If the translator role is acted by other ABRs, the current router keeps the capability within the **stability-interval** time. If the router is not configured as the translator again within this period, after the **stability-interval** time expires, LSAs translated from Type-7 to Type-5 will be removed from the AS.

Note

The Type-5 LSAs aggregated and translated from the Type-7 LSAs are removed immediately after the current router becomes translator-disabled without waiting for the **stability-interval** timeout to prevent routing loop.

In the same NSSA area, it is recommended to configure the *translator always* parameter for only one ABR.

5.2.6 Configuring OSPF Route Aggregation

5.2.6.1 Configuring the Route Aggregation between Areas

An area border router (ABR) has at least two interfaces that belong to different areas, one of which must be a backbone area. The ABR acts as the pivot in the OSPF routing area. It can advertise the routes of one area to another area. If the network addresses of the routes of this area are continual, the ABR can advertise only one aggregated route to other areas. The route aggregation function between areas greatly reduces the size of the routing table and improves the network efficiency.

Use the following command to configure the route aggregation between areas in routing process configuration mode.

Command	Function
Qtech (config-router)# **area** *area-id* **range** *ip-address mask* [**advertise** | **not-advertise**] [**cost** *cost*]	Configures the route aggregation between areas.

Note

If the route aggregation is configured, the ABR does not advertise the detailed routes in this area to other areas.

5.2.6.2 Configuring the External Route Aggregation

When routes are redistributed in other routing processes and imported into the OSPF routing process, every route is advertised to the OSPF-enabled router as a separate link. If the injected routes have continuous IP addresses, an ASBR can advertise only one aggregated route, therefore reducing the size of the routing table significantly.

Use the following command to configure the external route aggregation in routing process configuration mode.

Command	Function
Qtech (config-router)# **summary-address** *ip-address mask* [**not-advertise** | **tag** *tag-id*]	Configures the external route aggregation.

5.2.6.3 Controlling the Aggregated Routes to Be Added to the Routing Table

The network range after the route aggregation may exceed the original network range in the routing table. If data are sent to the network beyond the aggregation range, routing loop may incur or load on the router may increase. Therefore, add a discard route to the routing table of the ABR or ASBR to prevent that problem.

Use the following commands to allow or forbid adding the discarded routes to the routing table in routing process configuration mode.

Command	Function
Qtech (config-router)# **discard-route** {**internal** | **external**}	Allows adding the discarded routes to the routing table.
Qtech (config-router)# **no discard-route** {**internal** | **external**}	Forbids adding the discarded routes to the routing table.

By default, adding the discarded routes to the routing table is allowed.

5.2.7 Creating a Virtual Link

In an OSPF routing area, the OSPF route update between non-backbone areas is performed by using the backbone area. All non-backbone areas are connected to the backbone area. If the backbone area is disconnected with the non-backbone areas, you need to configure virtual links to connect the non-backbone areas to the backbone area. Otherwise, the network communication fails. Create virtual links for the connections when physical links cannot meet the requirements due to the limitation of the network topology.

A virtual link can be created between two ABRs. The common area that the two ABRs belong to is a transit area. A stub area and NSSA area cannot be used as the transit area. The virtual link can be seen as a logical connection channel established between the two ABRs via the transit area. On both ends of the virtual link deploy ABRs and configuration on both ends must be performed synchronously. The virtual link is identified with the router ID of the peer router. The area that provides the two ends of the virtual link with an internal non-backbone area route is called the transit area, whose number must be specified during the configuration.

The virtual link will be activated after the route in the transit area has been calculated (that is, the route to the peer router). You can see it as a point-to-point link, on which most parameters of the interface, such as the *hello-interval* and *dead-interval* parameters, can be configured like a physical interface.

A logical channel means that multiple routers running the OSPF routing protocol between the two ABRs. The logical channel is only used to forward packets. (Since the destination of the protocol packets are not these routers, the packets are transparent to them and are simply forwarded as common IP packets.) The two ABRs exchange routing information directly, and the synchronization mode in the area is not changed. The routing information means the Type-3 LSAs generated by the ABR.

Use the following command to create a virtual link in routing process configuration mode.

Command	Function
Qtech (config-router)# **area** *area-id* **virtual-link** *router-id* [[**hello-interval** *seconds*]| [**retransmit-interval** *seconds*] |[transmit-delay *seconds*]|[**dead-interval** *seconds*]| [**authentication** [**message-digest** | **null**] |[[**authentication-key** [**0**|**7**] *key* | **message-digest-key** *keyid* **md5** [**0**|**7**] *key*]]]	Creates a virtual link.

Caution

If the autonomous system is divided into more than one area, one of the areas must be the backbone area to which the other areas must be connected directly or logically. Also, the backbone area must be in good connection.

Note

The *router-id* is the ID of an OSPF neighbor router. If you are not sure of the router-id, you can use the **show ip ospf** or **show ip ospf neighbor** command to check it. For information about how to manually configure the router-id, see the "Using the Loopback Interface Address as the Route ID" section.

5.2.8 Generating a Default Route

An ASBR can be forced to generate a default route, which is injected to the OSPF routing area. If a router is forced to generate the default route, it is configured to be the ASBR automatically. However, the ASBR does not automatically generate the default route.

Use the following command to force the ASBR to generate a default route in routing process configuration mode.

Command	Function
Qtech (config-router)# **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** ***type***-*value*] [**route-map** ***map***-*name*]	Generates a default route.

Note

When the stub area is configured, the ABR generates the default route automatically and advertises the default route to all routers within this stub area.

5.2.9 Using the Loopback Interface Address as the Router ID

In an OSPF routing process, the largest interface IP address is always used as the router ID. If the interface is disabled or the IP address does not exist, the router ID must be calculated again and all the routing information is sent to the neighbors.

If the loopback interface address (local loop address) is configured, then in the routing process, the IP address of the loopback interface is used as the router ID. If there are multiple loopback interfaces, the largest IP address is selected as the router ID. The loopback address always exists, therefore improving the stability of the routing table.

Use the following commands to configure the loopback address in global configuration mode.

Command	Function
Qtech (config)# **interface loopback** *1*	Creates the loopback interface.
Qtech (config-if)# **ip address** *ip-address mask*	Configures the loopback IP address.

Note

When the IP address of a common interface is specified as the route identifier in the OSPF routing process, even if the loopback interface is configured, the identifier is not specified once again in the OSPF process.

5.2.10 Changing the Default OSPF Management Distance

The management distance of a route represents the credibility of the route source. The management distance ranges from 0 to 255. The greater this value is, the lower the credibility of the route source is.

The OSPF function supported on Qtech products supports intra-area, inter-area, and external routes, whose management distances are all 110 by default. A route belongs to a same area is called the intra-area route, a route to another area is called the inter-area route, and a route to another routing area (learned through redistribution) is called the external route.

Use the following command to change the OSPF management distance in routing process configuration mode.

Command	Function
Qtech(config-router)# **distance** {*distance* | **ospf** {**intra-area** *distance* | **inter-area** *distance* | **external** *distance*}}	Changes the OSPF management distance.

5.2.11 Configuring the Route Calculation Timer

When receiving a notification about route topology changes in the OSPF routing process, the system runs the SPF algorithm for route calculation after a time delay. You can configure this delay and also configure the minimum interval between two SPF calculations.

Use the following command to configure the OSPF route calculation timer in routing process configuration mode.

Command	Function
Qtech (config-router)# **timers throttle spf** *spf-delay spf-holdtime spf-max-waittime*	Configures the route calculation timer.

Note

The *spf-delay* refers to the delay time from the time when the topology changes to the time when the SPF calculation is performed. The *spf-holdtime* refers to the minimum time interval between two SPF calculations. Later, the time interval of the consecutive SPF calculations shall be at least twice as the last time interval until the time interval reaches the *spf-max-waittime* value*.* If the time interval between two SPF calculations has exceeded the minimum value, then the time interval is recalculated from the *spf-holdtime*.

Normally, reducing the value of *spf-delay* and *spf-holdtime* can speed up the OSPF convergence if the link turbulence occurs occasionally. Increasing the value of the *spf-max-waittime* can avoid the CPU consumption by the OSPF routing process due to the consecutive link turbulence.

For example, timers throttle spf 1000 5,000 100,000

If the topology changes constantly, the time interval for SPF calculations increases in the ascend order as follows when calculated by using the binary exponential backoff algorithm, but this time interval does not exceed the *spf-max-waittime*:

1 second, 6 seconds, 16 seconds, 36 seconds, 76 seconds, 156 seconds, 256 seconds, 256+100 seconds…

Use the following command to configure only the OSPF route calculation delay and hold-time in routing process configuration mode.

Command	Function
Qtech (config-router)# **timers spf** *spf-delay spf-holdtime*	Configures the route calculation timer in seconds.

Caution

The **timers spf** and **timers throttle spf** commands overwrite each other during the configuration. The latter configured command takes effect. If neither of the two commands are configured, the default value is the value configured in the **timers throttle spf** command.

The **timers throttle spf** command is more powerful than the **timers spf** command. Therefore, you are advised to use the **timers throttle spf** command.

5.2.12 Changing the LSA Group Pacing Timer

Each LSA has its own refresh and aging time. Calculating the refresh and aging time for each LSA respectively consumes lots of CPU. To make full use of the CPU, perform the refresh and aging operations for the LSAs uniformly.

The default refresh and aging time is 4 minutes. You do not need to adjust this parameter often. The optimum group pacing interval is inversely proportional to the number of LSAs that need to be calculated. For example, if approximately 10,000 LSAs exist in the database, decreasing the pacing interval would be better. If only 40 to 100 LSAs exist in the database, increasing the pacing interval to 10 to 20 minutes might be better.

Use the following commands in routing process configuration mode.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **timers pacing lsa-group** *seconds*	(Optional) Changes the LSA group pacing.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves the configurations.

To restore the default value, use the **no timers pacing lsa-group** in global configuration mode.

5.2.13 Configuring the Cost for the OSPF Interface

The OSPF system calculates the destination route based on cost. The route with the least cost is the shortest route. The default route cost is based on network bandwidth. When you configure the OSPF-enabled router, you can set the link cost according to the factors such as link bandwidth, time delay or economic cost. The lower the link cost is, the higher the possibility is for the link to be selected as the route. If route aggregation is enabled, the maximum cost of all the aggregated links are used as the cost of the aggregated information.

Routing configuration includes two steps. First, specify a reference value for the generated cost based on the bandwidth. This value and the interface bandwidth are used to calculate the default cost. Second, set the cost for each interface by using the **ip ospf cost** command. In this case, the default cost takes no effective on the interface. For example, if the default reference value is 100 Mbit/s, and the bandwidth of an Ethernet interface is 10 Mbit/s, the default cost of this interface is 100/10 + 0.5 ≈ 10.

The interface cost is selected in the following way according to the OSPF protocol: The cost of the interface specified by the user has the highest priority. If you have specified the interface cost, the cost is used as the interface cost. If you do not specify the interface cost but the automatic cost generation function is enabled, the automatically calculated value is used. If the automatic cost generation function is disabled, the default value 10 is used.

Use the following commands to perform the configuration.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **auto-cost reference-bandwidth** *ref-bw*	(Optional) Sets the default cost based on the bandwidth on an interface. The cost value is determined based on the *ref-bw* parameter.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show ip protocols**	Displays the routing protocol that is running currently.
Qtech # **write**	(Optional) Saves the configurations.

To remove the setting, use the **no ip ospf cost** and **auto-cost** command.

5.2.14 Configuring an OSPF Stub Router

A router that only forwards packets to its directly-connected links is called a stub router. To prevent low-level routers from handling massive LSAs or to enable the routers to smoothly join/exit a network, you can configure such routers as stub routers. The stub router can advertise its maximum cost so that other routers will not preferentially use this router as a transit node during SPF calculation.

After the **max-metric router-lsa** command is enabled, the metric of non-stub links carried in the Router LSA generated by the router will be used as the maximum value (0xFFFF). After the user removes the setting or the timer expires, the default metric of the links is restored.

By default, after this command is enabled, the ordinary metric of stub links is advertised, namely the cost of the egress. If the *include-stub* parameter is configured, the maximum metric of the stub links is advertised.

If you do not want to transmit the data in an area, use the *summary-lsa* parameter to configure the summary LSA as the maximum metric for the ABR.

If you do not want to transmit the data in an external area, use the *external-lsa* parameter to configure the external LSA as the maximum metric for the ASBR.

The **max-metric router-lsa** command is generally used in the following circumstances:

- Restart the router. After the router is restarted, the IGP protocol is converged more quickly, and other routers may try to forward the data through the restarted router. If the router is still building BGP routing tables and certain BGP routes have not be learned, packets sent to such router will be discarded. In this case, use the on-startup parameter to configure a delay timer, so that the restarted router can act as the transit node after the timer runs out.

- Connect the router to the network without using the router to transmit the packets. If alternative paths exist, the current router will not be used to transmit the packets. If no alternative path exists, the current router will still be used to transmit the packets.

- Gracefully remove the router from the network. By using this command, the current router can advertise a maximum metric value, so that other routers on the network will select the alternative paths to transmit the packets before the router is shut down.

Use the following commands to configure a router to advertise a maximum metric in routing process configuration mode.

Command	Function

Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** [*seconds*]] [**summary-lsa** [*max-metric-value*]]	(Optional) Configures the router to advertise a maximum metric.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show ip protocols**	Displays the routing protocol that is running currently.
Qtech # **write**	(Optional) Saves the configurations.

⚠ Caution

In the earlier versions of the OSPF (RFC 1247 or earlier versions), links with the maximum metric (0xFFFF) in the LSAs are not involved in the SPF calculation, that is, no packet is sent to routers generating these LSAs.

5.2.15 Configuring Whether to Perform the MTU Check on an Interface

When the OSPF system receives database description packets, it will check whether the MTU of a neighbor interface is the same as its own MTU. If the MTU of the interface indicated in the received database description packets is greater than that of the receiving interface, the adjacency relationship cannot be established. In this case, you can disable the MTU check function.

Use the following command to disable the MTU check on an interface in interface configuration mode.

Command	Function
Qtech (config-if)# **ip ospf mtu-ignore**	Disables the MTU check on the interface when the interface receives the database description packets.

The MTU check on an interface is disabled by default.

5.2.16 Disabling an Interface to Send the OSPF Packets

To prevent other routers on the network from dynamically learning the routing information of the local router, you can use the **passive-interface** command to set the specified network interface of the local router as a passive interface to prevent the local router from sending the OSPF packets.

Use the following commands to configure an interface as a passive interface in privileged EXEC mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **passive-interface** *interface-name*	(Optional) Sets the specified interface as a passive interface.
Qtech(config-router)# **passive-interface default**	(Optional) Sets all network interfaces as the passive interfaces
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

By default, all interfaces are allowed to receive and send the OSPF packets. To re-enable the network interface to send the routing information, use the **no passive-interface *interface-id*** command. To re-enable all network interfaces, use the keyword **default.**

5.2.17 Configuring Whether to Perform the Source Address Check

According to the OSPF requirements, the source address of the received packets must be in the same network segment with the address of the interface that receives the packets. However, for a point-to-point link, the addresses of the two link ends are configured independently, so the addresses are not required to be in the same network segment. In the negotiation process of a point-to-point link, the address information about the peer end is advertised. Therefore, the OSPF system will check whether the source address of the packets is the address advertised by the peer end during the negotiation. If the two addresses are not consistent, the system will treat the packets as unauthorized packets and discard the packets. The negotiated address may be shielded in some applications. In this case, disable this source address check function to establish the OSPF adjacency relationship properly. In particular, this function is disabled on the unnumbered interface all the time.

Use the following command to configure whether to perform the source address check on a point-to-point link in interface configuration mode.

Command	Function
Qtech (config-if)# **ip ospf source-check-ignore**	Disables the source address check on the point-to-point link.

The source address check on a point-to-point link is enabled by default.

5.2.18 Configuring the OSPF Fast Convergence Function

5.2.18.1 Configuring the OSPF Fast Hello

The OSPF Fast Hello function facilitates fast discovery of OSPF neighbors and supports quick detection of lost OSPF neighbors. The OSPF Fast Hello function is enabled by specifying the **minimal** and **hello-multiplier** keywords and the *multiplier* parameter. The **Minimal** keyword is used to set the dead interval to 1 second, and the **hello-multiplier** keyword is used to configure the times for sending Hello packets during the dead interval, therefore the interval for sending the Hello packets is reduced to less than 1 second.

When the Fast Hello function is enabled on the interface, the **Hello interval** field for the interface sending the Hello packets is set to 0. The **Hello interval** field for the interface receiving the Hello packets is ignored.

No matter whether the Fast Hello function is enabled or not, the dead interval must be consistent on a same network segment. However, the *hello-multiplier* parameter is not required to be consistent on a same network segment as long as at least one Hello packet is received within the dead interval.

Use the following commands to configure the Fast Hello on an interface.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **ip routing**	Enables the IP routing function (if disabled).
Qtech(config)# **interface** *interface-id*	Enters interface configuration mode.
Qtech(config-if)# **ip ospf dead-interval minimal hello-multiplie**r *multiplier*	(Optional) Enables the OSPF Fast Hello on the interface.
Qtech (config-if)# **end**	Returns to privileged EXEC mode.
Qtech # **show ip ospf** [*process-id*]**interface** [*interface-id*]	Displays the OSPF interface information.
Qtech # **write**	Saves the configurations.

Use the following commands to configure the Fast Hello on a virtual link.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech (config)# **ip routing**	Enables the IP routing function (if disabled).
Qtech (config)# **router ospf** *process_id* [**vrf** *vrf-name*]	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **area** *area-id* **virtual-link** *router-id* [**dead-interval** | **minimal hello-multiplier** *multiplier*]	Enables the Fast Hello on the virtual link
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **write**	Saves the configurations.

⚠ Caution

You cannot configure the *dead-interval minimal hello-multiplier* parameter and the *hello-interval* parameter at the same time.

5.2.18.2 Configuring the OSPF Two-Way Maintenance

On a large scale network, a large number of packets may be received and transmitted, which occupies high CPU and memory resources, therefore causing the delay or drop of certain packets. If the time for processing the Hello packets goes beyond the dead interval, the corresponding adjacent routers will be disconnected. In this case, enable the OSPF two-way maintenance function. If a large number of packets exist on the network, besides the Hello packets, the DD, LSU, LSR and LSAck packets from a certain neighbor can also be used to maintain the two-way adjacency relationship, therefore avoiding the disconnection of neighbors caused by the delay or drop of the Hello packets.

The OSPF two-way maintenance function is enabled by default. Use the following commands to disable the OSPF two-way maintenance function in routing process configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **no two-way-maintain**	(Optional) Disables the OSPF two-way maintenance function.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

5.2.18.3 Configuring the Interval of Receiving the Same LSA

On a broadcast network or in the environment featured by frequent network oscillation, the router may receive the same LSA updates from one or multiple interfaces and different neighbors. If the same LSAs are processed every time, excessive system resources are wasted. According to the OSPF protocol, the same LSAs are considered to be valid after a period of time. The same LSAs received within a short period of time will be ignored. This time interval is the constant MinLSArrival with the value set to 1 second.

Different types of networks have different requirements on the interval for processing LSA changes. The user can configure this parameter according to different network planning and performance requirements to optimize the network.

Use the following commands to configure the interval of receiving the same LSA in routing process configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **timers lsa arrival** ***arrival***-*time*	(Optional) Configures the interval of receiving the same LSA. The default value is 1000 milliseconds.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

5.2.18.4 Configuring to Send the LSA Packet Updates

To relieve the impacts on network devices caused by the flooding of a large number of update packets, the LSP packet update function is introduced. By specifying the delay interval for the update packets, the LSAs to be flooded during the interval can be collected, so that these LSAs can be sent with the least number of packets. Meanwhile, the CPU can process other tasks and the system performance is optimized.

When a large number of LSAs exist on the network and the router loads excessively, you need to configure the **transmit-time** and **transmit-count** commands properly to control the number of LS-UPD packets flooded on the network. When the load of CPU is low and the network bandwidth is small, you can reduce the transmit-time value and increase the transmit-count value to speed up the network convergence.

Use the following commands to configure the LSA to send the packet updates in routing process configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **timers pacing lsa-transmit** *transmit-time transmit-count*	(Optional) Configures the LSA to send the packet updates.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

5.2.18.5 Configuring the Exponential Backoff Algorithm for Generating LSAs

To prevent multiple events from triggering the same LSAs within a short time, causing frequent LSA updates, and consuming excess CPU resources, you can specify the minimum time interval "MinLSInterval" for generating the LSAs according to the OSPF routing protocol. The default minimum time interval is 5 seconds. During this time interval, the same LSA instances cannot be generated repeatedly, therefore preventing frequent LSA oscillation from causing impacts on the network. However, this configuration slows down the LSA generation speed and fails to advertise the network topology changes immediately.

To quickly respond to the network topology changes and avoid excessively frequent route calculations, use the exponential backoff algorithm to dynamically change the time interval for generating the LSAs. The **timers throttle lsa all** command has three parameters: *delay-time*, *hold-time*, and *max-wait-time*, which allow the system to automatically adjust the time interval for generating the LSAs according to the frequency of the network topology changes. Generally, *delay-time* is set to a small value or 0 to trigger LSA instances immediately when the network topology is comparatively stable. When the network topology changes frequently, the time interval for generating the LSAs increases from the *hold-time* and follows the algorithm of *hold-time* x 2^{n-1}. n refers to the times of changes. With the times for generating LSAs repeatedly increasing, the time interval for generating the LSAs becomes greater and greater until the *max-wait-time* is reached. When the time interval for generating the LSAs is greater than the *max-wait-time*, the *delay-time* for generating the LSAs restores to the initial value.

By default, the initial value is 0 milliseconds, the *hold-time* is 5000 milliseconds, and the *max-wait-time* is 5000 milliseconds. The shortest interval for consecutively generating the same LSA is the *MinLSInterval*, which complies with the rules defined in RFC 2328.

Use the following commands to configure the exponential backoff algorithm for generating the LSAs in routing process configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **timers throttle lsa all** *delay-time hold-time max-wait-time*	(Optional) Configures the exponential backoff algorithm for generating the LSAs. By default, the initial value is 0 milliseconds, the *hold-time* is 5000 milliseconds, and the *max-wait-time* is 5000 milliseconds.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

Caution

During the configuration, the *hold-time* cannot be less than the *delay-time*, and the *max-wait-time* cannot be less than the *hold-time*.

5.2.19 Configuring the OSPF Capacity Protection Function

When the memory lacks, the OSPF system enters the overflow state. In the overflow state, the OSPF protocol triggers the following operations:

- For the learned LSAs: receive the inter-area and intra-area LSAs, and only receive the external LSAs indicating that the route to the destination address is a specific non-default route.
- For the external LSAs generated by itself: clear the external LSAs except for the LSAs indicating the default routes.
- The incompleteness of route learning and advertisement may lead to the routing loop on the network. The OSPF system generates a default route to the NULL interface to prevent the routing loop. The generated default route exists in the overflow state all the time.

Use the following commands to configure the OSPF router to enter the overflow state when the memory lacks.

Command	Function
Qtech(config)#**router ospf** *process-id*	Enters OSPF configuration mode.
Qtech(config-router)#**overflow memory-lack**	Configures the OSPF system to enter the overflow state when the memory lacks.

Note

By default, the OSPF system enters the overflow state automatically when the memory lacks. You can use the **no overflow memory-lack** command to disable this function.

Caution

You must use the **clear ip ospf process** command or restart the OSPF protocol to exit from the overflow state.

5.2.20 Configuring the OSPF Network Management Function

5.2.20.1 Configuring the OSPFv2 MIB Binding

The user can only operate a sole OSPFv2 process by SNMP since the OSPFv2 MIB does not have the OSPFv2 process information. By default, the OSPFv2 MIB is bound to the OSPFv2 process with the smallest scale, and this process takes effect over all user operations.

The user can bind the OSPFv2 MIB to the process manually to operate the specified OSPFv2 process by using SNMP.

Use the following commands in routing process configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **enable mib-binding**	(Optional) Binds the OSPFv2 MIB to the specified OSPFv2 process.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

5.2.20.2 Configuring the OSPFv2 TRAP Binding

The OSPFv2 protocol defines several types of OSPF TRAP information, which is used to report various events about the OSPFv2 protocol. Sending the OSPFv2 TRAP information is not restricted by the MIB binding to the OSPFv2 process. The TRAP switch is allowed to be enabled for different processes at the same time.

Use the following commands in global configuration mode.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **snmp-server enable traps ospf**	(Optional) Enables the OSPF TRAP sending switch.
Qtech(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Qtech(config-router)# **enable traps [error [IfAuthFailure | IfConfigError | IfRxBadPacket | VirtIfAuthFailure | VirtIfConfigError | VirtIfRxBadPacket] | lsa [LsdbApproachOverflow | LsdbOverflow | MaxAgeLsa | OriginateLsa] | retransmit [IfTxRetransmit | VirtIfTxRetransmit] | state-change [IfStateChange | NbrRestartHelperStatusChange | NbrStateChange | NssaTranslatorStatusChange | RestartStatusChange | VirtIfStateChange | VirtNbrRestartHelperStatusChange | VirtNbrStateChange]]**	(Optional) Enables the specified OSPF TRAP switch.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **write**	Saves the configurations.

5.2.21 Configuring the OSPF GR Function

The graceful restart (GR) function is used to enable data packets to be forwarded continuously during the restart process of the OSPF protocol. Currently, the GR function is supported on the switchover between our primary and secondary high-end devices to ensure that the key service is not interrupted.

5.2.21.1 Working Principles of the OSPF GR

OSPF GR standard:

RFC3623: Graceful OSPF Restart

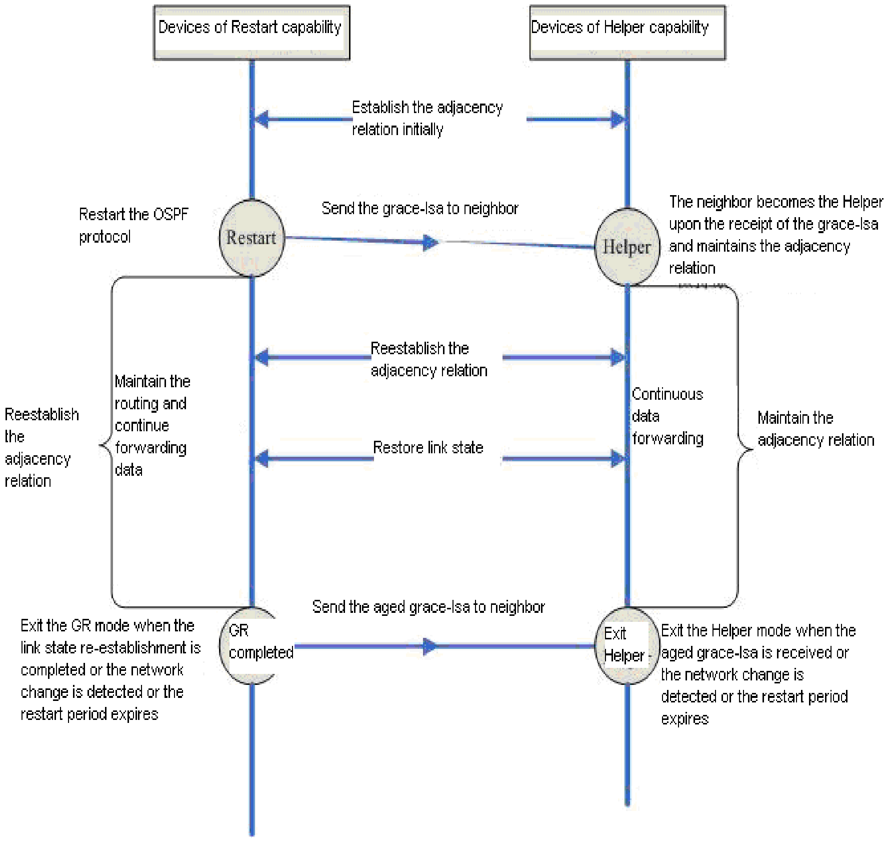
Working principles of RFC3623:

As a standard GR protocol defined by the IETF for the OSPF, RFC3623 defines the conditions, operations and precautions required for executing the Graceful Restart. As specified in RFC3623, two GR principles are important. Namely, the network topology should be stable and the router for restarting the protocols can maintain the forwarding table during the restarting process.

The execution of OSPF GR is not an independent process. The OSPF GR has the GR Restart and GR Help functions. The device with the GR Restart capability can automatically perform the graceful restart operation, and the device with the GR Help capability can receive Grace_LSAs and help the neighbors to perform the graceful restart operation.

Generally, the device that has the GR Restart capability and is performing the GR operation is called the GR Restarter. The device that has the GR Help capability and is helping the GR Restarter to perform the GR operation is called the GR Helper. The GR process begins from the operation where the GR Restarter sends a Grace LSA. The neighbor becomes the GR Helper upon receiving the Grace LSA and assists the GR Restarter to reestablish the adjacency relationship. Meanwhile, the neighbor maintains the adjacency relationship with the GR Restarter for continuous data forwarding.

OSPF GR execution flowchart


Devices of Restart capability
Devices of Helper capability
Establish the adjacency relation initially
Restart the OSPF protocol
Restart
Send the grace-lsa to neighbor
Helper
The neighbor becomes the Helper upon the receipt of the grace-lsa and maintains the adjacency relation
Reestablish the adjacency relation
Maintain the routing and continue forwarding data
Reestablish the adjacency relation
Restore link state
Continuous data forwarding
Maintain the adjacency relation
Send the aged grace-lsa to neighbor
GR completed
Exit Helper
Exit the GR mode when the link state re-establishment is completed or the network change is detected or the restart period expires
Exit the Helper mode when the aged grace-lsa is received or the network change is detected or the restart period expires

The above figure outlines the execution process of the OSPF GR. The GR period is the longest time for reestablishing the link status. When the period for the link reestablishment or the graceful restart expires, the GR Restarter exits the GR operation.

5.2.21.2 Configuring the OSPF GR Restarter

Use the **graceful-restart** command to enable the OSPF GR restarter.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.

Qtech (config-router)# **graceful-restart**	Enables the OSPF GR restarter.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves the configurations.

By default, the GR restarting period is 120 seconds. Use the **graceful-restart grace-period** command to modify the restarting period.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **graceful-restart** grace-period *100*	Enables the OSPF GR restarter and sets the GR restarting period to 100 seconds.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves the configurations.

Note The routers do not support this function.

5.2.21.3 Configuring the OSPF GR Helper

The OSPF GR Helper is enabled by default. The software provides the functions to disable the GR Helper and configure the GR Helper to detect the network changes. The following example shows how to disable and re-enable the GR Helper function and how to configure the Helper to detect the network changes.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Qtech (config-router)# **graceful-restart helper disable**	Disables the OSPF GR Helper (for disabling the GR help to neighbors).
Qtech (config-router)# **no graceful-restart helper disable**	Enables the OSPF GR Helper again.
Qtech (config-router)# **graceful-restart helper** {**strict-lsa-checking** | **internal-lsa-checking**}	Enables the OSPF GR Helper to check the LSA changes to detect the network changes. If the network changes, exit the GR Helper. By default, the network changes are not detected after the GR Helper is enabled. **strict-lsa-checking:** checks the changes of types 1 to 5 and type 7 LSAs. **internal-lsa-checking:** checks the changes of types 1 to 3 LSAs.
Qtech (config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves the configurations.

Changes in a part of the network may disable the GR function and cause low convergence performance on the entire network. Therefore, it is not recommended that the user enable the LSA detection function when the network is in a large scale.

5.2.22 Configuring the OSPF BFD Function

For details about the OSPF BFD configuration, see *BFD Configuration Guide*.

5.2.23 Configuring the OSPF VPN Function

For details about the OSPF VPN configuration, see *Configuring the OSPF VPN extension*.

5.2.24 Monitoring and Maintaining the OSPF

The following table shows the data such as the OSPF routing table, cache, and database that can be displayed.

Command	Function
show ip ospf [*process-id*]	Displays the general information about the corresponding processes of the OSPF protocol. All processes are displayed if no process number is specified.
show ip ospf [*process-id area-id*] **database** [**adv-router** *ip-address* | {**asbr-summary** | **external** | **network** | **nssa-external | opaque-area | opaque-as | opaque-link | router** | **summary**} [*link-state-id*] [{**adv-router** *ip-address* | **self-originate**}] | **database-summary** | **max-age** | **self-originate**]	Displays the OSPF database information. You can view the information about each type of LSAs in the specified process.
show ip ospf [*process-id*] **border-routers**	Shows the routing information about the specified process after reaching the ABR and ASBR.
show ip ospf interface [*interface-name*]	Shows the information about the interface involved in the OSPF routing.
show ip ospf [*process-id*] **neighbor**[*interface-name*] [*neighbor-id*] **[detail]**	Shows the information about the adjacent routers of the interface. *interface-name*: local interface connected to the neighbor *neighbor-id*: router ID of the neighbor.
show ip ospf [*process-id*] **virtual-links**	Views the virtual link information about the specified process.
show ip ospf [*process-id*] **route [count]**	Shows the routes in the OSPF routing table.
show ip ospf [*process-id*] **spf**	Shows the times for calculating inter-area routes.

For specific explanations about the commands, see *OSPF Routing Protocol Configuration Command*. The commonly used commands for monitoring and maintenance are described as follows:

5) Show the status of the OSPF neighbors.

Use the **show ip ospf** [*process-id*] **neighbor** command to show all information about neighbors in the OSPF process, including the status, role, router ID, IP address, and BFD state.

```
Qtech# show ip ospf neighbor
OSPF process 1:
Neighbor ID  Pri State BFD State Dead Time    Address:            Interface
10.10.10.50 1 Full/DR  UP 00:00:38            10.10.10.50  eth0/0
OSPF process 100:
Neighbor ID  Pri State BFD State Dead Time    Address:     Interface
10.10.11.50 1      Full/Backup  DOWN 00:00:31       10.10.11.50  eth0/1
Qtech# show ip ospf 1 neighbor
OSPF process 1:
Neighbor ID  Pri State BFD State Dead Time    Address:     Interface
10.10.10.50 1      Full/DR           UP 00:00:38  10.10.10.50  eth0/0
Qtech# show ip ospf 100 neighbor
OSPF process 100:
Neighbor ID  Pri State    BFD State Dead Time       Address:     Interface
10.10.11.50 1Full/Backup  DOWN 00:00:31       10.10.11.50  eth0/1
```

6) Show the status of the OSPF interfaces

According to the following message, the FastEthernet 0/1 interface belongs to Area 0 in the OSPF routing area, the router ID is 192.168.1.1, and the network type is BROADCAST. Pay special attention to the parameters such as *Area*, *Network Type*, *Hello*, and *Dead*. If these parameters are different from the neighbor, no adjacency relationship is established.

```
Qtech# sh ip ospf interface fastEthernet 0/1
FastEthernet 1/0 is up, line protocol is up
Internet Address 192.168.1.1/24, Ifindex: 2 Area 0.0.0.0, MTU 1500
Matching network config: 192.168.1.0/24,
Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 192.168.1.1, Interface Address 192.168.1.1
Backup Designated Router (ID) 192.168.1.2, Interface Address 192.168.1.2
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:04
Neighbor Count is 1, Adjacent neighbor count is 1
```

```
Crypt Sequence Number is 30
Hello received 972 sent 990, DD received 3 sent 4
LS-Req received 1 sent 1, LS-Upd received 10 sent 26
LS-Ack received 25 sent 7, Discarded 0
```

7) Show the information about the OSPF routing process

Run the following command to show the information about the route ID, router type, area information, and area route aggregation.

```
Qtech# show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
Enable two-way-maintain
This router is an ASBR (injecting external routing information)
Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
Pacing lsa-group: 240 secs
Number of incoming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 4. Checksum 0x0278E0
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 4
External LSA database is unlimited.
Number of LSA originated 6
Number of LSA received 2
Log Neighbor Adjacency Changes: Enabled
Graceful-restart disabled
Graceful-restart helper support enabled
Number of areas attached to this router: 1
Area 0 (BACKBONE)
Number of interfaces in this area is 1(1)
Number of fully adjacent neighbors in this area is 1
Area has no authentication
SPF algorithm last executed 00:01:26.640 ago
SPF algorithm executed 4 times
Number of LSA 3. Checksum 0x0204bf
Routing Process "ospf 20" with ID 2.2.2.2
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
Enable two-way-maintain
Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
Pacing lsa-group: 240 secs
Number of incoming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 0. Checksum 0x000000
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 0
External LSA database is unlimited.
Number of LSA originated 0
Number of LSA received 0
Log Neighbor Adjacency Changes: Enabled
Number of areas attached to this router: 0
```

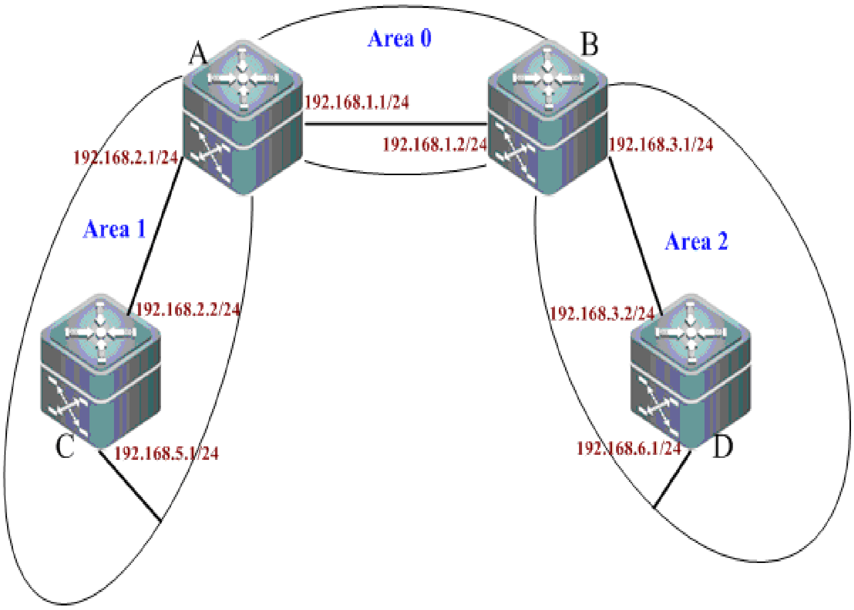
5.3 Configuration Examples

5.3.1 Example of Multi-Area OSPF Configuration

5.3.1.1 Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. Each router runs the OSPF routing protocol.

Networking topology for the multi-area OSPF configuration



5.3.1.2 Applications

Configure Router A and Router B as area border routers (ABR) and Router C and Router D as intra-AS routers. Based on the basic OSPF configurations, every switch can successfully learn the routes in the autonomous system to all network segments.

5.3.1.3 Configuration Tips

- ➢ Configure the IP address for each interface on the routers.
- ➢ Enable the basic OSPF functions.

1. Enable the routing function (enabled by default).
2. Create an OSPF routing process.
3. Specify the IP address range associated with this routing process and the OSPF area to which the IP addresses within the range belong.

5.3.1.4 Configuration Steps

- Configuration steps of A

Step 1: Configure the IP address for the interface.

```
A(config)#interface gigabitEthernet 0/1
A(config-if-GigabitEthernet 0/1)#ip address 192.168.1.1 255.255.255.0
A(config-if-GigabitEthernet 0/1)#exit
A(config)#interface gigabitEthernet 0/2
A(config-if-GigabitEthernet 0/2)#ip address 192.168.2.1 255.255.255.0
A(config-if-GigabitEthernet 0/2)#exit
```

Step 2: Configure the basic OSPF functions.

```
A(config)#router ospf 1
A(config-router)#network 192.168.1.0 0.0.0.255 area 0
A(config-router)#network 192.168.2.0 0.0.0.255 area 1
```

- Configuration steps of B

Step 1: Configure the IP address for the interface.

```
B(config)#interface gigabitEthernet 0/1
B(config-if-GigabitEthernet 0/1)#ip address 192.168.1.2 255.255.255.0
B(config-if-GigabitEthernet 0/1)#exit
B(config)#interface gigabitEthernet 0/2
B(config-if-GigabitEthernet 0/2)#ip address 192.168.3.1 255.255.255.0
B(config-if-GigabitEthernet 0/2)#exit
```

Step 2: Configure the basic OSPF functions.

```
B(config)#router ospf 1
B(config-router)#network 192.168.1.0 0.0.0.255 area 0
B(config-router)#network 192.168.3.0 0.0.0.255 area 2
```

- Configuration steps of C

Step 1: Configure the IP address for the interface.

```
C(config)#interface gigabitEthernet 0/3
C(config-if-GigabitEthernet 0/3)#ip address 192.168.2.2 255.255.255.0
C(config-if-GigabitEthernet 0/3)#exit
C(config)#interface gigabitEthernet 0/4
C(config-if-GigabitEthernet 0/4)#ip address 192.168.5.1 255.255.255.0
C(config-if-GigabitEthernet 0/4)#exit
```

Step 2: Configure the basic OSPF functions.

```
C(config)#router ospf 1
C(config-router)#network 192.168.2.0 0.0.0.255 area 1
C(config-router)#network 192.168.5.0 0.0.0.255 area 1
```

- Configuration steps of D

Step 1: Configure the IP address for the interface.

```
D(config)#interface gigabitEthernet 0/3
D(config-if-GigabitEthernet 0/3)#ip address 192.168.3.2 255.255.255.0
D(config-if-GigabitEthernet 0/3)#exit
D(config)#interface gigabitEthernet 0/4
D(config-if-GigabitEthernet 0/4)#ip address 192.168.6.1 255.255.255.0
D(config-if-GigabitEthernet 0/4)#exit
```

Step 2: Configure the basic OSPF functions.

```
D(config)#router ospf 1
D(config-router)#network 192.168.3.0 0.0.0.255 area 2
D(config-router)#network 192.168.6.0 0.0.0.255 area 2
```

5.3.1.5 Verification

Step 1: Display information about neighbors (taking A and B as the examples).

```
A#show ip ospf neighbor
OSPF process 1, 2 Neighbors, 2 is Full:
Neighbor ID  Pri  State   Dead Time  Address         Interface
192.168.1.2 1  Full/DR  00:00:40 192.168.1.2 GigabitEthernet 0/1
192.168.2.2 1 Full/BDR  00:00:34 192.168.2.2  GigabitEthernet 0/2
B#show ip ospf neighbor
OSPF process 1, 2 Neighbors, 2 is Full:
Neighbor ID Pri  State Dead Time  Address         Interface
192.168.1.1 1 Full/BDR  00:00:32 192.168.1.1  GigabitEthernet 0/1
192.168.3.2 1 Full/BDR  00:00:30 192.168.3.2  GigabitEthernet 0/2
```

Step 2: Display OSPF routing information about A.

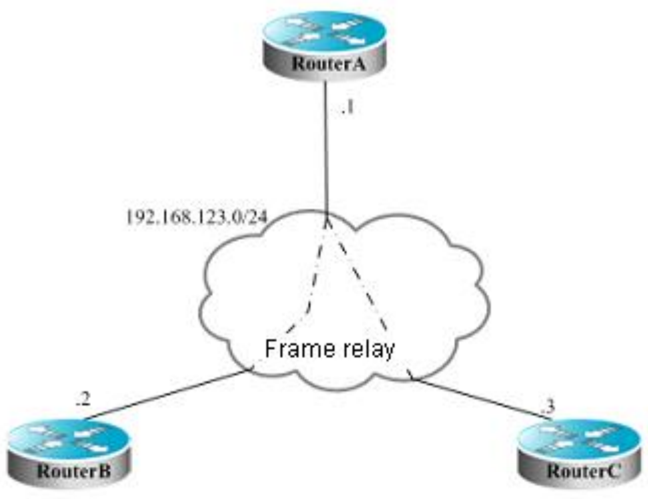
```
A#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.1.1/32 is local host.
C    192.168.2.0/24 is directly connected, GigabitEthernet 0/2
C    192.168.2.1/32 is local host.
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 00:18:03, GigabitEthernet 0/1   //inter-
AS route
O    192.168.5.0/24 [110/2] via 192.168.2.2, 00:00:02, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:01:02, GigabitEthernet 0/1   //inter-
AS route
C#show ip route
Gateway of last resort is no set
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:19:05, GigabitEthernet 0/3   //inter-
AS route
C    192.168.2.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.2.2/32 is local host.
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 00:19:05, GigabitEthernet 0/3
C    192.168.5.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.5.1/32 is local host.
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 00:03:19, GigabitEthernet 0/3   //inter-
AS route
```

5.3.1.6 Example of OSPF NBMA Network Configuration

5.3.1.7 Configuration Requirements

Full mesh connection of three routers can be implemented through a frame relay network. Each router has only one frame relay link and the same link bandwidth and PVC rate. The following figure shows details about the IP address assignment and connections of the three routers.

Networking topology for OSPF NBMA network configuration



Requirements:

- The network among A, B, and C must configured as an NBMA network.
- A is the designated router, and B is the backup designated router.
- All networks are in the same area.
- Topological convergence is quickened.

5.3.1.8 Specific Configurations

Since there is no special configuration about the OSPF, you can detect neighbors in the multicast manner. If the NBMA network has been configured for the interface, the interface does not send OSPF multicast packets. Therefore, you must specify the IP addresses of neighbors. You can configure shorter SPF calculation wait-time to quicken the topological convergence.

Configurations on Router A:

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.1 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
ip ospf priority 10
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
neighbor 192.168.123.2     priority 5
neighbor 192.168.123.3
timers throttle spf 500 1000 10000
```

Configurations on Router B:

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.2 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
ip ospf priority 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
```

```
neighbor 192.168.123.1 priority 10
neighbor 192.168.123.3
timers throttle spf 500 1000 10000
```

Configurations on Router C:

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.3 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
```

# Configure the OSPF routing protocol.

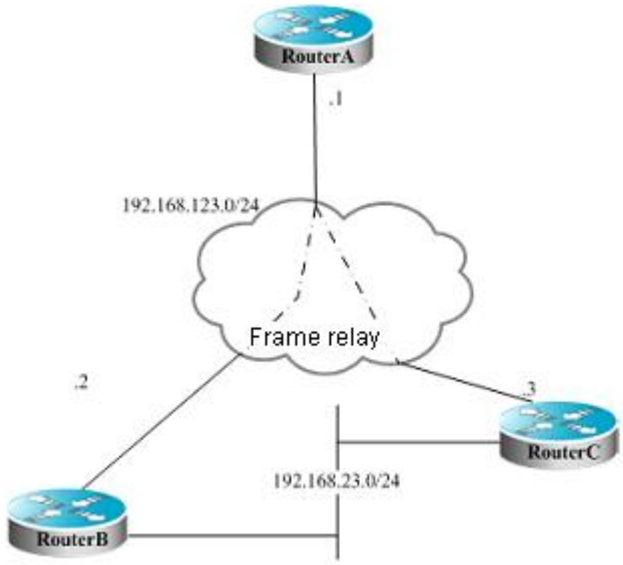
```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
neighbor 192.168.123.1 10
neighbor 192.168.123.2 5
timers throttle spf 500 1000 10000
```

5.3.2 Example of OSPF Point-to-multipoint Broadcasting Network Configuration

5.3.2.1 Configuration Requirements

Interconnection of three routers can be implemented through a frame relay network. Each router has only one frame relay link and the same link bandwidth and PVC rate. The following figure shows details about the IP address assignment and connections of the three routers.

Networking topology for OSPF point-to-multipoint network configuration



Requirements:

- The network among A, B, and C must configured as a point-to-multipoint network.

5.3.2.2 Specific Configurations

The point-to-multipoint network has been configured for the interface. For this network type, there is no need to specify the designated router. The OSPF operations are similar to the steps of configuring the point-to-point network.

Configurations on Router A:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
```

```
ip address 192.168.12.1 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.1 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.23.2 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.2 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Configurations on Router C:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.23.3 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.3 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Assuming that there is another configuration requirement for the above figure:

Router A selects Router B with priority to reach the target network 192.168.23.0/24. To meet the routing requirement, you must specify the cost for a neighbor when configuring the neighbor.

You can execute the following commands on Router A:

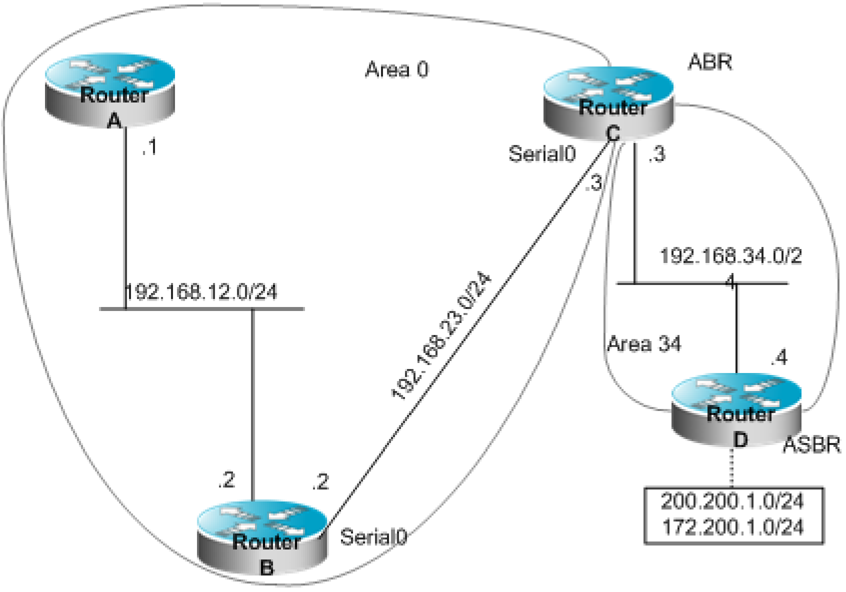
```
router ospf 1
neighbor 192.168.123.2 cost 100
neighbor 192.168.123.3 cost 200
```

5.3.3 Example of OSPF ABR/ASBR Configuration

5.3.3.1 Configuration Requirements

Four routers form an OSPF routing area. The networks 192.168.12.0/24 and 192.168.23.0/24 belong to Area 0, and the network 192.168.34.0/24 belongs to Area 34. The following figure shows details about the IP address assignment and router connection.

Networking topology for OSPF ABR/ASBR configuration



As shown in the figure, Router A and Router B are intra-area routers. Router C is an area border router. Router D is an AS boundary router. 200.200.1.0/24 and 172.200.1.0/24 are network segments outside the OSPF routing area. All OSPF routers shall be able to learn external routes after configuration. External routes shall be type 1 routes and carry tag 34.

5.3.3.2 Specific Configurations

While the OSPF redistributes routes of other sources, the routes to be redistributed are type-II routes and carry no tag by default.

Configurations on Router A:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.12.2 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.23.2 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
network 192.168.23.0 0.0.0.255 area 0
```

Configurations on Router C:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.34.3 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.23.3 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.34.0 0.0.0.255 area 34
```

Configurations on Router D:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.34.4 255.255.255.0
```

# Configure interfaces on the Ethernet adapter.

```
interface FastEthernet 0/1
ip address 200.200.1.1 255.255.255.0
interface FastEthernet 0/2
ip address 172.200.1.1 255.255.255.0
```

# Configure the OSPF routing protocol and redistribute the RIP routes.

```
router ospf 1
network 192.168.34.0 0.0.0.255 area 34
redistribute rip metric-type 1 subnets tag 34
```

# Configure the RIP routing protocol.

```
router rip
network 200.200.1.0
network 172.200.0.0
```

The OSPF routes generated on Router B are shown as follows: (Note that the type of external routes has changed to E1.)

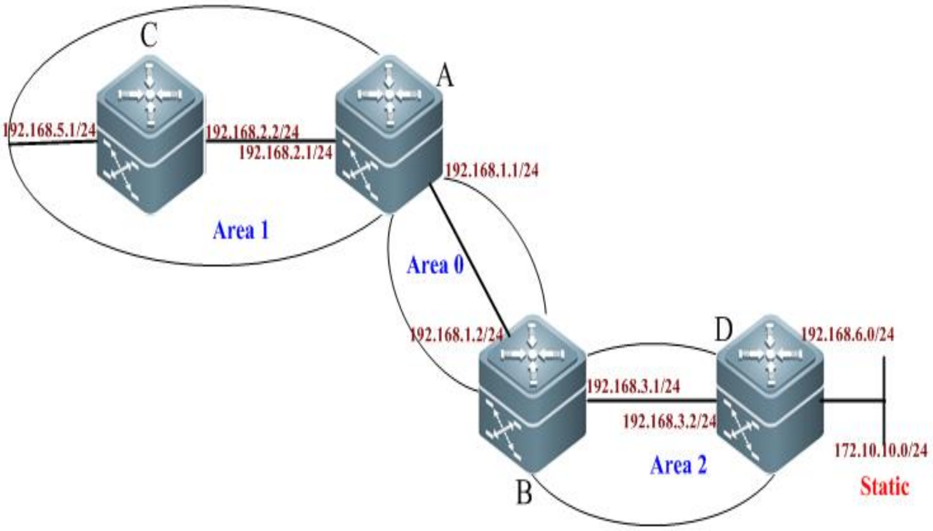
```
O E1 200.200.1.0/24 [110/85] via 192.168.23.3,00:00:33,Serial 1/0
O IA 192.168.34.0/24 [110/65] via 192.168.23.3,00:00:33,Serial 1/0
O E1 172.200.1.0 [110/85] via 192.168.23.3,00:00:33,Serial 1/0
```

5.3.4 Example of OSPF Static Route Redistribution Configuration

5.3.5 Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segment 172.10.10.0 is outside the routing area.

Networking topology for OSPF static route redistribution configuration



5.3.5.1 Applications

Configure Router A and Router B as area border routers (ABR) and Router C as an intra-area router. Configure Router D as an ASBR and introduce an external static route, so that all OSPF routers in non-stub area can successfully learn this external route.

5.3.5.2 Configuration Tips

- Configure the IP address for interfaces on the routers (omitted).
- Configure the basic OSPF functions (see "Example of Multi-Area OSPF Configuration")
- Introduce and configure the external static route.

5.3.5.3 Configuration Steps

Step 1: On Router D, configure a static route to the network segment 172.10.10.0.

```
D(config)#ip route 172.10.10.0 255.255.255.0 192.168.6.2
```

Step 2: Display the routing table of Router A.

```
A#show ip route ospf
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 15:33:00, GigabitEthernet 0/1
O    192.168.5.0/24 [110/2] via 192.168.2.2, 15:14:59, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:17:58, GigabitEthernet 0/1
```

In this case, there is no route to the network segment 172.10.10.0.

Step 3: Redistribute the static route on Router D

```
D(config)#router ospf 1
D(config-router)# redistribute static subnets
```

5.3.5.4 Verification

Step 1: Display the routing table of Router D.

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default
```

```
Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 15:25:19, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 15:25:19, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 15:11:56, GigabitEthernet 0/3
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

Step 2: View OSPF information about Router D. Key point: Router D is an AS boundary router (ASBR).

```
D#show ip ospf
 Routing Process "ospf 1" with ID 192.168.3.2
 Process uptime is 15 hours 27 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Enable two-way-maintain
 This router is an ASBR (injecting external routing information)
 Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
 Pacing lsa-group: 240 secs
 Number of incoming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 1. Checksum 0x006DB0
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 1
 External LSA database is unlimited.
 Number of LSA originated 2
 Number of LSA received 173
 Log Neighbor Adjacency Changes: Enabled
  Number of areas attached to this router: 1: 1 normal 0 stub 0 nssa
    Area 2
        Number of interfaces in this area is 2(2)
        Number of fully adjacent neighbors in this area is 1
        Number of fully adjacent virtual neighbors through this area is 0
        Area has no authentication
        SPF algorithm last executed 00:06:27.540 ago
        SPF algorithm executed 9 times
        Number of LSA 6. Checksum 0x0212ff
```

Step 3: Display the routing table of Router A.

```
A#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.1.2, 00:07:37, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 15:33:00, GigabitEthernet 0/1
O    192.168.5.0/24 [110/2] via 192.168.2.2, 15:14:59, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:17:58, GigabitEthernet 0/1
```

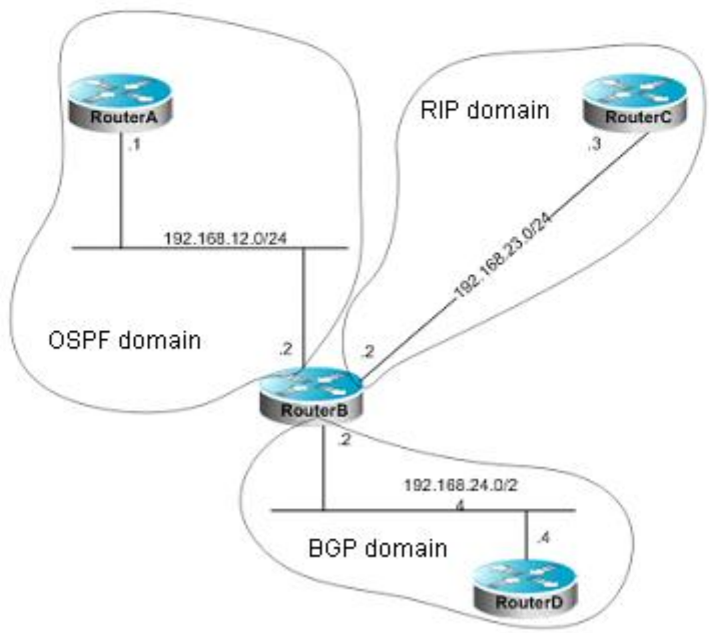
In this case, Router A has successfully learned the route to the network segment 172.10.10.0.

5.3.6 Example of OSPF Dynamic Route Redistribution Configuration

5.3.6.1 Configuration Requirements

The following figure shows the topology of four routers. Router A belongs to the OSPF routing area. Router C belongs to the RIP routing area. Router D belongs to the BGP routing area. Router B is connected to the three routing areas. Router A advertises two routes: 192.168.10.0/24 and 192.168.100.1/32. Router C advertises two routes: 192.168.3.0/24 and 192.168.30.0/24. Router D advertises two routes: 192.168.4.0/24 and 192.168.40.0/24.

Networking topology for dynamic routing protocol redistribution



On Router B, the OSPF redistributes routes (type-1) in the RIP routing area and the BGP routes that carry the community attribute 11:11 in the BGP routing area. The RIP redistributes the route 192.168.10.0/24 in the OSPF routing area and advertises a default route to the RIP routing area. The metric of this route is set to 2.

5.3.6.2 Specific Configurations

When the routing protocol redistribute the routes among each other, the simple route filtering can be controlled by using the distribution list. However, different attributes must be set for different routes, which cannot be implemented by using the distribution list. In this case, a route-map must be used for control. The route-map provides more control functions than the distribution list, but the router configuration is more complex. Therefore, do not use the route-map if possible. The following examples use the route-map to match the community attribute of the BGP routes.

Configurations on Router A:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
Qtech(config)# interface loopback 1
Qtech(config-if-Loopback 1)# ip address 192.168.100.1 255.255.255.255
Qtech(config-if-Loopback 1)# no ip directed-broadcast
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF.

```
Qtech(config)# router ospf 12
Qtech(config-router)# network 192.168.10.0 0.0.0.255 area 0
Qtech(config-router)# network 192.168.12.0 0.0.0.255 area 0
Qtech(config-router)# network 192.168.100.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.12.2 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.24.2
Qtech(config)# interface Serial 1/0
Qtech(config-Serial 1/0)# ip address 192.168.23.2 255.255.255.0
```

# Configure the OSPF and specify the type of routes to be redistributed.

```
Qtech(config)# router ospf 12
Qtech(config-router)# redistribute rip metric 100 metric-type 1 subnets
Qtech(config-router)# redistribute bgp route-map ospfrm subnets
Qtech(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

# Configure the RIP and use the distribute list to filter the redistributed routes.

```
Qtech(config)# router rip
Qtech(config-router)# redistribute ospf 12 metric 2
Qtech(config-router)# network 192.168.23.0
Qtech(config-router)# distribute-list 10 out ospf
Qtech(config-router)# default-information originate always
Qtech(config-router)# no auto-summary
```

# Configure the BGP.

```
Qtech(config)# router bgp 2
Qtech(config-router)# neighbor 192.168.24.4 remote-as 4
Qtech(config-router)# address-family ipv4
Qtech(config-router-af)# neighbor 192.168.24.4 activate
Qtech(config-router-af)# neighbor 192.168.24.4  send-community
```

# Configure the route-map.

```
Qtech(config)# route-map ospfrm
Qtech(config-route-map)# match community cl_110
```

# Define the access list.

```
Qtech(config)# access-list 10 permit 192.168.10.0
```

# Define the community list.

```
Qtech(config)# ip community-list standard cl_110 permit 11:11
```

Configurations on Router C:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.30.1 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.3.1 255.255.255.0
Qtech(config)# interface Serial 1/0
Qtech(config-if-Serial 1/0)# ip address 192.168.23.3 255.255.255.0
```

# Configure the RIP.

```
Qtech(config)# router rip
Qtech(config-router)# network 192.168.23.0
Qtech(config-router)# network 192.168.3.0
Qtech(config-router)# network 192.168.30.0
```

Configurations on Router D:

# Configure the network interface.

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.40.1 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.4.1 255.255.255.0
Qtech(config)# interface gigabitEthernet 0/3
Qtech(config-if-GigabitEthernet 0/3)# ip address 192.168.24.4 255.255.255.0
```

# Configure the BGP.

```
Qtech(config)# router bgp 4
Qtech(config-router)# neighbor 192.168.24.2 remote-as 2
Qtech(config-router)# redistribute connected route-map bgprm
Qtech(config-router)# address-family ipv4
Qtech(config-router-af)# neighbor 192.168.24.2 activate
Qtech(config-router-af)# neighbor 192.168.24.2 send-community
```

# Configure the route-map.

```
Qtech(config)# route-map bgprm
Qtech(config-route-map)# set community 22:22
```

The OSPF routes learned by Router A:

```
O E1 192.168.30.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
O E1 192.168.3.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
O E1 192.168.23.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
```

The RIP routes learned by Router C:

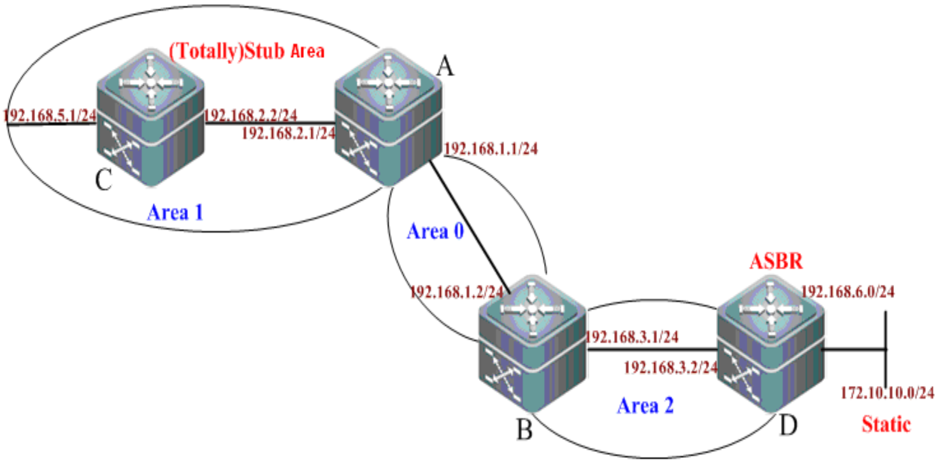
```
R    0.0.0.0/0 [120/1] via 192.168.23.2, 00:00:00, Serial 1/0
R    192.168.10.0/24 [120/2] via 192.168.23.2, 00:00:00, Serial 1/0
```

5.3.7 Example of OSPF (Totally) Stub Area Configuration

5.3.8 Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segment 172.10.10.0 is outside the routing area.

Networking topology for OSPF (Totally) Stub area configuration



5.3.8.1 *Applications*

Configure Router A and Router B as area border routers (ABR) and Router C as an intra-area router. Configure Router D as an ASBR and introduce one an external static route.

To reduce the size of the routing table inside the AS border and the number of routes exchanged, configure the specific area to be a (Totally) Stub area.

Routing information can be correctly transmitted in the OSPF autonomous system.

5.3.8.2 *Configuration Tips*

Do not configure the backbone area (Area 0) cannot be configured as a (Totally) Stub area, and there must be no ASBR exists in the (Totally) Stub area. That is, the external routes of the autonomous system cannot be propagated transmitted in this area. In this example, Area 1 is configured as the (Totally) Stub area.

When configuring an area as a Stub area, you must configure the **stub** command on all routers in this area. In this example, you need to configure this attribute on Router A and Router C.

When configuring an area as a Totally Stub area, you must configure the **stub** command on all routers (Router C) in this area and the **stub [no-summary]** command on the ABR (Router A).

5.3.8.3 Configuration Steps

The following information only shows how to configure an OSPF (Totally) Stub area. For other configurations, see "Example of Multi-Area OSPF Configuration" and "Example of OSPF Static Route Redistribution Configuration".

Step 1: Display the routing table of Router C when this router is in a normal area.

```
C#show ip route  ospf
O E2 172.10.10.0/24 [110/20] via 192.168.2.1, 4d,02:28:07, GigabitEthernet 0/3
                             //AS external route
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 4d,17:52:14, GigabitEthernet 0/3
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 4d,17:52:14, GigabitEthernet 0/3
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 4d,02:38:27, GigabitEthernet 0/3
```

In this case, the routing table contains AS external routes.

Step 2: Configure the Stub area

- Configurations on Router A:

```
A(config)#router ospf 1
A(config-router)#area 1 stub
```

- Configurations on Router C:

```
C(config)#router ospf 1
C(config-router)#area 1 stub
```

- Display the routing table of Router C when this router is in a stub area.

```
C#show ip route ospf
O*IA 0.0.0.0/0 [110/2] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
                                                //default route
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
```

In this case, the routing table contains no AS external route. The original AS external route in the routing table has been replaced with a default route.

Step 3: Configure the Totally Stub area

- Configurations on Router A:

```
A(config)#router ospf 1
A(config-router)#area 1 stub stub no-summary
```

- Configurations on Router C:

```
C(config)#router ospf 1
C(config-router)#area 1 stub
```

- Display the routing table of Router C when this router is in a totally stub area.

```
C#show ip route ospf
O*IA 0.0.0.0/0 [110/2] via 192.168.2.1, 00:30:53, GigabitEthernet 0/3
```

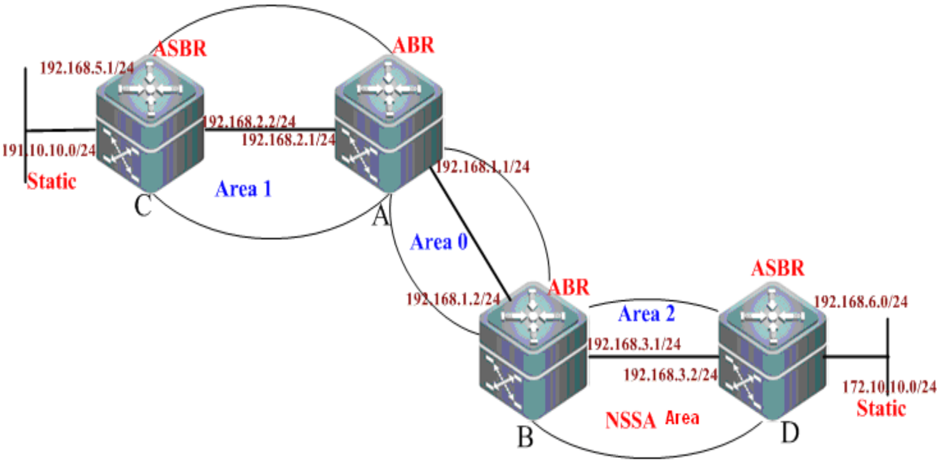
In this case, the routing table only contains one default route to the external area.

5.3.9 Example of OSPF NSSA Area Configuration

5.3.9.1 Networking topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segments 192.10.10.0 and 172.10.10.0 are outside the OSPF routing area.

Networking topology for OSPF NSSA area configuration



5.3.9.2 Applications

1. Configure Router A and Router B as area border routers (ABR). Configure Router C and Router D as ASBRs and introduce an AS external static route for Router C and Router D respectively.

2. Area 2 shall be configured as an NSSA area in order to reduce the size of routing table of the intra-area router and the number of routes exchanged. Meanwhile, prohibit Router B from sending summary LSAs (Type-3 LSA) to the NSSA area.

3. Routing information can be correctly transmitted in the OSPF autonomous system.

5.3.9.3 Configuration Tips

Tips for configuring the NSSA area are as follows:

1. The backbone area (Area 0) cannot be configured as the NSSA area.

2. The ASBRs can exist in the NSSA area, and certain number of AS external routes can be imported to the OSPF routing area.

3. When configuring an area as the NSSA area, you must use the area nssa command on all routers (Router B and D) connected to the NSSA area.

5.3.9.4 Configuration Steps

The following information only shows how to configure the NSSA area. For the basic OSPF configurations, see the above examples.

Step 1: Configure static route redistribution.

➢ Configurations on Router C:

! Configure a static route.

```
C(config)#ip route 191.10.10.0 255.255.255.0 192.168.5.2
```

! Redistribute the static route based on the OSPF.

```
C(config)#router ospf 1
C(config-router)#redistribute static subnets
```

- Configurations on Router D:

! Configure a static route.

```
D(config)#ip route 172.10.10.0 255.255.255.0 192.168.6.2
```

! Redistribute the static route based on the OSPF.

```
C(config)#router ospf 1
C(config-router)#redistribute static subnets
```

Step 2: Configure the NSSA.

- Configurations on Router B (ABR):

```
B(config)#router ospf 1
```

! Define the NSSA area and prohibit this ABR from sending summary LSAs (Type-3 LSA) to the NSSA area.

```
B(config-router)#area 2 nssa no-summary
```

- Configurations on Router D (ASBR):

```
D(config)#router ospf 1
D(config-router)#area 2 nssa
```

5.3.9.5 *Verification*

Step 1: Display the routing information when Area 2 is configured as a normal area.

- Display the routing table of Router D (ASBR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O E2 191.10.10.0/24 [110/20] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

- Display the OSPF routing table of Router B (ABR).

```
B#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.3.2, 17:53:35, GigabitEthernet 0/2    O E2
191.10.10.0/24 [110/20] via 192.168.1.1, 00:57:46, GigabitEthernet 0/1
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 5d,15:39:01, GigabitEthernet 0/1
O IA 192.168.5.0/24 [110/3] via 192.168.1.1, 01:10:34, GigabitEthernet 0/1
O    192.168.6.0/24 [110/2] via 192.168.3.2, 17:53:36, GigabitEthernet 0/2
```

Step 2: Display the routing information about each router in the NSSA area when Area 2 is configured as an NSSA area.

- Display the OSPF routing table of Router B (ABR).

```
B#show ip route ospf
```

```
O N2 172.10.10.0/24 [110/20] via 192.168.3.2, 00:01:00, GigabitEthernet 0/2
O E2 191.10.10.0/24 [110/20] via 192.168.1.1, 01:11:26, GigabitEthernet 0/1
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 5d,15:52:41, GigabitEthernet 0/1
O IA 192.168.5.0/24 [110/3] via 192.168.1.1, 01:24:14, GigabitEthernet 0/1
O    192.168.6.0/24 [110/2] via 192.168.3.2, 00:01:01, GigabitEthernet 0/2
```

In this case, the ABR in the NSSA area has translated the AS external routes imported into this area into N2 (OSPF NSSA external type 2) routes and transmitted to other areas.

➢ Display the routing table of Router D (ASBR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

In this case, the AS external routes imported into other areas cannot reach this area when Router D is in an NSSA area.

Step 3: Display the routing information of the NSSA area when configuring the attribute of the NSSA area to be no-summary on Router B (ABR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is 192.168.3.1 to network 0.0.0.0
O*IA 0.0.0.0/0 [110/2] via 192.168.3.1, 00:00:40, GigabitEthernet 0/3
S    172.10.10.0/24 [1/0] via 192.168.6.2
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

In this case, the routing table contains a default route which replaces the inter-area route.

Step 4: Display the OSPF routing information on routers in other areas. Key point: Note whether there is any AS external route imported into the NSSA area.

```
SwitchA#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.1.2, 02:08:08, GigabitEthernet 0/1
O E2 191.10.10.0/24 [110/20] via 192.168.2.2, 03:18:35, GigabitEthernet 0/2
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 5d,17:59:01, GigabitEthernet 0/1
O    192.168.5.0/24 [110/2] via 192.168.2.2, 03:31:25, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 02:08:09, GigabitEthernet 0/1
```

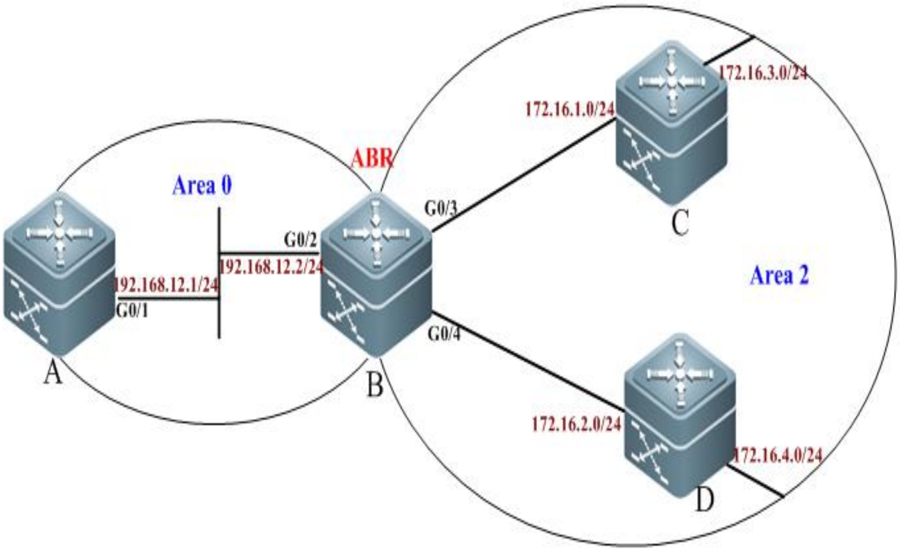
In this case, the routing table of Router A contains an AS external route imported into the NSSA area.

5.3.10 Example of OSPF Inter-area Route Aggregation Configuration

5.3.10.1 Networking Topology

The following figure shows the topological topology of an OSPF autonomous system, in which the network segment 192.168.12.0/24 belongs to Area 0 and the network segments 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 belong to Area 2.

Networking topology for OSPF inter-area route aggregation configuration



5.3.10.2 Applications

To reduce the size of routing table, configure Router B so that Router B only advertises the summary route of four network segments (172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24) instead of separately advertising the routes of these four network segments.

5.3.10.3 Configuration Tips

1. Since the network segments 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 are consecutive addresses, you can configure route aggregation on the area border router (Router B) to alleviate route calculation. Use this command **area range** to configure the route aggregation between the OSPF inter-areas".

2. During route aggregation, the aggregated route range may exceed the actual network range in the routing table. Routing loop may incur or load on the router may increase if packets are sent to a network that is beyond the aggregated route range. So you need to add a "discard" route into the routing table on the ABR (Router B) or ASBR. Use the inter-area route aggregation command area range to add the discard route. This function is enabled by default.

3. The aggregated route address of 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 is 172.16.0.0/21. Routes falling within this range will not be advertised to other areas by the ABR.

5.3.10.4 Configuration Steps

Step 1: Configure the IP address for interfaces.(omitted)

Step 2: Configure the basic OSPF functions.

- Configure Router A.

```
A(config)#router ospf 1
A(config)# network 192.168.12.0 0.0.0.255 area 0
```

- Configure Router B.

```
B(config)#router ospf 1
B(config-router)#network 192.168.12.0 0.0.0.255 area 0
B(config-router)#network 172.16.1.0 0.0.0.255 area 2
B(config-router)#network 172.16.2.0 0.0.0.255 area 2
```

- Configure Router C.

```
C(config)#router ospf 1
C(config-router)#network 172.16.1.0 0.0.0.255 area 2
C(config-router)#network 172.16.3.0 0.0.0.255 area 2
```

- Configure Router D.

```
D(config)#router ospf 1
D(config-router)#network 172.16.2.0 0.0.0.255 area 2
D(config-router)#network 172.16.4.0 0.0.0.255 area 2
```

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 172.16.1.0/24 [110/2] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.2.0/24 [110/2] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.3.0/24 [110/3] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.4.0/24 [110/3] via 192.168.12.2, 00:06:19, GigabitEthernet 0/1
```

In this case, the detailed routing information about Area 2 is advertised to Area 0.

Step 3: Configure the inter-area route aggregation on the ABR (Router B).

```
B(config)#router ospf 1
B(config-router)#area 2 range 172.16.0.0 255.255.248.0
```

Step 4: On ABR (Router B), configure to control the addition of the aggregated route entry into the core routing table. This function is enabled by default.

```
B(config-router)# discard-route internal
```

5.3.10.5 Verification

- After configuring route aggregation, display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 172.16.0.0/21 [110/2] via 192.168.12.2, 00:01:04, GigabitEthernet 0/1
```

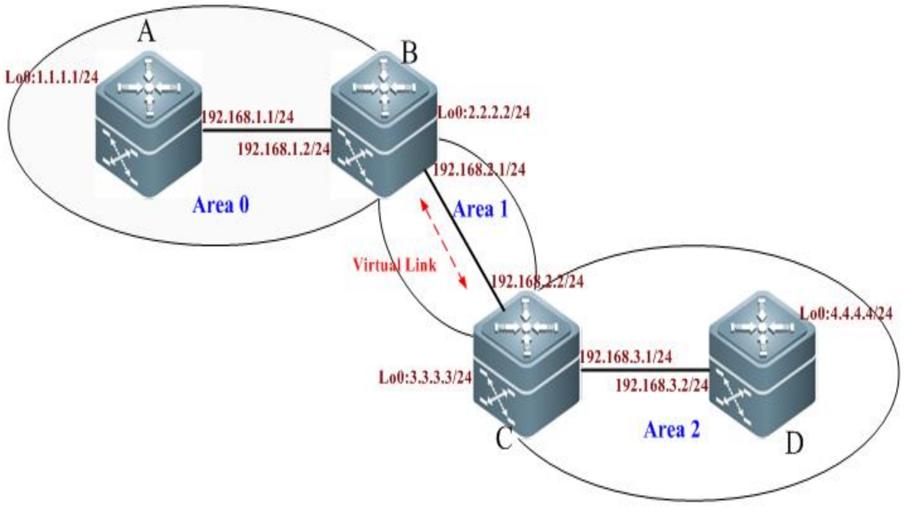
In this case, only the aggregated routes are advertised. Specific routes will not be advertised by the ABR to other areas. The size of the routing table is decreased substantially.

5.3.11 Example of OSPF Virtual Link Configuration

5.3.11.1 Networking Topology

The following figure shows an OSPF routing area. The network segment 192.168.1.0 belongs to Area 0. The network segment 192.168.2.0 belongs to Area 1. The network segment 192.168.3.0 belongs to Area 2. Due to the limitation of physical conditions, other specific areas cannot be deployed around the backbone area. As shown in the following figure, Area 2 is not directly connected to Area 0.

Networking topology for OSPF virtual link configuration



5.3.11.2 Applications

Through configuration, Router D shall be able to receive routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1). Meanwhile, Router B shall be able to learn the routes of the network segment 192.168.3.0/24 (Area 2).

Details about IP address assignment are shown as follows:

Router name	Router ID	Interface address
A	1.1.1.1	Gi0/1: 192.168.1.1/24
B	2.2.2.2	Gi0/1: 192.168.1.2/24 Gi0/3: 192.168.2.1/24
C	3.3.3.3	Gi0/3: 192.168.2.2/24 Gi0/5: 192.168.3.1/24
D	4.4.4.4	Gi0/5: 192.168.3.2/24

5.3.11.3 Configuration Tips

When the OSPF routing area is composed of multiple areas, each area must be directly connected to the backbone area (Area 0). Otherwise, these areas cannot be interconnected. If there is no direct physical link, create virtual links to logically connect each area to the backbone area. Configuration tips are shown as follows:

- Configure the IP address for the interfaces. (Omitted)
- Configure the basic OSPF functions.
- Configure OSPF virtual links

The virtual link must be configured on ABRs. This example configures virtual links on Router B and Router C.

Use the **area** *area-id* **virtual-link** *router-id* command to configure virtual links on the ABRs. The router-id refers to the identifier of a peer device.

5.3.11.4 Configuration Steps

Step 1: Configure the basic OSPF functions.

- Configurations on Router A:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
A(config)#router ospf 1
A(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

```
A(config-router)#exit
```

! Configure the loopback IP 1.1.1.1 as the router ID of Router A.

```
A(config)#interface loopback 0
A(config-Loopback 0)#ip address 1.1.1.1 255.255.255.0
```

- Configurations on Router B:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
B(config)#router ospf 1
B(config-router)#network 192.168.1.0 0.0.0.255 area 0
B(config-router)#network 192.168.2.0 0.0.0.255 area 1
B(config-router)#exit
```

! Configure the loopback IP 2.2.2.2 as the router ID of Router B.

```
B(config)#interface loopback 0
B(config-Loopback 0)#ip address 2.2.2.2 255.255.255.0
```

- Configurations on Router C:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
C(config)#router ospf 1
C(config-router)#network 192.168.2.0 0.0.0.255 area 1
C(config-router)#network 192.168.3.0 0.0.0.255 area 2
C(config-router)#exit
```

! Configure the loopback IP 3.3.3.3 as the router ID of Router C.

```
C(config)#interface loopback 0
C(config-Loopback 0)#ip address 3.3.3.3 255.255.255.0
```

- Configurations on Router D:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
D(config)#router ospf 1
D(config-router)#network 192.168.3.0 0.0.0.255 area 2
D(config-router)#exit
```

! Configure the loopback IP 4.4.4.4 as the router ID of Router D.

```
D(config)#interface loopback 0
D(config-Loopback 0)#ip address 4.4.4.4 255.255.255.0
```

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:32:48, GigabitEthernet 0/1
```

Since Area 2 is not directly connected to Area 0, the routing table of Router A contains no routing information about Area 2

Step 2: Configure OSPF virtual links.

- Configure Router B.

```
B(config)#router ospf 1
B(config-router)#area 1 virtual-link 3.3.3.3
```

- Configure Router C.

```
C(config)#router ospf 1
C(config-router)#area 1 virtual-link 2.2.2.2
```

5.3.11.5 Verification

- Display the OSPF routing table of Router B.

```
B#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C     2.2.2.0/24 is directly connected, Loopback 0
C     2.2.2.2/32 is local host.
C     192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C     192.168.1.2/32 is local host.
C     192.168.2.0/24 is directly connected, GigabitEthernet 0/3
C     192.168.2.1/32 is local host.
O IA 192.168.3.0/24 [110/2] via 192.168.2.2, 00:02:49, GigabitEthernet 0/3
```

In this case, after the virtual link is configured, Router B has successfully learned the routes of the network segment 192.168.3.0/24 (Area 2).

- Display the OSPF routing table of Router D.

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C     4.4.4.0/24 is directly connected, Loopback 0
C     4.4.4.4/32 is local host.
O IA 192.168.1.0/24 [110/3] via 192.168.3.1, 00:04:45, GigabitEthernet 0/5
O IA 192.168.2.0/24 [110/2] via 192.168.3.1, 00:05:02, GigabitEthernet 0/5
C     192.168.3.0/24 is directly connected, GigabitEthernet 0/5
C     192.168.3.2/32 is local host.
```

In this case, after the virtual link is configured, Router D has successfully learned the routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1).

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:51:22, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/3] via 192.168.1.2, 00:07:58, GigabitEthernet 0/1
```

In this case, after the virtual link is configured, Router A has successfully learned the routes of the network segment 192.168.3.0/24 (Area 2).

- Display the OSPF virtual link information about Router B.

```
B#show ip ospf 1 virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
Adjacency state Full
```

- Display the OSPF virtual link information about Router C.

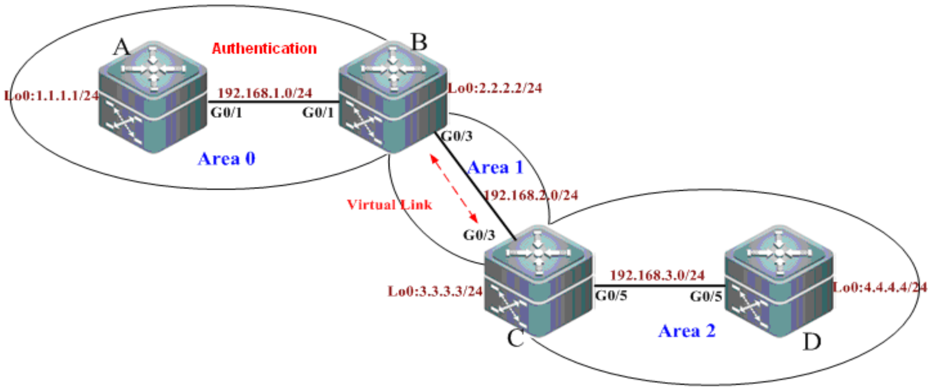
```
C#show ip ospf 1 virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.2/32
  Remote address 192.168.2.1/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
    Adjacency state Full
```

5.3.12 Example of OSPF Authentication Configuration

5.3.12.1 Networking Topology

The following figure shows an OSPF routing area. The network segment 192.168.1.0 belongs to Area 0. The network segment 192.168.2.0 belongs to Area 1. The network segment 192.168.3.0 belongs to Area 2. Due to the limitation of network structures, Area 2 is connected to Area 0 through virtual links.

Networking topology for OSPF authentication configuration



5.3.12.2 Applications

1. To prevent the device from learning unauthenticated and invalid routes and advertising valid routes to unauthenticated devices, it is required to configure area authentication in the backbone area (Area 0), with the authentication type being MD5.

2. Router D shall be able to learn routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1). Meanwhile, Router B shall be able to learn the routes of the network segment 192.168.3.0/24 (Area 2).

5.3.12.3 Configuration Tips

To configure the OSPF area authentication, configure the area authentication on all routers in the same area with the same authentication type. This example enables the area authentication in Area 0, namely all routers (Router A and Router B) in Area 0 shall be configured with the same authentication type.

When OSPF virtual links are used to connect a non-backbone area (Area 2) with a backbone area, if ID authentication is enabled in the backbone area (Area 0), the identity authentication shall also be configured on the ABR (Router C) in the non-backbone area.

Tips for configuring the OSPF area authentication are shown as follows:

8) In OSPF route configuration mode, specify the authentication type for the area.

9) Configure the authentication type and key on the interface.

5.3.12.4 Configuration Steps

The following information only shows how to configure the OSPF area authentication. For other configurations, see "Example of OSPF Virtual Link Configuration".

- **Configure Router A.**

Step 1: In OSPF route configuration mode, specify Area 0 to enable the MD5 authentication.

```
A(config)#router ospf 1
A(config-router)#area 0 authentication message-digest
A(config-router)#exit
```

Step 2: Configure the authentication type and key on the interface.

```
A(config)#interface gigabitEthernet 0/1
A(config-if-GigabitEthernet 0/1)#ip ospf message-digest-key 1 md5 hello
```

- **Configure Router B.**

Step 1: In OSPF route configuration mode, specify Area 0 to enable the MD5 authentication.

```
B(config)#router ospf 1
B(config-router)#area 0 authentication message-digest
B(config-router)#exit
```

Step 2: Configure the authentication type and key on the interface.

```
B(config)#interface gigabitEthernet 0/3
B(config-if-GigabitEthernet 0/3)#ip ospf message-digest-key 1 md5 hello
```

- **Configure Router C.**

! Enable the identity authentication of the backbone area (Area 0) on Router C.

```
C(config)#router ospf 1
C(config-router)#area 0 authentication message-digest
```

5.3.12.5 Verification

Step 1: Display the OSPF information about the routers when the authentication is enabled only on Router A and Router B. (disabled on Router C)

! Display the virtual link configurations of Router B.

```
B#show ip ospf virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
Adjacency state Down
```

In this case, the adjacency state is down.

! Display the virtual link configurations of Router C.

```
C#show ip ospf virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.2/32
  Remote address 192.168.2.1/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
Adjacency state Down
```

In this case, the adjacency state is down.

! Display the OSPF routing information about Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:10:59, GigabitEthernet 0/1
```

In this case, Router A has failed to learn the routes of Area 2.

Step 1: Display the OSPF information about the routers after the authentication is enabled on Router A and Router B and the identity authentication of Area 0 is enabled on Router C.

! Display the virtual link configurations of Router B.

```
B#show ip ospf virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
Adjacency state Full
```

In this case, the adjacency state is full.

! Display the virtual link configurations of Router C.

```
C#show ip ospf virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
Transit area 0.0.0.1 via interface GigabitEthernet 0/3
Local address 192.168.2.2/32
Remote address 192.168.2.1/32
Transmit Delay is 1 sec, State Point-To-Point,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:00
Adjacency state Full
```

In this case, the adjacency state is full.

! Display the OSPF routing information about Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:21:30, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/3] via 192.168.1.2, 00:03:18, GigabitEthernet 0/1
```

In this case, Router A has successfully learned the routes of Area 2.

Step 3: Display general OSPF information about Router A.

```
A#show ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1
 Process uptime is 18 hours 22 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Enable two-way-maintain
 Initial SPF schedule delay 1000 msecs
 Minimum hold time between two consecutive SPFs 5000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Initial LSA throttle delay 0 msecs
 Minimum hold time for LSA throttle 5000 msecs
 Maximum wait time for LSA throttle 5000 msecs
 Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
 Minimum LSA arrival 1000 msecs
 Pacing lsa-group: 240 secs
 Number of incoming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum 0x000000
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 0
 External LSA database is unlimited.
 Number of LSA originated 2
 Number of LSA received 244
 Log Neighbor Adjacency Changes: Enabled
  Number of areas attached to this router: 1: 1 normal 0 stub 0 nssa
    Area 0 (BACKBONE)
        Number of interfaces in this area is 1(1)
        Number of fully adjacent neighbors in this area is 1
        Area has message digest authentication
        SPF algorithm last executed 17:24:38.030 ago
        SPF algorithm executed 11 times
        Number of LSA 7. Checksum 0x032955
```

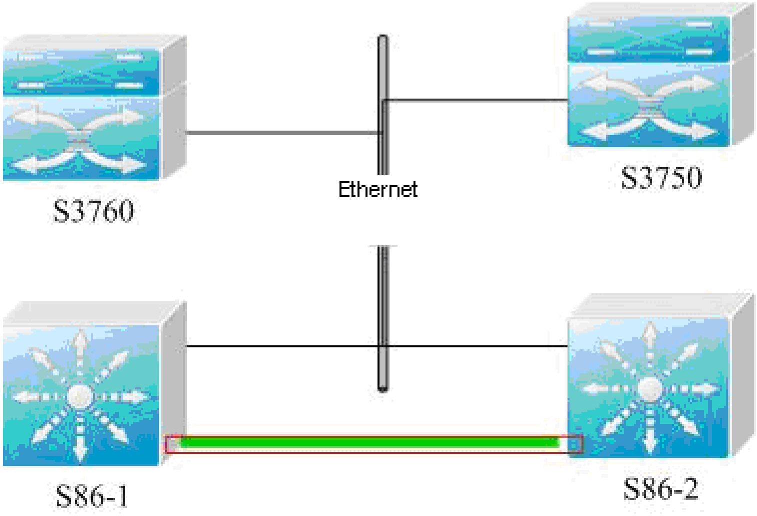
The above information shows that the area authentication has been enabled.

5.3.13 Example of OSPF GR Configuration

5.3.13.1 Configuration Requirements

The following figure shows that two S86 high-end switches have the GR Restart capability and are equipped with primary and secondary engines to support redundant backup at the control plane. S86-1 establishes the OSPF adjacency relationship with S86-2, S3760, and S3750. The OSPF GR capability is supported by all devices. The connection layout is shown as follows:

OSPF GR configuration



It is required that two S86 devices shall support non-stop packet forwarding to enhance the reliability of core devices.

5.3.13.2 Specific Configurations

Configure S3760.

```
Qtech(config)# router ospf 1
Qtech(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S3750.

```
Qtech(config)# router ospf 1
Qtech(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S86-1.

```
Qtech(config)# router ospf 1
Qtech(config-router)# graceful-restart
Qtech(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S86-2.

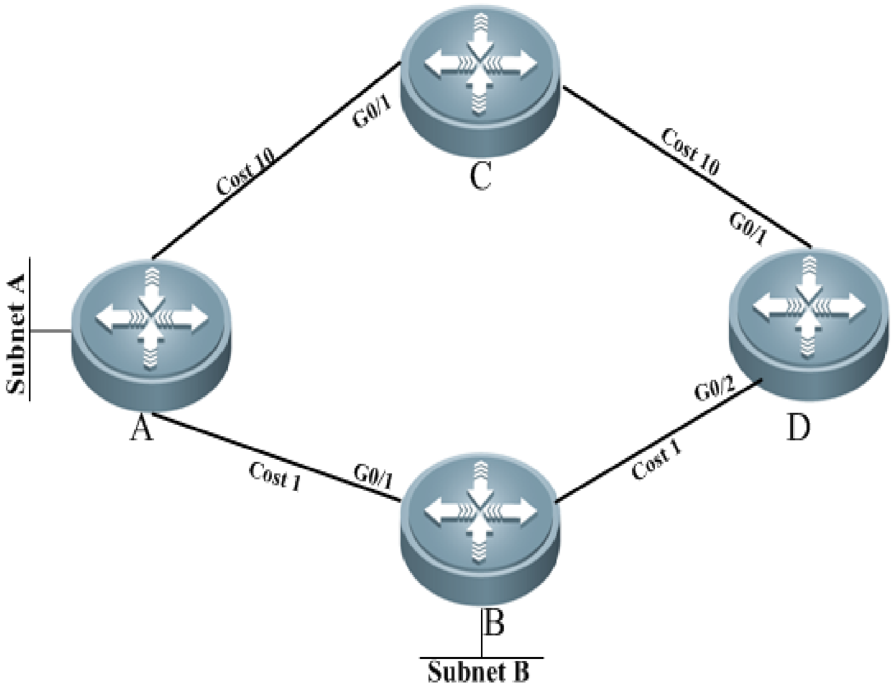
```
Qtech(config)# router ospf 1
Qtech(config-router)# graceful-restart
Qtech(config-router)# graceful-restart helper strict-lsa-checking
```

5.3.14 Example of OSPF Stub Router Configuration

5.3.14.1 Configuration Requirements

Four routers form an OSPF routing area. The connection layout is shown in the following figure. According to the rule for optimal routing, the route from D to subnet A passes B. It is expected that the route passes C by changing configurations of B only.

OSPF Stub Router configuration



It is required that B only transmit routes to Subnets B and C transmit other routes.

5.3.14.2 Specific Configurations

Configure IP addresses and OSPF processes on the four routers, and make the following configurations after the adjacency relationship have been established successfully.

Configurations on D:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 10
interface gigabitEthernet 0/2
ip ospf cost 1
```

Configurations on C:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 10
```

Configurations on B:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 1
```

Configure the OSPF routing protocol.

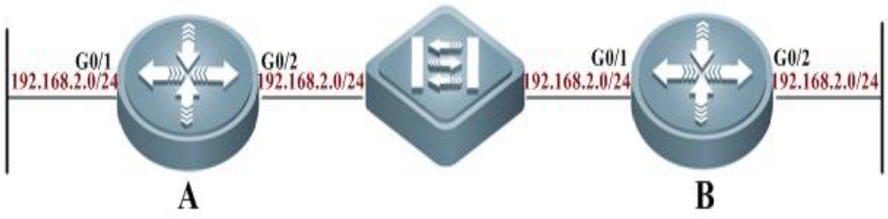
```
router ospf 1
max-metric router-lsa
```

5.3.15 Example of OSPF Fast Convergence Configuration

5.3.15.1 Configuration Requirements

Routers A and B are interconnected through a layer-2 switch. Run the OSPF protocol on them to establish routes. The following figure shows details about IP address assignment and connection layout.

OSPF fast convergence configuration



After link failure between B and the layer-2 switch occurs, A shall be able to detect adjacency changes within 1 second and quickly respond to the network changes.

5.3.15.2 Specific Configurations

The Fast Hello function reduces the time for detecting adjacency changes to less than 1 second. Meanwhile, the LSA fast convergence function facilitates to adaptation to the swift network changes.

Configurations on A:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip address 192.168.1.1 255.255.255.0
interface gigabitEthernet 0/2
ip address 192.168.2.1 255.255.255.0
ip ospf dead-interval minimal hello-multiplier 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
timers arrival-time 100
timers throttle lsa all 0 100 500
```

Configurations on B:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip address 192.168.3.1 255.255.255.0
interface gigabitEthernet 0/2
ip address 192.168.2.2 255.255.255.0
ip ospf dead-interval minimal hello-multiplier 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.2.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0
timers arrival-time 100
timers throttle lsa all 0 100 500
```

6 CONFIGURING OSPFV3

OSPFV2 (RFC2328, OSPFv2) runs under the IPv4. The RFC5340 describes OSPFV3, the extension of OSPFv2 that provides support for IPv6 routes. This document briefly describes the OSPFv3 protocol and its configuration.

Before learning this document, you must know the OSPFv2 protocol and related configuration.

The OSPFv3 protocol extends the OSPFv2 protocol with the main operating mechanisms and most configuration the same as the OSPFv2.

6.1 Overview

As an Interior Gateway Protocol (IGP), the OSPF runs among the layer 3 devices within an Autonomous System (AS).

Unlike a vector distance protocol, the OSPF is a link-state protocol. By exchanging various types of link-state advertisements (LSAs) recording link state between devices, it synchronizes link state information between devices and then calculates OSPF route entries through the Dijkstra algorithm.

The OSPFv3 is described in the RFC5340 and supports the IPv6. This section describes the differences from the OSPFv2 in implementation.

LSA Association Change

Interface Configuration

Router ID Configuration

Authentication Mechanism Configuration

LSA Association Change

Just as described above, the OSPF is a link-state protocol and its implementation is based on LSAs. Through LSAs, we can know the topologies of networks and address information. In contrast to the IPv4, the IPv6 uses 128-bit IP addresses. The design of LSAs is modified accordingly. Firstly, the LSA types are described as follows:

Router-LSAs (Type 1)

Each device generates this type of LSAs by itself. They describe the states of its links in specified areas and the cost spent in reaching the links. In contrast to the OSPFv2, the Router-LSAs of the OSPFv3 only indicate the state information of links. They do not record the information about the network addresses connected to routers. The information will be acquired by newly added types of LSAs. Additionally, in the OSPFv2, only one Router-LSA is allowed to be generated for each device in each area. While in the OSPFv3, multiple Router-LSAs are allowed to be generated. Thus, when performing the SPF calculation, we must consider all the Router-LSAs generated by the device. Router-LSAs and Network-LSAs describe the link topology of areas together.

Through the flag bits on Router-LSAs, we can know whether the routers are Area Border Routers (ABR), AS boundary routers (ASBR) or those at one end of a virtual link.

Network-LSAs (Type 2)

Network-LSAs only exist in broadcast networks or NBMA networks and are generated by DRs (Designated Routers) in a network. They describe the information about all the routers connected in specified areas on a network. Like Router-LSAs, Network-LSAs also only indicate the link-state information and do not record the network address information. Network-LSAs and Router-LSAs describe the link topology of areas together.

Inter-Area-Prefix-LSAs (Type 3)

They are generated for an area by the ABRs in the area and used to describe the network information about reaching other areas. They replace type 3 summary-LSAs in OSPFv2. In contrast to the OSPFv2, they use a prefix structure to describe the destination network information.

Inter-Area-Router-LSAs (Type 4)

They are generated for an area by the ABRs in the area, used to describe the path information about reaching the ASBRs in other areas, and replace type 4 summary-LSAs in OSPFv2.

AS-external-LSAs (Type 5)

This type of LSAs is generated by ASBRs and used to describe the network information about reaching outside the AS. Usually, the network information is generated through other routing protocols. In contrast to the OSPFv2, it uses a prefix structure to describe the destination network information.

NSSA-LSA (Type 7)

Their function is same as that of type 5 AS-external-LSAs. However, they are generated by ASBRs in the NSSA area.

Link-LSAs (Type 8)

In the OSPFv3, the newly added LSA type is generated by each device for each connected link and describes the local link address of the device in the current link and all set IPv6 address prefix information.

Intra-Area-Prefix-LSAs (Type 9)

It is a newly added LSA type in the OSPFv3 and provides additional address information for Router-LSAs or Network-LSAs. Therefore, it plays two roles:

Associating network-LSAs and recording the prefix information of a transit network.

Associating router-LSAs and recording the prefix information on all Loopback interfaces, point-to-point links, point-to-multipoint links, virtual links and stub networks of the router in the current area.

Other major changes associated with LSA:

LSA flooding scope

In the OSPFv2, the LSA flooding occurs inside areas and ASs. In the OSPFv3, flooding occurs also in local links. Type 8 Link-LSAs is the type that can flood only inside a local link.

Handling an unknown LSA type

This is an improvement of OSPFv3 based on OSPFv2.

In the OSPFv2, database synchronization is necessary in the initial establishment of the adjacency relationship. If there is an unrecognizable LSA type in the database description message, this relationship cannot be established properly. If there is an unrecognizable LSA type in a link-state updating message, then the type of LSAs will be dropped.

In the OSPFv3, it is allowed to receive an unknown LSA type. By using the information recorded in the LSA header, we can determine how to handle the unrecognizable LSA type received.

6.1.1 Interface Configuration

In the OSPFv3, the changes based on interface configuration are as follows:

In order for an interface to run OSPFv3, enable the OSPFv3 directly in the interface configuration mode. For OSPFv2, however, run the **network** command in the OSPF route configuration mode.

If an interface runs OSPFv3, all the addresses on the interface will run IPv6. In the OSPFv2,however, all the addresses are enabled via the **network** command.

In the environment where the OSPFv3 runs, a link can support multiple OSPF entities and different devices connecting this link can run one of these OSPF instances. The OSPFv3 adjacency can only be established between the devices with the same instance ID. The OSPFv2 does not support this function.

6.1.2 Router ID Configuration

RFC5340 specifies the OSPFv3 Router ID is in the format of 32-bit IPv4 address but not the IPv6 address.

By default, the methods of electing OSPFv3 Router ID are the same as the OSPFv2 process . The automatic election method is adopted. Firstly, the largest IPv4 address for the loopback interface is elected as the Router ID. if the loopback interface of IPv4 addresses has not been configured, OSPFv3 process will select the largest IPv4 address for other interfaces as the Router ID. With multiple OSPFv3 processes running on the device, the OSPFv3 process selects the Router ID with the highest priority from the unselected IPv4 addresses in the above way. Different Router IDs are for the different processes.

If the IPv4 addresses available for the Router ID selection are insufficient, the OSPFv3 process will fail to auto-obtain the Router ID. You can use the **router-id** command to configure a Router ID to enable the OSPFv3 process.

The Router ID for each router in the AS must be unique. With multiple OSPFv3 processes running on the same device, the Router ID for each process must also be unique.

6.1.3 Authentication Mechanism Configuration

OSPFv2 itself supports two authentication modes: plain text authentication and key authentication based on MD5. Authentication fields have been removed from OSPFv3 packet headers. OSPFv3 does away its support for authentication entirely, instead relying on IPsec framework offered by IPv6. When OSPFv3 runs on IPv6, OSPFv3 requires the IPv6 authentication header (AH) or IPv6 encapsulating security payload (ESP) to ensure integrity and confidentiality of routing exchanges.

Configure authentication commands to enable IP AH, which only provides authentication on data integrity and consistency. Configure encryption commands to enable IP ESP, which covers AH functions and ensures confidentiality. Namely, authentication and encryption are performed simultaneously.

OSPFv3 authentication configuration can be based on an interface, an area or a virtual link. If you want to achieve higher security, configure different IPsec authentications on every interface. Area configuration is effective for all interfaces except the virtual link within this area. If area configuration and interface configuration are both performed, IPsec of interface configuration has a higher priority.

6.1.4 Basic Configuration of OSPFv3

The OSPFv3 protocol of Qtech Network has the following features:

Supports multi-instance OSPF;

Supports network type setting;

Supports virtual links;

Supports passive interfaces;

Supports an interface to select a participant OSPF instance;

Supports stub area;

Supports route redistribution;

Supports route aggregation;

Supports timer setting;

Supports link detection using BFD mechanism

To be implemented:

Supports NSSA areas;

Supports authentication. The OSPFv3 will use the IPSec authentication mechanism.

Default OSPFv3 Configuration:

Router ID		Undefined
Interface Configuration	Interface type	Broadcast network
	Interface cost	Undefined
	Hello message sending interval	10 seconds
	Dead interval of adjacent device	4 times of the hello interval.
	LSA sending delay	1 seconds
	LSA retransmitting interval	5 seconds
	Priority	1

	MTU check of database description messages	Enabled
Virtual Link	Virtual Link	Undefined
	Hello message sending interval	10 seconds
	Dead interval of adjacent device	4 times the hello interval.
	LSA sending delay	1 seconds
	LSA retransmitting interval.	5 seconds
	Fast Hello function	Disabled
Area Configuration	Area	Undefined
	Default router cost for stub areas	1
Routing information Aggregation	Inter-area route aggregation	Off
	External route aggregation	Off
Management Distance	Intra-area route	110
	Inter-area route	110
	External route	110
Auto cost generation		Enabled The default cost reference is 100 Mbps.
Shortest path first (SPF) timer		Time from receiving the topology change to running next SPF calculation :5 seconds The least interval between two calculations: 10 seconds
Route redistribution		Off
Route filtering		Off
Passive interface		Off

6.1.5 Enabling OSPFv3

Perform the following steps in the privileged mode to enable the OSPFv3:

Command	Function
configure terminal	Enter the global configuration mode
ipv6 router ospf *process-id*	Start an OSPFv3 routing process and enter the OSPFv3 configuration mode.
router-id *router-id*	Configure the Router ID used for OSPFv3 running on this device.
interface interface-id	Enter the interface configuration mode

ipv6 ospf process-id **area** *area-id* **[instance** *instance-id*]	Enable the OSPFv3 on the interface. instance-id：Set the interface instance ID that participates the OSPFv3. The interfaces of different devices connected to the same network can choose different OSPFv3 instances to participate.
copy running-config startup-config	Save the configuration.

The OSPFv3 instance ID and process ID are different. OSPFv3 process ID is valid for the device itself only, not influencing the interaction with other routers. While the OSPFv3 instance ID influences the interaction with other routers. Only the devices with the same instance ID can set up the OSPFv3 neighbor relationship.

First enable the interface to participate in the OSPFv3 and then configure the OSPFv3 process in the interface configuration mode. Once the process is configured, the interface will automatically participate in the corresponding process. Currently, our products can support up to 32 OSPFv3 processes.

6.1.6 Configuring OSPFv3 Parameters on the Interface

You can modify the interface parameters in the interface configuration mode according to the actual application.

To configure the OSPFv3 interface parameters, execute the following commands in the interface configuration mode:

Command	**Function**
ipv6 ospf *process-id* **area** *area-id* [**instance** *instance-id*]	Set the interface to participate in the OSPFv3 routing process.
ipv6 ospf network {**broadcast** | **non-broadcast** | **point-to-point** | **point-to-multipoint** [**non-broadcast**]} [**instance** *instance-id*]	Set the network type of an interface. The default is the broadcast network type.
ipv6 ospf neighbor *ipv6-address* {[**cost** *<1-65535>*] | [**poll-interval** *<0-2147483647>* | **priority** *<0-255>*]} [**instance** *instance-id*]	(Optional) Set the OSPFv3 neighbor.
ipv6 ospf cost *cost* [**instance** *instance-id*]	(Optional) Define the interface cost.
ipv6 ospf hello-interval *seconds* [**instance** *instance-id*]	(Optional) Set the interval for sending the Hello packets on the interface. For all nodes in adjacency on the network, this value must be identical.
ipv6 ospf dead-interval *seconds* [**instance** *instance-id*]	(Optional) Set the adjacency dead-interval on the interface. For all nodes in adjacency on the network, this value must be identical.
ipv6 ospf transmit-delay *seconds* [**instance** *instance-id*]	(Optional) Set the delay in sending LSA on the interface.
ipv6 ospf retransmit-interval *seconds* [**instance** *instance-id*]	(Optional) Set the interval for retransmitting LSA on the interface.
ipv6 ospf priority *number* [**instance** *instance-id*]	(Optional) Set the priority of the interface for elect the DR and BDR.

Command	Function
ipv6 ospf authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	(Optional) Sets the same interface authentication parameters on both sides.. *spi*: security parameter index within the ragne from 256 to 4294967295. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key.
ipv6 ospf encryption ipsec spi *spi* **esp** *null* [**md5** | **sha1**] [**0** | **7**] *key*	(Optional) Sets the same interface authentication parameters on both sides.. *spi*: security parameter index within the ragne from 256 to 4294967295. **null**: specifies null encryption mode. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key.

Use the **no** form of the above command to invalidate the configuration.

You can modify the parameter settings according to the actual needs. But note that some parameter settings must be consistent with those of neighbors otherwise no adjacency can be established. These parameters include the following: **instance**, **hello-interval**, **dead-interval**, **authentication** and **encryption**

6.1.7 Configuring OSPFv3 Area Parameter

The OSPF protocol applies the concept of “hierarchical structure”, allowing a network to be divided into a group of parts connected through a “backbone” in a mutual independence way. These parts are called Areas. The backbone part is called Backbone Area and always indicated by the numerical value 0 (or 0.0.0.0).

By using this hierarchical structure, each device is allowed to keep the link state database in the area where it resides and the topology inside the area is invisible to the outside. In this way, the link state database of each device can be always in a reasonable size, the route calculation time is not too much and the number of packets is not too big.

In the OSPF, the following types of special areas have been defined to meet actual needs:

stub Area.

If an area is at the end of the whole network, then we can design the area as a stub area.

A stub area cannot learn the external routing information of an AS (type 5 LSAs). In practical application, external routing information takes a great proportion in the link state database. Therefore, the devices inside a stub area will learn very little routing information, thus reducing the system resources for running the OSPF protocol.

A device inside a stub area can reach outside of an AS through the default route entry (type3 LSA) generated from the default routing information published by Area Border Routers in the stub area.

NSSA area (Not-So-Stubby Area)

NSSA is the extension of the stub area. By preventing from flooding type 5 LSAs to the devices in the NSSA, it reduce the consumption of device resources. However, unlike a stub area, it allows a certain amount of external routing information of the AS to enter an NSSA in other ways, namely, inject into the NSSA in the form of type 7 LSAs.

So far, the NSSA area functions of the OSPFv3 have not be implemented.

To configure the OSPFv3 area parameters, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **stub** [**no-summary**]	Configure a stub area. no-summary: configure the area to a totally stub area, preventing the area border router in the stub area from sending type3 and type4 LSAs to the stub area.
area *area-id* **default-cost** *cost*	Configure the cost of the default route sent to a stub area.

Use the **no** form of the above command to invalidate the configuration.

You can configure the parameter **default-cost** after the configuration of the stub area is made.If the stub area is changed into an ordinary area, the default-cost configuration will be deleted automatically.

6.1.8 Configuring OSPFv3 Virtual Link

In the OSPF, all areas must be connected to the backbone area to ensure the communication with other areas. If some areas cannot be connected to the backbone area, virtual links are required to connect the backbone area.

To establish a virtual link, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**dead-interval** *seconds*] [**transmit-delay** *seconds*] [**retransmit-interval** *seconds*] [**instance** *instance-id*] [**authentication ipsec spi** *spi* [**md5** | **sha1**] [**0** | **7**] *key*] [**encryption ipsec spi** *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*]	Configure a virtual link. By default, the virtual link is not configured. *area-id:* the ID for the area where the virtual link is. *router-id:* the router-id for the virtual link neighbor. *spi*: security parameter index within the ragne from 256 to 4294967295. **null**: specifies null encryption mode. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key. Other parameters have the same meanings as the interface parameters.

Use the **no** form of the command to invalidate the configuration.

It is not allowed to create a virtual link in the stub area .A virtual link can be taken as a special interface, so its configuration are the same as that of a normal interface. You must ensure that the configuration of **instance, hello-interval**, **dead-interval**, **authentication** and **encryption** configured at the two ends of the virtual link are identical.

6.1.9 Configuring OSPFv3 Route Aggregation

Without route aggregation, each device on the network has to maintain the routing information to every network.By aggrgating some information together, route aggregation can alleviate the burden on the L3 device and network bandwidth. As the size of a network is growing, the importance of route aggregation increases..

Qtech's L3 devices support two types of route aggregation configuration: inter-area route aggregation and external route aggregation.

Configuring Inter-area Route Aggregation

The ABR in an area needs to advertise the routes in an area to other areas. If the route addresses are continual, the ABR aggregates the routing information and then advertises it.

To configure the inter-area route aggregation, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **range** *ipv6-prefix/prefix-length* [**advertise** | **not-advertise**]	Configure inter-area route aggregation. *area-id*: ID of the area for aggregation. *ipv6-prefix/prefixlength*: Set the ipv6 prefix of the aggregated route. **advertise|not-advertise**: Advertise the summary-LSA created by aggregation or not.

Use **no area** *area-id* **range** *ipv6-prefix /prefix-length* to remove the inter-area aggregation configured.

Configuring External Route Aggregation

The route aggregation is allowed when redistributing the generated Type-5 LSA on the ASBR.

To configure the external route aggregation, execute the following commands in the OSPFv3 configuration mode:

Command	Function
summary-prefix *ipv6-prefix / prefix-length* [**not-advertise** | **tag** *tag-value*]	Configure external route aggregation. *ipv6-prefix/prefixlength*: Set the ipv6 prefix of the aggregated route. **not-advertise**: Not advertise the LSA created by aggregation. *tag-value*: The valid range is *<0-4294967295>*, used to specify the tag value for the LSA created by aggregation.

Use **no summary-prefix** *ipv6-prefix/prefix-length* to remove the external route aggregation configured.

6.1.10 Configuring Bandwidth Reference Value of OSPFv3 Interface Metric

The metric for the OSPF protocol is a bandwidth value based on the interface. The cost value of the interface is calculated based on its bandwidth.

For example, if the bandwidth reference value of an interfaces is 100 Mbps and the bandwidth of the network interfaces is 10Mbps, the automatically calculated interface cost is 100/10=10.

Currently, the default reference value of the network interface bandwidth of our products is 100Mbps.

To modify the reference value of the OSPFv3 interface bandwidth, execute the following commands in the OSPFv3 configuration mode:

Command	Function
auto-cost [**reference-bandwidth** *ref-bw*]	Configure the bandwidth reference value for interface metric, in Mbps.

You can run the **ipv6 ospf cost** *cost-value* command in the interface configuration mode to set the cost for a specified interface, which takes precedence over the one calculated based on bandwidth reference value.

6.1.11 Configuring MTU Check of DD Packets Received on OSPFv3 Interfaces

When the OSPFv3 receives the DD(Database Description) packets, it checks whether the MTU for the neighbor interface is the same as the MTU for its own interface. If the former is larger than the latter, the adjacency relationship cannot be established.

By default, this function of MTU check is disabled. To enable the MTU check on an interface, execute the following command in the interface configuration mode:

Command	Function
no ipv6 ospf mtu-ignore [**instance** *instance-id*]	Enable the MTU check on the interface when receiving database description (DD) packets.

By default, the MTU check function of the interface is disabled.

6.1.12 Configuring OSPFv3 Default Route

In the OSPFv3 protocol, the default route can be generated in many ways.

As described in section “Configuring OSPFv3 Area Parameters”, the default route represented by Type-3 LSA will be automatically generated in a stub area.

You can configure a default route represented by Type 5 LSA and advertise it to the whole OSPF AS.

Execute the following command in the OSPFv3 configuration mode:

Command	Function
default-information originate [always] [metric *metric-value*] **[metric-type** *type-value*] **[route-map** *map-name*]	Configure the generation of a default route. **always:** With this parameter configured, no matter what the condition the system routing is in, a default route LSA is always generated. With this parameter not configured, only when the default routing existd in the core routing table, the default route LSA is generated and advertised. *metric:* Initial metric value of the route. The valid range is 0-16777214. **metric-type:** The external routing type corresponding to the default routing. **route-map:** the corresponding route-map rule to set the generated LSA.

Use **no default-information originate** to remove the default routing generated.

This command cannot be configured on the devices in a stub area.

Once configured, the device automatically becomes the ASBR.

6.1.13 Configuring OSPFv3 Routing Redistribution

Routing information redistribution allows the routing information of a routing protocol to be redistributed to another routing protocol.

To configure the OSPFv3 route redistribution, execute the following commands in the OSPFv3 configuration mode:

Command	Function

Command	Function
redistribute {**bgp** | **connected** | **isis** [*area-tag*] | **ospf** *process-id* | **rip** | **static**} [{**level-1** | **level-1-2** | **level-2**} | **match** {**internal** | **external** [**1**|**2**]} | **metric** *metric-value* | **metric-type** {*1*|*2*} | **route-map** *route-map-name* | **tag** *tag-value*]	Redistribute the routing information of other routing protocols. And set the conditions of redistribution. At present, the OSPFv3 supports redistribution of static, connect, rip, bgp, isis and ospf routes. When redistributing ISIS routes, you can configure the level parameter to redistribute the ISIS routes at the specified level. When redistributing OSPF routes, you can configure the match parameter to redistribute the OSPF routes of the specific sub type.
default-metric *number*	Configure the default metric for route redistribution.

Use the **no** redistribute *protocol* to disable the routing information redistribution.

- The isis parameter is not supported by S8600 and S12000 series in v10.4(3b17).

6.1.14 Configuring OSPFv3 Timer

After receiving the notice of network topology changes, the OPSPFv3 routing process will wait for a period of time before starting the SPF calculation. The SPF calculation delay is configurable, you can also use the command to configure the minimum and maximum interval between two SPF calculations.

To configure the OSPFv3 routing calculation timer, execute the following command in the routing process configuration mode:

Command	Function
timers throttle spf *spf-delay spf-holdtime spf-max-waittime*	Configure the OSPFv3 timer of routing calculation, in ms.

The parameter *spf-delay* refers to the delay from the topology change to the beginning of the SPF calculation.

The parameter *spf-holdtime* refers to the minimum interval of the first and the second SPF calculations triggered. Afterwards, the next SFP holdtime shall at least be twice as the last interval till the interval reaches the configured *spf-max-waittime*. If the SPF calculation intervals have exceeded the minimum value, it will re-calculate the SPF calculation interval from the *spf-holdtime.*

In normal conditions, when the link changes occassionally, reducing the *spf-delay* and *spf-holdtime* value can speed up the OSPF convergence. Setting a large *spf-max-waittime* avoids high CPU consumption by OSPF due to the continuous link fluctuation.

For example, **timers throttle spf** *1000 5,000 100,000*

If the topology keeps changing, the SPF calculation intervals(the SPF calculation interval increases by the binary exponential backoff algorithm, but cannot exceed the max-wait-time) are 1s, 6s, 16s, 36s, 76s, 156s, 256s, 256+100, ……

To configure the delay and holdtime for OSPFv3 routing calculation only, execute the following command in the routing process configuration mode:

Command	Function
Qtech (config-router)# timers spf *spf-delay spf-holdtime*	Configure the routing calculation timer in second.

The **timers spf** and **timers throttle spf** commands are overwritten, and the one configured later is valid. With both commands not configured, the default value is **timers throttle spf.**

The function of the command **timers throttle spf** has covered the function of **timers spf**, and is even stronger. It is recommended to use the **timer throttle spf** command.

6.1.15 Configuring OSPFv3 Passive Interface

To prevent other Layer 3 devices in the network from learning the routing information of this device, you can set a network interface to a passive interface in the routing protocol configuration mode

For the OSPFv3 protocol, if a network interface is configured as a passive network interface, then this network interface will receive/send no OSPF message.

To configure an OSPFv3 passive interface, execute the following command in the OSPFv3 configuration mode:

Command	Function
passive-interface {**default** | *interface-type* *interface-number* }	Configure a passive interface. default: with this parameter configured, all interfaces will be set as the passive interfaces. Interface: set the specified interface as the passive interface. Combining **passive-interface default** and **no passive-interface *interface*** can specify some interfaces as non-passive interfaces and the others as passive interfaces.

Use the command **no passive-interface** {*interface-id* | **default**} to remove the passive interface setting.

6.1.16 Configuring OSPFv3 Authenticated Encryption

Authenticated encryption is configured to avoid learning unauthenticated encryption and invalid routes, and prevent valid routes from being announced to the unauthenticated encryption device. On a broadcast network, authenticated encryption can also help avoid the possibility of specifying the unauthenticated encryption device to ensure stability and invasion-resistance of the routing system.

Run the following demands to configure OSPFv3 authenticated encryption in routing process configuration mode or interface configuration mode:

Command	Function
area *area-id* authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	Enables authentication on the area. Sets authentication mode and key.
area *area-id* **encryption ipsec spi** *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*	Enables authenticated encryption on the area. Sets encrypted authentication mode and key.
ipv6 ospf authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	Enables authentication on the interface. Sets authentication mode and key.
ipv6 ospf encryption ipsec spi *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*	Enables authenticated encryption on the interface. Sets encrypted authentication mode and key.

Use the **no** form of this command to disable configuration.

Note

Connected interfaces within the same area must be configured with the same authenticated encryption parameters. Authenticated encryption configured on the area is effective for all interfaces (except the virtual link) within this area but authenticated encryption configured on the interface has a higher priority.

Authenticated encryption parameters configured on two ends of the virtual link must be the same. All *spi* parameters must be unique.

Please refer to *configuring OSPFv3 virtual link* section for virtual link authentication configuration.

6.1.17 Configuring the OSPFv3 Route Management Distance

The route management distance, representing the reliability of the route source, is used to compare the priorities for different routing protocols. The valid range for the management distance is 0-255. The smaller the management distance is, the higher the route priority is, and the higher the route source reliability is.

By default, the OSPFv3 route management distance is 110. You can set different management distances for different OSPFv3 routes, the intra-area, inter-area and external routes.

To change the OSPFv3 route management distance, execute the following command in the routing process configuration mode:

Command	Function
distance {*distance* | **ospf** { **intra-area** *distance* | **inter-area** *distance* | **externa**l *distance* }}	Modify the OSPFv3 route management distance.

The management distance must be used to compare the priorities of the different routes originated from different OSPFv3 processes.

6.2 Configuring the OSPFv3 BFD

Refer to relevant sections in BDF Configuration Guide for the OSPFv3 BFD configuration.

6.2.1 OSPFv3 Debugging & Monitoring

OSPFv3 supports a large range of debugging and monitoring commands.

6.2.1.1 OSPFv3 Debugging Commands

Use the following commands to enable the OSPFv3 process debugging in the privileged configuration mode:

Command	Function
debug ipv6 ospf events	Show the OSPFv3 event information.
debug ipv6 ospf ifsm	Show the state machine events and changes of an egress interface.
debug ipv6 ospf lsa	Show the related OSPFv3 LSA information.
debug ipv6 ospf nfsm	Show state machine events and changes of a neighbor.
debug ipv6 ospf nsm	Show the related OSPFv3 and NSM module information.
debug ipv6 ospf packet	Show the OSPFv3 packet information.
debug ipv6 ospf route	Show the OSPF routing calculation and addition information.

Use the **undebug** form of the above commands to disable the above **debug** commands.

The **debug** commands are provided for technicians.

Running a **debug** command will affect the performance of the system in a certain extent. Therefore, after running a **debug** command, use an **undebug** command to disable the debug command.

6.2.1.2 OSPFv3 Monitoring Commands

Use the following commands to enable the OSPFv3 process monitoring in the privileged configuration mode:

Command	Function
show ipv6 ospf	Show the OSPFv3 process information.
show ipv6 ospf [*process-id*] **database** [**isa-type** [**adv-router** *router-id*]]	Show the database information of the OSPF process.
show ipv6 ospf interface *[interface-type interface-number]*	Show the interface information of the OSPFv3 process.
show ipv6 ospf [*process-id*] **neighbor** [*interface-type interface-number* [**detail**]] *[neighbor-id*] [**detail**]	Show the neighbor information of the OSPFv3 process.
show ipv6 ospf [*process-id*] **route**	Show the OSPFv3 routing information.
show ipv6 ospf [*process-id*]**summary-prefix**	Show the OSPFv3 external route summary information
show ipv6 ospf [*process-id*] **topology** [**area** *area-id*]	Show each area topology of the OSPFv3.
show ipv6 ospf [*process-id*] **virtual-links**	Show the virtual link information of the OSPFv3 process.

6.3 OSPFv3 Configuration Examples

6.3.1 OSPFv3 Basic Configuration Example

The following configuration example shows the commands related to OSPF configuration.

Topological Diagram

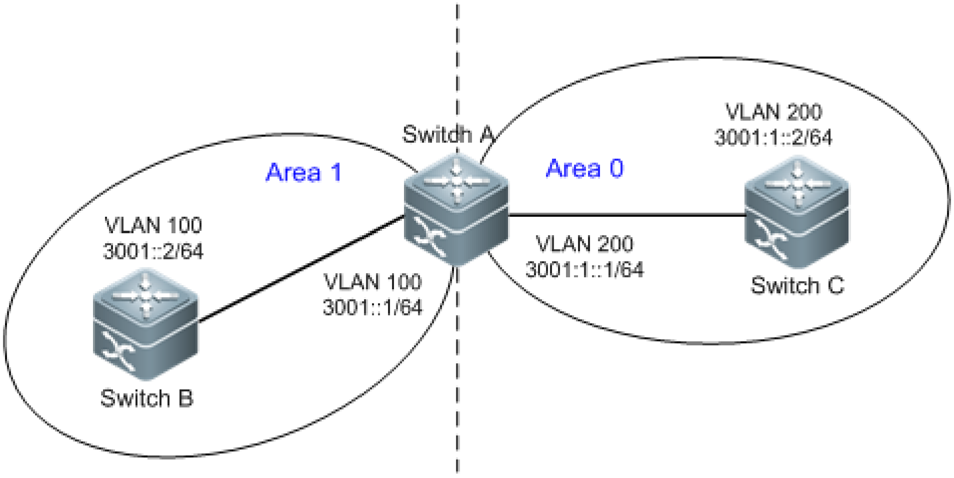


Figure 1 OSPFv3 basic configuration

SwitchA and SwitchB belong to Area 0, while Switch A and Switch C belong to Area 1. The intercommunication between three switches is realized via the vlan interface.

6.3.1.1 Application Requirements

Enable the OSPFv3 on all switches and divide them into two areas between which IPv6 packets can be communicated.

Configuration Tips

Key points

Configure Area0 and Area1, and enable OSPFv3 on the corresponding VLAN interface of the switch (interface vlan 100 or vlan 200 of SwtichA/SwitchB/SwitchC in this example)

Cautions

The router-id must be specified, or else the adjacency cannot be created. Automatic acquisition of router-id is supported in 10.4 and subsequent releases.

Vlan must be created first, or else the VLAN interface cannot join OSPFv3.

Configuration Steps

Configuring SwitchA

Step 1, Create a VLAN and set the IPv6 address

```
SwitchA# config terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#exit
SwitchA(config-vlan)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

! Create and configure interface vlan200

```
SwitchA(config)#vlan 200
SwitchA(config-vlan)#interface vlan 200
SwitchA(config-if-VLAN 200)#ipv6 enable
SwitchA(config-if-VLAN 200)#ipv6 address 3001:1::1/64
SwitchA(config-if-VLAN 200)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf  10
SwitchA(config-router)#router-id 1.1.1.1
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchA(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchA(config-if-VLAN 100)#exit
```

Step 4, Enable the OSPFv3 on interface vlan 200, with the area being Area1

```
SwitchA(config)#interface vlan 200
SwitchA(config-if-VLAN 200)#ipv6 ospf 10 area 1
SwitchA(config-if-VLAN 200)#end
```

Configuring SwitchB

Step 1, create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchB(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchB(config-if-VLAN 100)#end
```

Configuring SwitchC

Step 1, create a VLAN and configure the IPv6 address

SwitchC#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan200

```
SwitchC(config)#vlan 200
SwitchC(config-vlan)#interface vlan 200
SwitchC(config-if-VLAN 200)#ipv6 enable
SwitchC(config-if-VLAN 200)#ipv6 address 3001:1::2/64
SwitchC(config-if-VLAN 200)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchC(config)#ipv6 router ospf 10
SwitchC(config-router)#router-id 3.3.3.3
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchC(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 200, with the area being Area1

```
SwitchC (config)#interface vlan 200
SwitchC (config-if-VLAN 200)#ipv6 ospf 10 area 1
SwitchC (config-if-VLAN 200)#end
```

Verifying configuration

Step 1: Verify whether the configuration are correct. Pay attention: whether the router-id is specified, whether the OSPFv3 is enabled on the interface, and whether such parameters as OSPFv3 timer are identical in the same area.

Configuring SwitchA

```
vlan 100
!
vlan 200
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
```

```
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::1/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

Configuring SwitchB

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

Configuring SwitchC:

```
vlan 200
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::2/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 3.3.3.3
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 2 Neighbors, 2 is Full:
Neighbor ID   Pri  State        Dead Time   Instance ID   Interface
2.2.2.2        1   Full/BDR      00:00:37    0             VLAN 100
3.3.3.3        1   Full/DR       00:00:34    0             VLAN 200
```

The information displayed on SwitchB and SwitchC is similar to the information displayed on SwitchA.

Step 3: Display OSPFv3 routes and ping IPv6 address in another area. Pay attention: whether all the IPv6 routes are learned and whether the routes can be pinged.

```
SwitchC#show ipv6 route
IPv6 routing table name is Default(0) global scope - 7 entries
```

```
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra area, OI - OSPF inter area,  OE1 - OSPF external type 1, OE2 -
OSPF external type 2
       ON1 - OSPF NSSA external type 1, ON2 - OSPF NSSA external type 2
       [*] - NOT in hardware forwarding table
L      ::1/128  via Loopback, local host
OI     3001::/64  [110/2] via FE80::21A:A9FF:FE15:4CB9, VLAN 200
C      3001:1::/64  via VLAN 200, directly connected
L      3001:1::2/128  via VLAN 200, local host
L      FE80::/10  via ::1, Null0
C      FE80::/64  via VLAN 200, directly connected
L      FE80::21A:A9FF:FE01:FB1F/128  via VLAN 200, local host
```

```
SwitchC#ping ipv6 3001::2
Sending 5, 100-byte ICMP Echoes to 3001::2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

The information displayed on SwitchA and SwitchB is similar to the information displayed on SwitchC.

6.3.2 OSPFv3 Redistribution Configuration Example

Configuration Requirements

There are three devices which are connected as shown in Figure 2.

- Enable the OSPFv3 protocol on RouterA; Enable BGP protocol and configure the static route on RouterC; For RouterB, redistribute the static route redistributed on RouterC to the OSPFv3 domain. Set the specified community attributes for the static route redistributed to BGP on RouterC and redistribute BGP route with the specified community attributes to the OSPFv3 domain on RouterB.
- Configure the external route summary on RouterB: aggregate the routes within the range of 2001:db8:77::/48 and advertise the summary to the OSPFv3 domain.
- To speed up the convergence, set the SPF calculation delay, holdtime and max-waittime for RouterA and RouterB to 5ms, 1000ms and 90000ms respectively.

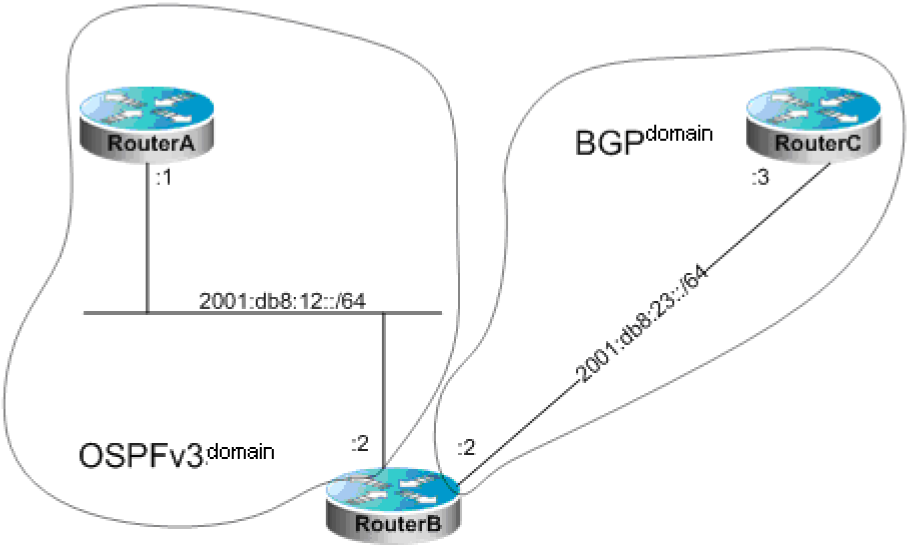


Figure 2 OSPFv3 Redistribution Configuration Example

6.3.2.1 Configuration Details

Router A configuration:

# Configure the network interface

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:12::1/64
Qtech(config-if)# ipv6 ospf 12 area 0
```

# Configure OSPFv3

```
Qtech(config)# ipv6 router ospf 12
Qtech(config-router)# router-id 1.1.1.1
Qtech(config-router)# timers throttle spf 5 1000 90000
```

Router B Configuration:

# Configure the network interface

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:12::2/64
Qtech(config-if)# ipv6 ospf 12 area 0
Qtech(config)# interface gigabitEthernet 0/2
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:23::2/64
```

# Configure OSPFv3

```
Qtech(config)# ipv6 router ospf 12
Qtech(config-router)# router-id 2.2.2.2
Qtech(config-router)# redistribute bgp route-map ospfrm
Qtech(config-router)# timers throttle spf 5 1000 90000
Qtech(config-router)# summary-prefix 2001:db8:77::/48
```

# Configure BGP

```
Qtech(config)# router bgp 2
Qtech(config-router)# neighbor 2001:db8:23::3 remote-as 3
Qtech(config-router)# address-family ipv6
Qtech(config-router-af)# neighbor 2001:db8:23::3 activate
```

# Configure route-map

```
Qtech(config)# route-map ospfrm
Qtech(config-route-map)# match community cl_110
```

# Define community list

```
Qtech(config)# ip community-list standard cl_110 permit 22:22
```

Router C Configuration:

# Configure the network interface

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address 2001:db8:23::3/64
```

# Configure BGP

```
Qtech(config)# router bgp 3
Qtech(config-router)# neighbor 2001:db8:23::2 remote-as 2
Qtech(config-router)# address-family ipv6
Qtech(config-router-af)# redistribute static route-map bgprm
Qtech(config-router-af)# neighbor 2001:db8:23::2 activate
Qtech(config-router-af)# neighbor 2001:db8:23::2 send-community
```

# Configure static route

```
Qtech(config)# ipv6 route 2001:db8:77:88::/64 null 0
Qtech(config)# ipv6 route 2001:db8:77:99::/64 null 0
```

# Configure route-map

```
Qtech(config)# route-map bgprm
Qtech(config-route-map)# set community 22:22
```

6.3.3 Example of Stub Area Configuration

6.3.3.1 Typology Diagram

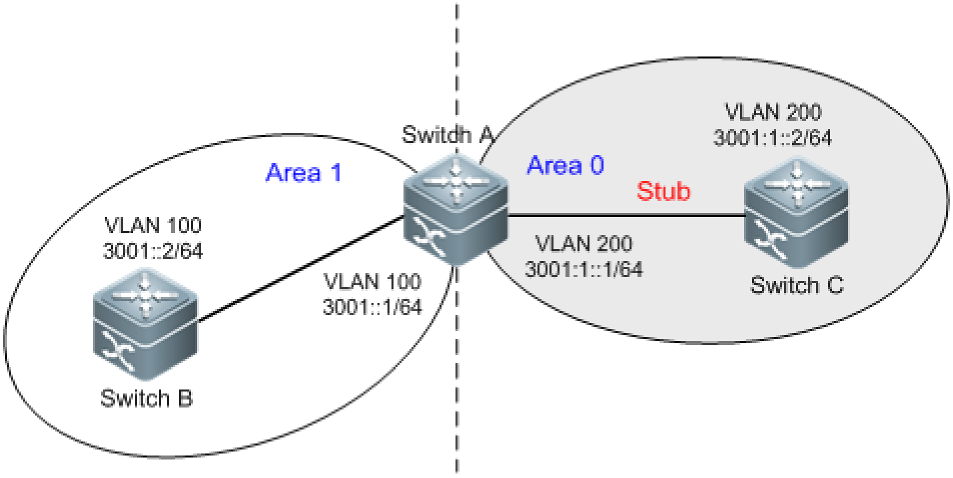


Figure 3 OSPFv3 stub area (the same as Figure 1)

6.3.3.2 Application Requirements

Configure Area 1 as a stub area in order to reduce the system overhead of switches in this area.

6.3.3.3 Configuration tips

Use the parameter of "stub no-summary" on the area border router (ABR) (Switch A in this example)

Use the parameter of "stub" on the non-area-border router (Switch C in this example)

6.3.3.4 Configuration Steps

Configuring SwitchA

Step 1: Enable OSPFv3 basic configuration, as in the OSPFv3 basic configuration example.

Step 2: Configure stub no-summary

```
SwitchA# conf
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#area 1 stub no-summary
SwitchA(config-router)#exit
```

Configuring SwitchC

Step 1: Enable OSPFv3 basic configuration, as in the OSPFv3 basic configuration example.

Step 2: Configure stub

```
SwitchC# conf
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#area 1 stub
SwitchA(config-router)#exit
```

6.3.3.5 Verifying configuration

Step 1: Verify whether the configuration are correct. While making sure the OSPFv3 basic configuration are correct, pay attention to the differences in stub parameters between ABR and the other router.

SwitchA Configuration

```
vlan 100
!
vlan 200
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::1/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
area 1 stub no-summary
!
```

SwitchC Configuration

```
vlan 200
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::2/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
```

```
 router-id 3.3.3.3
area 1 stub
!
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 2 Neighbors, 2 is Full:
Neighbor ID   Pri  State       Dead Time   Instance ID   Interface
2.2.2.2        1   Full/BDR     00:00:37    0             VLAN 100
3.3.3.3        1   Full/DR      00:00:34    0             VLAN 200
```

Similar information will be displayed on SwitchC.

Step 3: Display OSPFv3 routes. Pay attention: whether the default route is generated, and whether the inter-area route exists

```
SwitchC #show ipv6 route
IPv6 routing table name is Default(0) global scope - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
  I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
  O - OSPF intra area, OI - OSPF inter area,  OE1 - OSPF external type 1, OE2 - OSPF
external type 2
ON1 - OSPF NSSA external type 1, ON2 - OSPF NSSA external type 2
       [*] - NOT in hardware forwarding table
OI     ::/0  [110/2] via FE80::21A:A9FF:FE15:4CB9, VLAN 200
L      ::1/128  via Loopback, local host
C      3001:1::/64  via VLAN 200, directly connected
L      3001:1::2/128  via VLAN 200, local host
L      FE80::/10  via ::1, Null0
C      FE80::/64  via VLAN 200, directly connected
L      FE80::21A:A9FF:FE01:FB1F/128  via VLAN 200, local host
```

6.3.4 Example of OSPFv3 DR Election Configuration

6.3.4.1 Typology Diagram

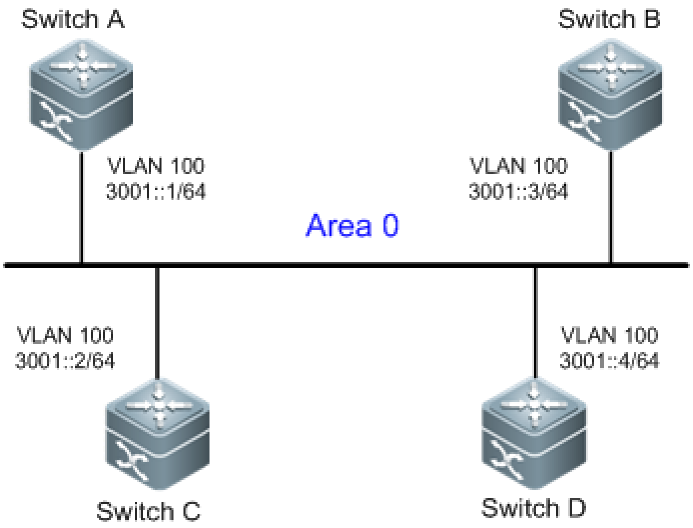


Figure 4 OSPFv3 DR election

SwitchA, SwitchB, SwitchC and SwitchD are in the same area (Area 0) and are interconnected via vlan 100. Switch A and Switch B are devices with the best configuration and the highest stability on the network

6.3.4.2 Application Requirements

Corresponding requirements: By adjusting the parameter **priority**, configure SwitchA as the DR and SwitchB as the BDR in order to avoid network route flap.

6.3.4.3 Configuration Tips

Configuration tips

Configure the priority of the interface of expected DR (Switch A in this example) to the highest (150 in this example) and the priority of the interface of BDR (Switch B) to the second highest (50 in this example)

Cautions

The default priority of interface is 1, and DR/BDR can be determined according to the router-id. Generally, the router with the largest router-id will be the DR, and the router with the second largest router-id will be the BDR.

6.3.4.4 Configuration Steps

Configuring SwitchA

Step 1, create a VLAN and configure the IPv6 address

SwitchA# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#exit
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the priority being 150

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchA(config-if-VLAN 100)# ipv6 ospf priority 150
SwitchA(config-if-VLAN 100)#end
```

Configuring SwitchB

Step 1, create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#exit
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

! Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
SwitchB(config-router)#exit
```

Step 2, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the priority being 50

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchB(config-if-VLAN 100)# ipv6 ospf priority 50
SwitchB(config-if-VLAN 100)#end
```

Configuring SwitchC

Step 1, create a VLAN and configure the IPv6 address

SwitchC# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchC(config)#vlan 100
SwitchC(config-vlan)#exit
SwitchC(config)##interface vlan 100
SwitchC(config-if-VLAN 100)#ipv6 enable
SwitchC(config-if-VLAN 100)#ipv6 address 3001::3/64
SwitchC(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchC(config)#ipv6 router ospf 10
SwitchC(config-router)#router-id 3.3.3.3
SwitchC(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0 and the priority using the default value.

```
SwitchC(config)#interface vlan 100
SwitchC(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchC(config-if-VLAN 100)#end
```

Configuring SwitchD

Step 1, create a VLAN and configure the IPv6 address

SwitchD# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchD(config-vlan)#vlan 100
SwitchD(config-vlan)#exit
SwitchD(config)#interface vlan 100
SwitchD(config-if-VLAN 100)#ipv6 enable
SwitchD(config-if-VLAN 100)#ipv6 address 3001::4/64
SwitchD(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchD(config)#ipv6 router ospf 10
SwitchD(config-router)#router-id 4.4.4.4
SwitchD(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0 and the priority using the default value.

```
SwitchD(config)#interface vlan 100
SwitchD(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchD(config-router)#end
```

6.3.4.5 Verifying configuration

Step 1: Verify whether the configuration are correct. Pay attention: whether the OSPF basic parameters and interface priority are correct

SwitchA Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
ipv6 ospf priority 150
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

SwitchB Configuration

```
vlan 100
!
```

```
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
 ipv6 ospf 10 area 0
ipv6 ospf priority 50
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

SwitchC Configuration

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::3/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 3.3.3.3
```

SwitchD Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::4/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 4.4.4.4
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created, and whether each switch plays the correct role.

```
SwitchD#show ipv6 ospf neighbor
Neighbor ID  Pri   State     Dead Time   Interface ID    Interface
3.3.3.3       1    2WAY/DROTHER  00:00:33    4196           Vlan100
1.1.1.1      150    FULL/DR      00:00:35    4196             Vlan100
2.2.2.2      50    FULL/BDR      00:00:35    4196             Vlan100
```

Adjacencies before priority configuration are shown below. We can see that DR/BDR can be specified by adjusting the priority.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 3 Neighbors, 2 is Full:
Neighbor ID  Pri   Stat       Dead Time   Instance ID   Interface
```

```
2.2.2.2         1    Full/BDR         00:00:33     0             VLAN 100
3.3.3.3         1    2-Way/DROther    00:00:35    0             VLAN 100
4.4.4.4         1    Full/DR          00:00:33     0             VLAN 100
```

6.3.5 Configuration Example of OSPFv3 Multiple Instances on One Link

6.3.5.1 Topological Diagram

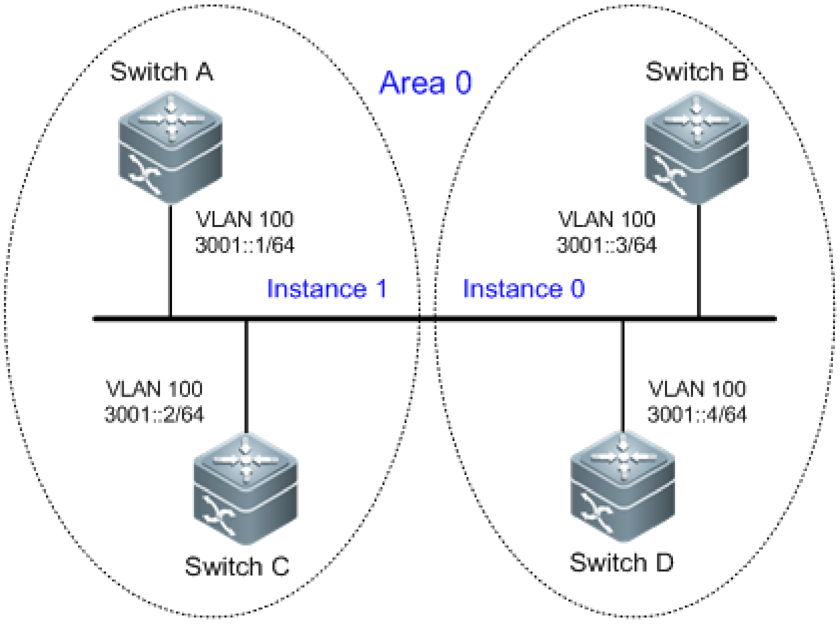


Figure 1-5 Multiple instances on one link

SwitchA, SwitchB, SwitchC and SwitchD are in the same area (Area 0) and are interconnected via vlan 100.

6.3.5.2 Application Requirements

On a broadcast link, especially within a vlan, adjacencies will be established among all the switches. This may result in increased system overhead and network oscillation.

Application requirements: Switches in the same area are divided into several groups, and OSPFv3 adjacencies can only be established between switches belonging to the same group.

6.3.5.3 Configuration Tips

Key points

By configuring multiple instances on the same link (the link on interface vlan100 in this example), adjacency establishment by group can be implemented (in this example, SwitchA and SwitchB form a group, with instance ID being 1; SwitchC and SwitchD form a group, with instance ID being 0).

Cautions

By default, the instance ID of the interface is 0. In this example, you only need to configure the instance ID on Switch A and Switch B.

6.3.5.4 *Configuring Steps*

Configuring SwitchA

Step 1, Create a VLAN and configure the IPv6 address

SwitchA# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the instance ID being 1

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0 instance 1
SwitchA(config-if-VLAN 100)# end
```

Configuring SwitchB

Step 1, Create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
SwitchB(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the instance ID being 1

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0 instance 1
SwitchB(config-if-VLAN 100)# end
```

6.3.5.5 *Verifying Configuration*

Step 1: Verify whether the configuration is correct. Pay attention: whether the instance ID for establishing adjacency between switches is correct.

SwitchA Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0 instance 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

SwitchB Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
ipv6 ospf 10 area 0 instance 1
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

SwitchC Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::3/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 3.3.3.3
```

SwitchD Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::4/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
```

```
 router-id 4.4.4.4
```

Step 2: Display the instance ID of the interface link and reconfirm that the switches in the same group have the same instance ID

```
SwitchA#show ipv6 ospf interface vlan 100
VLAN 100 is up, line protocol is up
  Interface ID 4196
  IPv6 Prefixes
    fe80::21a:a9ff:fe15:4cb9/64 (Link-Local Address)
    3001::1/64
  OSPFv3 Process (10), Area 0.0.0.0, Instance ID 1
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State BDR, Priority 1
    Designated Router (ID) 2.2.2.2
      Interface Address fe80::2d0:f8ff:fe22:88b1
    Backup Designated Router (ID) 1.1.1.1
      Interface Address fe80::21a:a9ff:fe15:4cb9
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:08
    Neighbor Count is 1, Adjacent neighbor count is 1
  Hello received 7 sent 8, DD received 3 sent 5
  LS-Req received 1 sent 1, LS-Upd received 5 sent 4
  LS-Ack received 3 sent 3, Discarded 0
```

```
SwitchB#show ipv6 ospf interface vlan 100
VLAN 100 is up, line protocol is up
  Interface ID 4196
  IPv6 Prefixes
    fe80::2d0:f8ff:fe22:88b1/64 (Link-Local Address)
    3001::2/64
  OSPFv3 Process (10), Area 0.0.0.0, Instance ID 1
    Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State DR, Priority 1
    Designated Router (ID) 2.2.2.2
      Interface Address fe80::2d0:f8ff:fe22:88b1
    Backup Designated Router (ID) 1.1.1.1
      Interface Address fe80::21a:a9ff:fe15:4cb9
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:05
    Neighbor Count is 1, Adjacent neighbor count is 1
  Hello received 16 sent 21, DD received 10 sent 8
  LS-Req received 2 sent 2, LS-Upd received 10 sent 9
  LS-Ack received 6 sent 6, Discarded 0
```

Step 3: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created, and whether the adjacencies are established only between the switches in the same group.

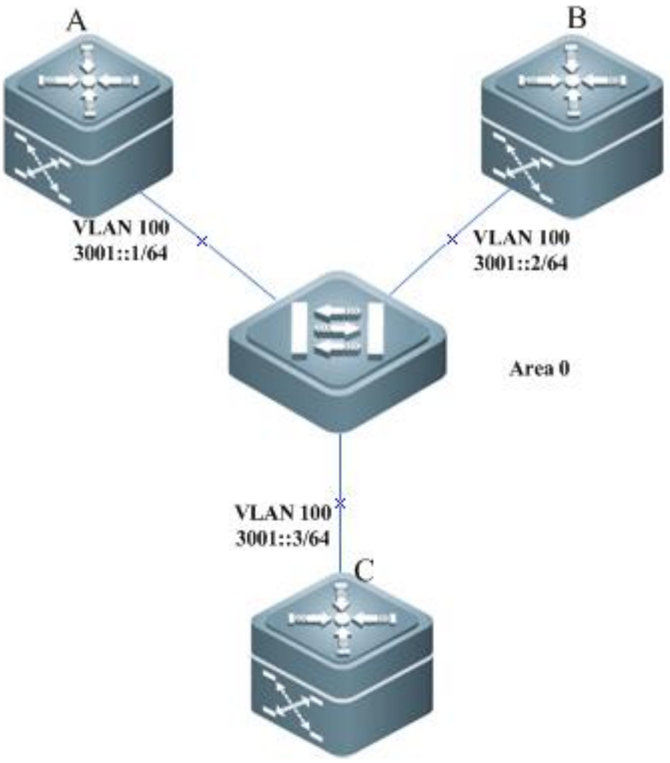
```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 1 Neighbors, 1 is Full:
Neighbor ID  Pri State       Dead Time    Instance ID   Interface
2.2.2.2        1    Full/DR    00:00:39      1             VLAN 100
```

```
SwitchB#show ipv6 ospf neighbor
OSPFv3 Process (10), 1 Neighbors, 1 is Full:
Neighbor ID  Pri  State         Dead Time    Instance ID   Interface
1.1.1.1        1    Full/BDR     00:00:34     1              VLAN 100
```

6.3.6 Configuration Example of OSPFv3 Authenticated Encryption

6.3.6.1 Topological Diagram

Figure 1-6 Encrypted authentication instance



6.3.6.2 Application Requirements

The OSPFv3 protocol runs on Device A, B and C, which belong to the same area. Connected interfaces establish adjacency only if the same authenticated encryption is configured on them.

The interfaces connecting Device A and B are configured with the same authentication parameters. Device C is not configured with authentication. Device A establishes adjacency with Device B and no adjacency with Device C.

6.3.6.3 Configuriation Tips

According to the topological requirements, the OSPFv3 protocol is configured to run on Device A, B and C, which belong to the same area

The Interfaces connecting Device A and B are configured with the same authentication parameters.

6.3.6.4 Configuration Steps

- Configuring Switch A

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Qtech(config)# vlan 100
Qtech(config-vlan)# exit
Qtech(config)# interface vlan 100
Qtech(config-if-vlan 100)# ipv6 address 3001::1/64
Qtech(config-if-vlan 100)# ipv6 ospf 1 area 0
Qtech(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Qtech(config-if-vlan 100)# exit
```

- Configuring Switch B

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Qtech(config)# vlan 100
Qtech(config-vlan)# exit
Qtech(config)# interface vlan 100
Qtech(config-if-vlan 100)# ipv6 address 3001::2/64
Qtech(config-if-vlan 100)# ipv6 ospf 1 area 0
Qtech(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Qtech(config-if-vlan 100)# exit
```

- Configuring Switch C

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Qtech(config)# vlan 100
Qtech(config-vlan)# exit
Qtech(config)# interface vlan 100
Qtech(config-if-vlan 100)# ipv6 address 3001::2/64
Qtech(config-if-vlan 100)# ipv6 ospf 1 area 0
Qtech(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Qtech(config-if-vlan 100)# exit
```

6.3.6.5 Verifying Configuration

- Step 1: Run the **show ipv6 ospf neighbor** command on Device A, B and C. It is shown that each two devices establish adjacency.
- Step 2: Neighbors.are shown only on Device A and B. Device C shows no neighbor. Run the **debug ipv6 ospf packet** command. Device A and B can receive packets from each other while Device C cannot receive packets.

```
Qtech#show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID   Pri   State      Dead Time   Instance ID   Interface
2.2.2.2       1     Full/BDR   00:00:37    0             VLAN 100

Qtech#show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID   Pri   State      Dead Time   Instance ID   Interface
```

```
1.1.1.1       1     Full/BDR    00:00:37    0             VLAN 100

Qtech#show ipv6 ospf neighbor
OSPFv3 Process (1), 0 Neighbors, 0 is Full:
```

7 CONFIGURING BGP

7.1 About BGP

The border gateway protocol (BGP) is an exterior gateway protocol (EGP) used for routers to communicate with one another in different autonomous systems. The protocol is designed to exchange information about network reachability among these autonomous systems (AS) and eliminate loops based on the characteristics of the BGP protocol.

The BGP protocol relies on the TCP protocol for reliable packet transmission.

A router which operates on the BGP protocol is referred to as a “BGP Speaker”, and BGP Speakers that have set up a BGP session are referred to as “BGP Peers”.

There are two modes of BGP session: IBGP (Internal BGP) and EBGP (External BGP). The IBGP refers to a BGP session in an AS, while the EBGP refers to a BGP session between different ASs. To summarize, the EBGP exchanges routing information among different ASs. The IBGP transmits routing information in an AS.

The BGP protocol has the following features:

- ■ Supports BGP-4
- ■ Supports path attributes
- ✓ ORIGN Attribute
- ✓ AS_PATH Attribute
- ✓ NEXT_HOP Attribute
- ✓ MULTI_EXIT_DISC Attribute
- ✓ LOCAL-PREFERENCE Attribute
- ✓ ATOMIC_AGGREGATE Attribute
- ✓ AGGREGATOR Attribute
- ✓ COMMUNITY Attribute
- ✓ ORIGINATOR_ID Attribute
- ✓ CLUSTER_LIST Attribute
- ✓ AS4_PATH Attribute
- ✓ AS4_AGGREGATOR Attribute
- ✓ Connector Attribute
- ■ Supports BGP peer groups
- ■ Supports loopback interface
- ■ Supports MD5 authentication of TCP
- ■ Supports the synchronization of BGP and IGP
- ■ Supports the aggregation of BGP routes
- ■ Supports BGP route flap dampening
- ■ Supports BGP routing reflector
- ■ Supports AS confederation
- ■ Supports BGP soft reset
- ■ Supports BGP Graceful Restart (defined in RFC4724)

7.2 Enabling the BGP Protocol

To enable the BGP protocol, execute the following commands in privileged EXEC mode:

Command	Function
Qtech# configure terminal	Enters global configuration mode.
Qtech(config)# **ip routing**	Enables the routing function (if the switch is disabled).
Qtech(config)# **router bgp** *as-number*	Enables the BGP and configures the AS number. The range of *AS-number* is 1 to 65535.
Qtech(config-router)# **bgp router-id** *router-id*	(Optional) Configures the ID used when this switch runs the BGP protocol.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech# **show run**	Shows current configuration.

Command	Function
Qtech# copy running-config startup-config	Saves the configuration.

Use the **no router bgp** command to disable the BGP protocol**.**

7.3 Default BGP Configuration

The BGP protocol is not enabled by default.

After the BGP protocol is enabled, the default BGP configuration is shown as follows:

Feature		Default Setting
Router ID		To configure a loopback interface, select the maximum address from loopback interface addresses. Otherwise, select the maximum interface address from the directly connected interface.
Synchronization of BGP and IGP		Enabled
Generation of Default Route		Disabled
Multi hops of EBGP	Status	Off
	Number of hops	255
TCP MD5 Authentication		Disabled
Timer	Keepalive Time	60 seconds
	Holdtime	180 seconds
	ConnectRetry Time	120 seconds
	AdvInterval(IBGP)	15 seconds
	AdvInterval(EBGP)	30 seconds
Path Attribute	MED	0
	LOCAL_PREF	100
Route Aggregate		Off
Route Flap Dampening	Status	Off
	Suppress Limit	2000
	Half-life-time	15 minutes
	Reuse Limit	750
	Max-suppress-time	4*half-life-time
Route Reflector	Status	Off
	Cluster ID	Undefined
	Route among reflection clients	Enabled
AS Confederation		Off
Soft Reset		Off
Traceful Restart		Disabled
Management Distance	External-distance	20
	Internal-distance	200
	Local-distance	200

7.4 Injecting Routing Information into the BGP Protocol

The BGP protocol has no routing information when running for the first time. There are two ways to inject routing information to the BGP:

Manually inject routing information to the BGP by using the **network** commands.

Inject routing information to the BGP from the IGP protocol through interaction with the IGP protocol.

The BGP will advertise the injected routing information to its neighbors. This section outlines the manual injection of routing information. For injecting routing information from the IGP protocol, refer to the *Configuration of BGP and IGP Interaction* in related sections.

To manually inject network information advertised by the BGP Speaker to other BGP Speaker, execute the following commands in BGP configuration mode:

Command	Function
Router(config-router)# **network** *network-number* **mask** *network-mask*[**route-map** *map-tag*]	Configures the network whose routing information will be injected into the BGP routing table.

Use the **no network** *network-number* **mask** *network-mask* command to remove the configuration. To cancel the used route-map, reconfigure it by using the *route-map not added* option. If the configured network information comes under standard class A, class B or class C network address, the mask option of this command cannot be used.

In BGP4+, you can use this command in IPv6 address family configuration mode to configure IPv6 routes.

⚠ Caution

■ The **network** command is used to inject IGP routes into the routing table of BGP, and the advertised networks can be direct-connected, static and dynamic routes.

■ For the external gateway protocol (EGP), the **network** command indicates the network to be advertised. This is different from the internal gateway protocol (IGP, such as OSPF and RIP). The IGP uses the **network** commands to determine where the routing update message will be sent to.

Sometimes, you may need to use an IGP route rather than an EBGP route. This can be done by using the **network backdoor** command. Execute the following operations in BGP configuration mode:

Command	Function
Qtech(config-router)# **network** *network-number* **mask** *network-mask* **backdoor**	Sets the backdoor route.

Use the **no network** *network-number* **mask** *network-mask* **backdoor** command to remove the configuration.

⚠ Caution

By default, the distance for network information management learned from the BGP Speakers which have established the EBGP connection is 20. Set the distance by using the **network backdoor** command as 200. As such, identical network information learned from the IGP presents a higher priority. The networks learned from the IGP are considered as backdoor networks, and will not be advertised.

7.4.1 Controlling Route Advertisement

The BGP protocol can control the routes advertised to the core routing table by using the **table-map** command. If a route is matched, the command modifies its attribute and advertises it. If a route is not matched or denied, the command advertises it without modifying its attribute.

By default, the **table-map** command advertises all routes without modifying their attributes.

To configure the **table-map** command, execute it in BGP configuration mode or IPv4 address family confiugration mode:

Command	Function
Router(config-router)# **table-map** *route-map-name*	Configures table-map. route-map-name indicates the name of the route-map you want to associate.

Use the **no table-map** command to remove the configuration.

For the configuration of the **table-map** command to take effect immediately, run the **clear ip bgp** [vrf *vrf-name*] **table-map** command to update the core routing table. The **clear ip bgp** [vrf *vrf-name*] **table-map** command will clear but add the routes in the core routing table. Instead, it uses the table-map to advertise route update messages without causing forwarding oscillation..

The **table-map** command supports the following options: rules-match, as-path/community/ip address/ip next-hop/metric/origin/route-type, set, metric/tag/next-hop.

7.4.2 Controlling the Route Redistribution from IBGP to IGP

The BGP protocol controls the redistribution of routes learnded from the IBGP protocol to IGP protocol by using the **bgp redistribute-internal** command. The routes learned from the EBGP protocol or confederation can be redistriubted to the IGP protocol.

This command is enabled by default in either VRF or global mode. Specifically, routers learned from the IBGP can be redistributed to the IGP protocol.

To redistribute a route to the IGP protocol (inclding RIP/OSPF/ISIS), execute the following command in BGP configuration mode, IPv4/IPv6 address family configuration mode or IPv4 VRF address family configuraiton mode:

Command	Function
Router(config-router)# **bgp-redistirbute-internal**	Redistributes IBGP routes to the IGP protocol.

Use the **no bgp redistribute-internal** command to remove the configuration.

7.5 Configuring BGP Peer (Group) and Its Parameters

Since the BGP is an external gateway protocol (EGP), it is necessary for a BGP Speaker to know who its peer (BGP Peer) is.

As mentioned in the overview of the BGP protocol, two modes can be used to set up the connection among BGP Speakers: IBGP (Internal BGP) and EBGP (External BGP). The protocol determines which connection will be established among BGP Speakers by using the AS of BGP Peer and BGP Speakers.

The BGP protocol supports IPv4 and IPv6. To check IPv6 function, verify whether the **address-family ipv6** command is executed in BGP configuration mode. Otherwise, IPv6 is not supported. An IPv4 address represents an IPv4 neighbor. An IPv6 address represents an IPv6 neighbor. Note that you should activate neighbors in the right address family.

In general, BGP Speakers with EBGP connection should be physically connected. BGP Speakers with IBGP connection, however, can be located anywhere within an AS.

To configure the BGP peer, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name* } **remote-as** *as-number*	Configures the BGP peer. *address* indicates the IP addresses of the BGP peer. *peer-group-name* indicates the name of the BGP peer group. The range of *as-number* is 1 to 65535.

Use the **no neighbor** {*address*\|*peer-group-name*} to delete one peer or peer group.

The BGP Speakers have some configurations in common (including the executed routing policy). To simplify configuration and improve efficiency, it is recommended that you use the BGP peer group.

To configure the BGP peer group, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** *peer-group-name* **peer-group**	Creates a BGP peer group.
Qtech(config-router)# **neighbor** *peer-group-name* **remote-as** *as-number*	(Optional) Configures the BGP peer group. The range of *as-number* is 1 to 4294967295.
Qtech(config-router)# **neighbor** *address* **peer-group** *peer-group-name*	(Optional) Sets the BGP peer as the member of the BGP peer group.

Use the **no neighbor** *address* **peer-group** to delete some members of the BGP peer group.

Use the **no neighbor** *peer-group-name* **peer-group** to delete the entire peer group.

Use the **no neighbor** *peer-group-name* **remote-as** to delete all members of the BGP peer group and AS numbers of the peer group.

To configure the peer of the BGP Speakers or the optional parameter of the BGP peer group, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router-af)# **neighbor** {*address* | *peer-group-name*} **activate**	(Optional) Activates the address family of the neighbor so that the router can exchange routing information with the address family.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **update-source** *interface*	(Optional) Configures the network interfaces to establish the BGP session with specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **ebgp-multihop** [*ttl*]	(Optional) Allows you to establish a BGP session among non-direct-connected EBGP peers (group). The range of TTL is 1 to 255, the EBGP is one hop by default, and the IBGP is 255 hops by default.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **password** *string*	(Optional) Enables the TCP MD5 authentication when the connection is established among specified BGP peer (group), and configures the password.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **times** *keepalive holdtime*	(Optional) Configures the Keepalive and Holdtime value to establish a connection with the specified BGP peer (group). The range of the *keepalive* is 0 to 65535 seconds, 60 seconds by default. The range of the *holdtime* is 0 to 65535 seconds, 180 seconds by default.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **advertisemet-interval** *seconds*	(Optional) Configures the minimal time interval of sending the routing update message to the specified BGP peer (group). The range of advertisement-interval is 1 to 600 seconds, 15 seconds for the IBGP peer by default, and 30 seconds for the EBGP peer by default.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **default-originate** [**route-map** *map-tag*]	(Optional) Configures the router to send a default route to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **next-hop-self**	(Optional) Configures the router to set the next routing information as this BGP speaker when the route is distributed to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **remove-private-as**	(Optional) Configures the router to delete the private AS number in the AS path attribute when distributing the routing information to the EBGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **send-community**	(Optional) Configures the router to send the community attribute to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **maximum-prefix** *maximum* [**warning-only**]	(Optional) Limits the number of the messages received from the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **distribute-list** *access-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing police according to the access control list when routing information is received from and sent to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the prefix list when the routing information is received from and sent to specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **filter-list** *path-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the AS path list when the routing information is received from and sent to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **unsuppress-map** *map-tag*	(Optional) Configures the router to selectively advertise the routing information suppressed by the **aggregate-address** command previously when it is distributed to the specified BGP peer.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **soft-reconfiguration** inbound	(Optional) Restarts the BGP session and reserve the unchanged routing information sent by the BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **route-reflector-client**	(Optional) Configures this switch as the route reflector and specify its client.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **shutdown**	(Optional) Disables the BGP peer (group).

Use the **no** form of the above commands to disable the configurations.

If a peer does not support **remote-as**, each of its members can use the **neighbor remote-as** command to configure it independently.

By default, each member of the BGP peer group will inherit all its configurations. However, each member can support the optional configurations without affecting the output update independently to replace the unified configuration of the BGP peer group.

⚠ Caution

Each member of the BGP peer group can support the optional configurations without affecting the output update independently to replace the unified configuration of the BGP peer group. That is to say, each member of the BGP peer group will inherit the following configurations: **remote-as**, **update-source**, **local-as**, **reconnect-interval**, **times**, **advertisemet-interval**, **default-originate**, **next-hop-self**, **password remove-private-as**, **send-community**, **distribute-list out**, **filter-list out**, **prefix-list out**, **route-map out**, **unspress-map**, **route-reflector-client.**

The **neighbor update-source** command can be used to select any valid interface for establishing a TCP connection. This command is designed mostly to provide available Loopback interfaces, which increase the stability of the connection to the IBGP Speaker.

By default, direct physical connection with BGP peers is required for establishing an EBGP connection. To establish the EBGP peers among non-direct-connected external BGP Speakers, the **neighbor ebgp-multihop** command can be used.

⚠ Caution

To avoid route loop and oscillation, the EBGP peers who need multiple hops for BGP connection must have non-default routes to each other.

For the sake of security, you can set the authentication for the BGP peers (group) which will establish the connection based on the MD5 algorithm. The authentication password for the BGP peer should be identical. The process of enabling the MD5 authentication on a BGP peer is shown as follows:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **password** *string*	Enables the TCP MD5 authentication and set the password when the BGP connection with the BGP peer is established.

Use the **no neighbor** {*address* | *peer-group-name*} **password** command to disable the BGP peer (group) from MD5 authentication.

Use the **neighbor shutdown** command to disable the valid connection established with the BGP peer (group), and delete all routing information related to the BGP peer (group).

⚠ Caution

To break the connection established with the specified BGP peer (group) and reserve the configuration information set for this specified BGP peer (group), use the **neighbor shutdown** command. If such configuration information is no longer required, use the **no neighbor** [**peer-group**] command.

7.6 Configuring the Management Policy

Whenever the routing policy (including the **neighbor distribute-list**, **neighbor route-map**, **neighbor prefix-list and neighbor filter-list)** changes, you need implement the new routing policy. The traditional way is to break and re-establish the BGP session.

This product supports implementing a new routing policy without ending the BGP session by using soft reset for BGP effectively.

To facilitate the description of BGP soft reset, the following section will refer to the routing policy that affects the input routing information as the input routing policy (such as the **In-route-map** and **In-dist-list**) and the policy that affects the output routing information as the output routing policy (such as the **Out-route-map** and **Out-dist-list**).

If the output routing policy changes, execute the following commands in BGP configuration mode:

Command	Function
Qtech# **clear ip bgp** {* | peer *address* | **peer-group** *peer-group-name* | **external**} **soft out**	Soft-resets the BGP session and executes the routing policy without resetting the BGP session.

A change in input routing policy changes complicates operations compared with the output routing policy, because the output routing policy is based on the routing table of this BGP Speaker. The implementation of the input routing policy is based on the routing information received from a BGP peer. To reduce memory consumption, the local BGP Speaker will not retain the original routing information received from BGP peers.

To modify the input routing policy if necessary, save the original routing information for each specified BGP peer in this BGP Speaker by using the **neighbor soft-reconfiguration inbound** command. The aim is to provide the original foundation of routing information to modify the input routing policy.

At present, the standard implementation method is referred to as the “Route Refresh Performance”, which can support modifying the routing policy without storing the original routing information. This product supports the feature.

If the input routing policy changes, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **soft-reconfiguration inbound**	Restarts the BGP session and reserve the unchanged routing information from the BGP peer (group). This command may consume more memory. If both parties support route refreshing performance, this command becomes unnecessary.
Qtech# clear **ip bgp** {* | *peer-address* | **peer-group** *peer-group-name* | **external**} **soft in**	Soft-resets the BGP session and executes the routing policy without resetting the BGP session.

You can determine whether the BGP peer supports route refreshing performance by the **show ip bgp neighbors** command. If so, you need to execute the **neighbor soft-reconfiguration inbound** command when the input routing policy changes.

7.7 Configuring Synchronization between BGP and IGP

The routing information can be transmitted to another AS through the local AS only when it passes through this AS and reaches another AS. The routing information will be advertised to all the routers in the local AS. Otherwise, if some routers running the IGP protocol within this AS have not learned this routing information, data packets may be discarded, because these routers do not know this route when these packets traverses this AS, which may cause a route black hole.

The BGP-IGP synchronization is designed to ensure all routers within this AS can learn the outgoing routing information. Simply, the BGP Speakers redistribute all of the routes learned by the BGP protocol to the IGP protocol to ensure that the routers within the AS learn such routing information.

The BGP-IGP synchronization mechanism can be cancelled in two situations:

10) There is no routing information passing through the local AS (In general, this AS is an end AS).

11) All routers within this AS run the BGP protocol and a full connection is established among all BGP Speakers (An adjacent relationship is established between any two BGP Speakers).

Caution

By default, synchronization is disabled. Enable synchronization when not all the routers are running BGP when traversing an AS.

To enable synchronization of BGP speakers, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **synchronization**	Enables synchronization of BGP and IGP.

Execute the **no synchronization** command to disable the synchronization mechanism.

7.8 Configuring Interaction between BGP and IGP

To inject the routing information generated by the IGP protocol into the BGP protocol, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **redistribute** [**connected** | **rip** | **static**] [**route-map** *map-tag*] [**metric** *metric-value*]	(Optional) Redistributes static route, direct route and the routing information generated by RIP.
Qtech(config-router)# **redistribute ospf** *process-id* [**route-map** *map-tag*] [metric *metric-value*] [**match internal external** [**1** | **2**] **nssa-external** [**1** |**2**]]	(Optional) Redistributes the routing information generated by OSPF.
Qtech(config-router)# **redistribute isis** [*isis-tag*] [**route-map** *map-tag*] [**metric** *metric-value*] [**level-1**| **level-1-2**| **level-2**]	(Optional) Redistributes the routing information generated by ISIS.

By default, distribution of a default route is disabled. To enable this function, execute the following command:

Command	Function
Qtech(config-router)# **default-information originate**	Redistributes the default route.

7.9 Configuring BGP Timer

The BGP uses the Kepalive timer to maintain an effective connection with the peers, and takes the Hldtime timer to determine whether the peers are valid. By default, the value of the Kepalive timer is 60s, and the value of the Holdtime timer is 180s. When a BGP session is established between BGP Speakers, both parties will negotiate with the Holdtime timer and the one with a smaller value will be selected. The selection of the Keepalive timer is based on the smaller one between 1/3 of the negotiated Holdtime timer and the configured Keepalive timer.

To adjust the value of the BGP timer based on all peers, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **timers bgp** *keepalive holdtime*	Adjusts the keepalive and holdtime values of BGP based on all peers. The range of the *keepalive* is 0 to 65535 seconds, and 60 seconds by default. The range of the *holdtime* is 0 to 65535 seconds, 180 seconds by default.

Certainly, you can adjust the value of the BGP timer based on the specified peers, and execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **times** *keepalive holdtime*	Configures the Keepalive and Holdtime value to establish a session with the specified BGP peer (group). The range of the keepalive is 0 to 65535 seconds, 60 seconds by default. The range of the holdtime is 0 to 65535 seconds, 180 seconds by default.

Use the **no** form of the right command to clear the value of configured timer.

7.10 Configuring BGP Path Attributes

7.10.1 AS_PATH Attribute

The BGP protocol controls the distribution of routing information in the following ways:

- IP address by using the **neighbor distribute-list** and **neighbor prefix-list** commands
- AS_PATH Attribute (refer to this section)
- COMMUNITY Attribute (refer to the COMMUNITY Attribute configuration)

You can use the AS path-based access control list to control the distribution of the routing information, where the AS path-based ACL will use Regular Expression to resolve the AS path.

To configure the AS path-based distribution of routing information, execute the following commands in privileged EXEC mode:

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **ip as-path access-list** *path-list-name* {**permit** | **deny**} *as-regular-expression*	(Optional) Defines an AS path list.
Qtech(config)# **ip routing**	Enables the routing function (if disabled)
Qtech(config)# **router bgp** *as-number*	Enables the BGP and configures this AS number to enter BGP configuration mode.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **filter-list** *path-list-name* {**in** | **out**}	(Optional) Implements the routing policy according to the AS path list when the routing information is received from and sent to the specified BGP peer (group).
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Implements the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group). In route-map configuration mode, you can use the match as-path to operate the AS path attribute based on the AS path list, or take the set as-path to operate the AS attribute value.

The BGP protocol will not consider the length of the AS path when selecting the optimal path as specified in RFC1771. In general, the shorter the AS path, the higher its priority. Hence, we take the length of the AS path as the optimal path. You can determine whether to consider the length of the AS path when selecting the optimal path according to actual condition.

If you wish to ignore the length of the AS path when selecting the optimal path, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp bestpath as-path ignore**	Compares the length of the AS path when selecting the optimal path.

⚠ Caution

Within the AS, all BGP Speakers consider the length of the AS path as consistent when selecting the optimal path. Otherwise, the optimal path information selected by different BGP Speakers will be different.

7.10.2 NEXT_HOP Attribute

To set the next hop as the local BGP Speaker for sending the routing information to the specified BGP peer, you can use the **neighbor next-hop-self** command, which is mainly used in non-mesh networks, such as frame relay and X.25. Execute the follwing commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **next-hop-self**	Sets the next hop as the local BGP speaker for distributing the routing information to the specified BGP peer (group).

You can also modify the next hop of the specified path by using the **set next-hop** command of Route-map.

⚠ Caution

This command is not recommended for a fully meshed network such as Ethernet, because it may cause additional hops and incur unnecessary overhead.

7.10.3 MULTI_EXIT_DISC Attribute Configuration

The BGP takes the MED value as the foundation for priority comparision of the paths learned from the EBGP Peers. The smaller the MED value, the higher the path priority.

By default, the protocol only compares it with the MED value for the path of the peers from the same AS when the optimal path is selected. If you hope to compare it with the MED value for the path of the peers from different ASs, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp always-compare-med**	Compares with the MED value for the path of different ASs.

By default, it will not compare with the MED value for the path of the peers for other ASs within the AS when the optimal path is selected. If you hope to compare with the MED value for the path of the peers from different AS confederations, execute the following commands in the BGP configuration mode.

Command	Function
Qtech(config-router)# **bgp bestpath med confed**	Compares with the MED value for the path of the peers from other ASs within the confederation.

By default, if a path with an undefined MED attribute is received, the MED value of this path will be taken as 0. The smaller the MED value, the higher the path priority. The MED value of this path reaches the highest priority. If you want the MED attribute for the path with undefined MED attribute to present the lowest priority, execute the following command in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp bestpath med missing-as-worst**	Sets the priority of the path whose MED attribute is not set as the lowest.

By default, they are compared with each other in the sequence the paths are received when the optimal path is selected. If you want to first compare with the path of the peers from the same AS, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp deterministic-med**	Compares first with the path of the peers from the same AS. By default, they will be compared with by the receiving sequence. The later received path will be compared with first.

7.10.4 LOCAL_PREF Attribute Configuration

The BGP takes the LOCAL_PREF as the foundation for priority comparision of the path learned from the IBGP peers. The larger the LOCAL_PREF value, the higher the path priority.

The BGP Speakers will add the local preference when they send the received external routes to the IBGP peers. To modify the local preference, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp default local-preference** *value*	Changes the default local preference. The range of the value is 0 to 4294967295, 100 by default.

You can also modify the local preference of the specified path by using the **set local-preference** command of Route-map.

7.10.5 COMMUNITY Attribute Configuration

COMMUNITY Attribute provides another way to control the distribution of the routing information.

The community is a set of destinations. The purpose is to implement the community-based routing policy so as to simplify the configuration for the distribution of the routing information in the BGP Speakers.

Each destination may have more than one community, and the manager of the AS can define the community destination.

By default, all destinations belong to the Internet community carried in the community attribute of the path.

At present, totally four common community attributes are predefined:

- **Internet**: Indicates the Internet community, and all paths are in this community.
- **no-export**: Indicates this path will not be exported to BGP peers.
- **no-advertise**: Indicates this path will not be advertised to BGP peers.
- **local-as**: Indicates this path will be advertised only in the local AS or the AS confederation if it is configured.

You can control the receiving, priority and distribution of the routing information by using the community attribute.

The BGP supports up to 32 COMMUNITY attributes for every route. When configuring the **route-map** command, you can set up to 32 COMMUNITY attributes for the parameters **match** and **set COMMUNITY.**

The BGP Speakers can set, add or modify the community attribute value when they learn, issue or redistribute a route. The aggregated path includes the community attribute of all aggregated paths when route aggregation is carried out.

To configure the community attribute-based distribution of routing information, execute the following commands in privileged EXEC mode:

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **ip community-list standard** *community-list-name* {**permit** | **deny**} *community-number*	(Optional) Creates the community list. The *community-list-name* is the name of the community list. The community-number is the concrete value of the community list in the range 1 to 4,294,967,295, or the well-known community attribute such as Internet, local-AS, no-advertise and no-export.
Qtech(config)# **ip routing**	Enables the routing function (if disabled).
Qtech(config)# **router bgp** *as-number*	Enables the BGP and configure this AS number to enter into BGP configuration mode.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **send-community**	(Optional) Configures the router to send the community attribute to the specified BGP peer (group).

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group). In the route-map configuration mode, you can use the match community-list [exact] and set community-list delete to operate the community attribute by the community list, or use the set community command to operate the community attribute value directly.

7.11 Other Related Configuration

By default, if two paths with identical path attributes are received from different EBGP peers during the selection of the optimal path, we will select the optimal path based on the path receiving sequence. You can select the path with a smaller router ID as the optimal path by using the following commands.

Command	Function
Qtech(config-router)# **bgp bestpath compare-routerid**	Allows the BGP to compare with the router ID when the optimal path is selected.

7.12 Selecting the Optimal Path for BGP

Optimal route selection forms an important part of the BGP protocol. The following section describes the selection process of the BGP route protocol in detail:

- An invalid routing table entry is not allowed in the selection of optimal routes.

⚠ Caution Invalid entries include those unreachable for the next hop and those in oscillation.

- Select the route with the high LOCAL_PREF attribute value.
- Select the route generated by the local BGP speaker.

The route generated by the local BGP speaker includes the one generated by the **network, redistribute, aggregate** command.

- Select the route with the shortest AS length.
- Select the route with the lowest ORIGIN attribute.
- Select the route with the smallest MED value.
- The EBGP path has a higher priority than the IBGP path and the AS confederation, and the priority is identical for the IBGP path and the AS confederation.
- Select the route with the smallest IGP metric to reach the next hop.
- Select the route received earlier from the EBGP routes.
- Select the route which advertises that the router ID of the BGP speaker is small.
- Select the route with the greater cluster length.
- Select the route: the value of neighbor address for which is high.

⚠ Caution Discussed above is the process of select the optimum route under the default configuration. You can change the selection process of the route by the CLI command. For instance, you can use the **bgp bestpath as-path ignore** command to make step 4 part of the process of invalidating the optimal route. Use the bgp bestpath compare-routerid command to invalidate Step 9 of the selection.

7.13 Configuring BGP Route Aggregation

Since the BGP-4 supports CIDR, aggregated entries can be created to downsize the BGP routing table. Certainly, BGP aggregated entries can be added to the BGP routing table only when there is a valid path within the aggregation scope.

To configure the BGP route aggregation, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **aggregate-address** *address mask*	(Optional) Configures the aggregated address.
Qtech(config-router)# **aggregate-address** *address mask* **as-set**	(Optional) Configures the aggregated address, and remain the AS path information of the path within the scope of the aggregated address.
Qtech(config-router)# **aggregate-address** *address mask* **summary-only**	(Optional) Configures the aggregated address and only advertise the aggregated path.
Qtech(config-router)# **aggregate-address** *address mask* **as-set summary-only**	(Optional) Configures the aggregated address, and remain the AS path information of the path within the scope of the aggregated address. At the same time, only the aggregated path is advertised.

Use the **no** form of the above commands to disable the configuration.

⚠ Caution

By default, the BGP will advertise all routing information both before and after aggregation. If you want to advertise only the aggregated path information, use the **aggregate-address summary-only** command.

7.14 Configuring Route Reflector for BGP

To speed up the convergence of routing information, all BGP Speakers within one AS will usually establish the full connection (The adjacent relationship is established between any two BGP Speakers). Too many BGP Speakers within the AS may increase the resource overhead of the BGP Speakers, raise the configuration workload and complexity of network administrators, and reduce the network scalability.

Therefore, route reflector and AS confederation are preferrably used to reduce the connections of the IBGP peers within an AS.

The route reflector provides a way to reduce the connections of the IBGP peer within the AS. One BGP Speaker is set as the route reflector, which divides the IBGP peer within this AS into two types, such as client and non-client.

The rule to implement the route reflector within the AS is shown as follows:

- Configure the route reflector and specify its client, so the route reflector and other clients form a cluster. The route reflector establishes the connection with clients.
- The clients of the route reflector within one cluster should not establish the connection with other BGP Speakers of other clusters.
- Within an AS, a full connection is established among the IBGP peer of non-clients. The IBGP peer of non-clients involves the following scenarios: among several route reflectors within one cluster, among the route reflector within the cluster and the BGP Speakers not involved in the route reflector function out of the cluster (In general, the BGP Speakers don’t support the route reflector function), among the route reflector within the cluster and the route reflector of other cluster.

The following rules applies when the route reflector receives one route:

- The route update received from the EBGP Speaker will be sent to all clients and non-clients.
- The route update received from the clients will be sent to other clients and all non-clients.
- The route update received from the IBGP non-clients will be sent to all its clients.

To configure the BGP route reflector, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **route-reflector-client**	Configures this product as the route reflector and specifies its clients.

In general, one group is only configured with one reflector. In this case, the Router ID of the route reflector can be used to identify this cluster. To increase the redundancy, you can set more than one route reflector within this cluster. You must configure the cluster ID, so that one route reflector can identify the route update from other route reflectors of this cluster.

Caution To set several route reflectors for one cluster, you need to configure a cluster ID for this cluster.

To configure the cluster ID of the BGP, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp cluster-id** *cluster-id*	Configures the cluster ID of the route reflector.

In general, it is not necessary to establish a connection between the clients of the route reflector within the cluster, as the route reflector will reflect the routes among clients. However, this function can be disabled if a full connection is established among all clients.

To disable the function of reflecting the client routes, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **no bgp client-to-client reflection**	Disables route reflection on clients.

7.15 Configuring Route Flap Dampening for BGP

Route flap means that a route changes between the valid status and the invalid status. The route flap usually causes instable routes to be transmitted on the Internet, thus resulting an unstable network. BGP route flap dampening provides a way to reduce route flap by monitoring the routing information of EBGP peers.

The route flap dampening of BGP uses the following terminologies:

- Route Flap: A route changes between the valid status and the invalid status.
- Penalty: The route flap dampening-enabled BGP Speakers will add a penalty for the route each time when a route flaps. The penalty will be accumulated to exceed the suppress limit.
- Suppress Limit: When the penalty of a route exceeds this value, the route will be suppressed.
- Half-life-time: The time elapsed when the penalty is reduced to half of its value.
- Reuse Limit: When the penalty of the route is lower than this value, route suppression is released.
- Max-suppress-time: The maximum amount of time the route can be suppressed.

Overview of route flap dampening: The BGP Speakers will add a penalty for the route each time when a route flaps. The penalty is accumulated. Once the penalty value reaches the suppress limit, the route will be suppressed. When the half-life-time is reached, the penalty value is reduced to half of its value. Once the penalty value is reduced to the reuse limit, the route will be activated again. A route can be suppressed for the maximum suppress time.

To configure the route flap dampening of the BGP, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp dampening**	Enables the route flap dampening of the BGP protocol.
Qtech(config-router)# **bgp dampening** *half-life-time reuse suppress max-suppress-time*	(Optional) Configures the parameters of the route flap dampening. half-life-time: in the range of 1 to 45minutes, 15minutes by default. reuse: in the range of 1 to 20000, 750 by default. suppress: in the range of 1 to 20000, 2000 by default. max-suppress-time: in the range of 1 to 255 minutes, 4*half-life-time by default.

To monitor the route flap dampening information if necessary, execute the following commands in privileged EXEC mode:

Command	Function
Qtech# **show ip bgp dampening flap-statistics**	(Optional) Shows the flap statistics information of all routers.
Qtech# **show ip bgp dampening dampened-paths**	(Optional) Shows the dampened statistics.

To clear the route flap dampening information or the dampened routes, execute the following commands in BGP configuration mode:

Command	Function
Qtech# **clear ip bgp flap-statistics**	(Optional) Clears flap statistics about all un-dampened route.

Command	Function
Qtech# **clear ip bgp flap-statistics** address mask	(Optional) Clears flap statistics about the specified route (excluding the dampened routes).
Qtech# **clear ip bgp dampening** [*address mask*]	(Optional) Clears flap statistics about all routes, and releases the suppressed routes.

7.16 Configuring AS Confederation for BGP

Confederation provides a way to reduce the connections of the IBGP peer within the AS.

One AS is divided into multiple sub ASs that can form a confederation by setting a unified confederation ID (namely, confederation AS number). An external confederation is still considered an AS and only the AS number of the confederation is visible. Within the confederation, the full IBGP peer connection is still established among the BGP Speakers, and the EBGP connection is established among the BGP Speakers within the sub AS. Although the EBGP connection is established among BGP Speakers within the sub ASs, the path attribute information of NEXT_HOP, MED and LOCAL_PREF remains intact when the information is exchanged.

To implement the AS confederation, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp confederation identifier** *as-number*	Configures the AS confederation number. The range of *as-number* is 1 to 4294967295.
Qtech(config-router)# **bgp confederation peers** *as-numbe* [as-number..]	Configures other sub AS numbers within the AS confederation. The range of *as-number* is 1 to 4294967295.

Use the **no** form of the above commands to disable the configuration.

7.17 Configuring BGP Management Distance

The management distance indicates the reliability of the routing information resource, within the range of 1 to 255. The larger the value of the management distance, the lower the reliability is.

The BGP sets different management distances for different information sources that have been learned, such as External-distance, Internal-distance and Local-distance.

- **External-distance:** The management distance of the route learned from the EBGP peers.
- **Internal-distance:** The management distance of the route learned from the IBGP peers.
- **Local-distance:** The management distance of the route learned from the peers. However, the optimal one can be learned from the IGP. In general, these routes are indicated by the **Network Backdoor** command.

To modify the management distance of the BGP protocol, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **distance bgp** *external-distance internal-distance local-distance*	Configures the management distance. The range of the distance is 1 to 255. For the default configuration: *external-distance 20* *internal-distance 200* *local-distance 200*

Use the **no** form to restore the default management distance of the BGP protocol.

⚠ Caution

It is not recommended that you change the management distance of the BGP route. If the change is necessary, please make sure:

■ The External-distance is lower than the management distance of other IGP route protocol (OSPF and RIP).

■ The Internal-distance and Local-distance is higher than the management distance of other IGP route protocol.

7.18 Configuring BGP Route Update Mechanism

The BGP route update mechanism comprises two parts: timing scanning update and event trigger update. The former means that the timer is used in the BGP to start the scanning mechanism periodically to update the routing table. The latter means that when BGP configuration or the next hop of BGP route changes, the BGP protocol starts the scanning mechanism to update the routing table.

To configure the BGP route update mechanism, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp scan-rib disable**	Enables the event trigger mechanism. By default, the timing scanning update mechanism is used.
Qtech(config-router)# **bgp scan-time** *scan-time*	(Optional) Sets the scanning interval. *scan-time*: In the range of 5 to 60 seconds, 60 seconds by default

You can also configure this command in IPv4/IPv6/VPNv4/IPv4 vrf address family mode.

Use the **no** form to remove the configuration.

Caution

When you run the **bgp scan-rib disable** command to enable the event trigger mechanism, the synchronization should be disabled and the BGP next hop trigger mechanism should be enabled. When synchronization is enabled or the BGP next hop trigger mechanism is disabled, the BGP updates the routing table in timing scanning mode.

7.19 Configuring BGP Nexthop Trigger Update Mechanism

The BGP next hop trigger update mechansim improves the converge of BGP routes. It monitors the next hop of BGP routes to speed up converge in stable network topology.

By default, the BGP next hop trigger update mechansim is enabled. After establishing connections with neighbors, the BGP will automatically monitor the next hops of the routes learned from neighbors. When the next hop changes, the BGP will receive a notification of updating the routing tble. This can reduce the time to check the change of next hop for better converge of BGP routes.

If the function is disabled, scan-timer will periodically scan updates to the next hope of BGP.

To configure the BGP next hop trigger update mechanism, execute the following commands in BGP configuration mode:

Command	Function
Qtech(config-router)# **bgp nexthop trigger enable**	Enables the function of triggering the next BGP route. This function is enabled by default.
Qtech(config-router)# **bgp nexthop trigger delay** ***delay****-time*	(Optional) Sets the delay of the BGP next hop trigger update. *delay-time*: In the range of 0 to 100 seconds, 5 seconds by default

You can also configure this command in IPv4/IPv6/VPNv4/IPv4 vrf address family mode.

Use the **bgp nexthop trigger enable** command to restore the setting to the default value.

The **bgp nexthop trigger enable** command and the bgp scan-time command control the same timer. When the timing scanning mechanism is enabled (bgp scan is enabled by default. The **bgp scan-rib disable** command is used to disable **bgp scan**）， the time of larger than 60 seconds set by the **bgp nexthop trigger enable** command does not take effect because the timing scanning mechanism is always activated before the delay time.

Caution

In an unstable network (the next hop changes frequently), especially when there are a large number of routes, this function carries out unnecessary route calculation and consumes more CPU resources. In this case, it is recommended that you disable the BGP next hop trigger update mechanism.

7.20 Configuring BGP GR

GR (Graceful Restart) can ensure continuous data forwarding during the resetting of the BGP protocol. Currently, Qtech supports GR during active and standby switching on its high-end devices to ensure service continuity.

7.20.1 Working Mechanism of GR

12) Standard

RFC4724: Graceful Restart Mechanism for BGP, which is represented by BGP GR later.

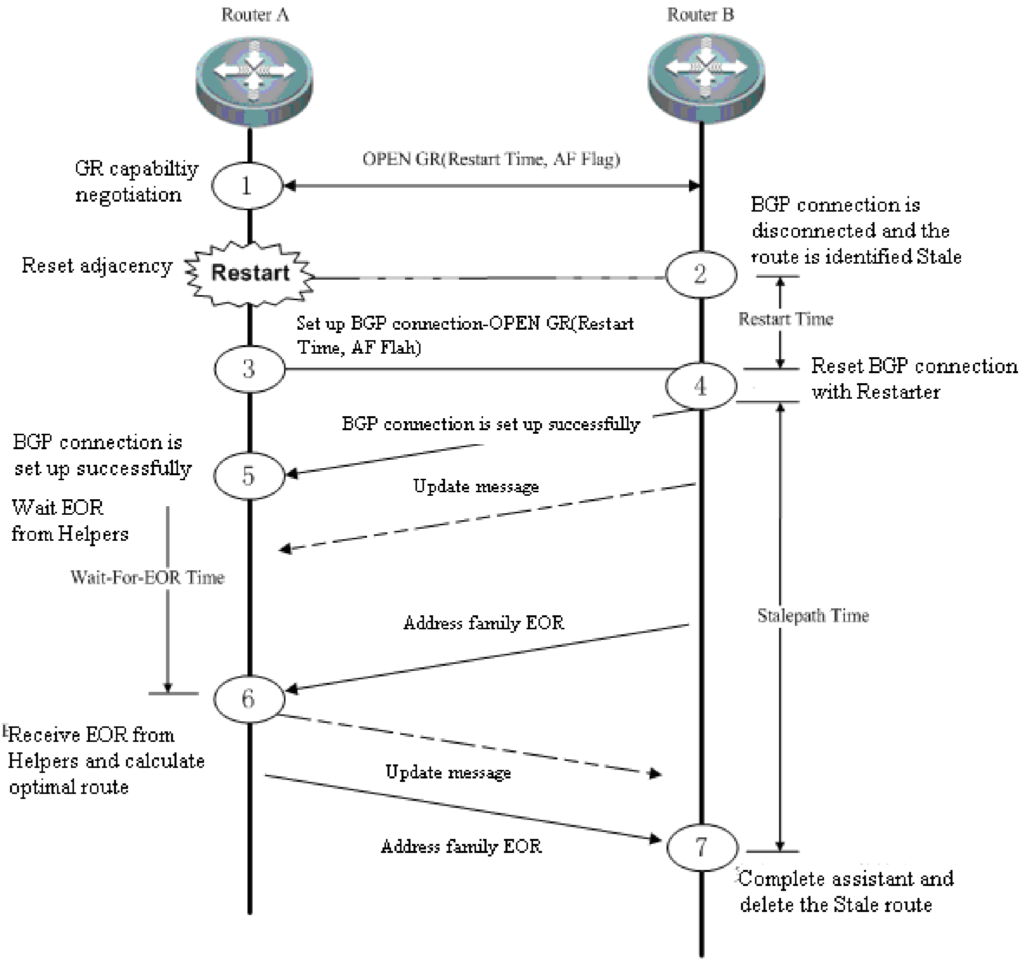
13) Working mechanism

RFC4724 is a standard GR protocol that IETF especially defines for the BGP protocol. This document outlines the principles of BGP GR, including:

- Graceful Restart Capability is added to the OPEN message of the BGP protocol, indicating that the BGP supports GR. The GR capability is negotiated by neighbors during the initiation of BGP connection.
- GR Restarter and GR Helper. GR Restarter means that the router restarts the BGP protocol, which can ensure continuous route forwarding when the route control panel fails. GR Helper is the BGP neighbor of the GR Restarter that assists the GR Restarter in BGP GR for continous forwarding across the network.
- In the update message, EOR (End-of-RIB) is added to indicate that the route message update is complete.

The following ffigure illustrates the process of BGR GR.

Figure 1 Process of graceful restart for BGP


Router A
Router B
GR capabiltiy negotiation
1
OPEN GR(Restart Time, AF Flag)
BGP connection is disconnected and the route is identified Stale
Reset adjacency
Restart
2
Restart Time
Set up BGP connection-OPEN GR(Restart Time, AF Flah)
3
4
Reset BGP connection with Restarter
BGP connection is set up successfully
BGP connection is set up successfully
5
Update message
Wait EOR from Helpers
Wait-For-EOR Time
Stalepath Time
Address family EOR
6
Receive EOR from Helpers and calculate optimal route
Update message
Address family EOR
7
Complete assistant and delete the Stale route

Initially, the BGP protocol establishes a adjacency and negotiates respective GR capability with the GR Capability field of the OPEN message. At a point, the device reboots and the BGP session is disconnected. The neighbor detects disconnection. With GR supported, the BGP neighbor keeps the route of the GR Restarter valid but identifies it in Stale (aged, not updated) state. The GR Restarter reboots and re-establishes connection with the GR Helper and waits the route update message and EOR label from the GR Helper. After receiving an EOR label from all neighbors, the BGP Restarter calculates routes and update the routing table, and begins to send update routes to the GR Restarter. Upon the receipt of these routes, the GR Helper removes the Stale tag from these routes and then deletes the routes (these routes have not been updated) tagged with Stale after receiving the EOR label from the BGP Restarter, calculates routes and updates the routing table.

Some key timers are defined to assist the implemetation of BGP GR:

- **Restart-Timer**: The GR Restarter notifies the GR Helper of restart time that the GR Helper needs to wait before reestablishing the BGP connection. You can modify this value by using the **bgp graceful-restart restart-time** command.
- **Wait-For-EOR Timer**: Time the GR Restarter needs to wait for the EOR label of all GR Helpers. After receiving the EOR label of all GR Helpers or the timer expires, the GR Restarter calculates optimal routes and updates the routing table. You can modify this value by using the **bgp update-delay** command.
- **StalePath Timer**: Time the GR Helper needs to wait before receiving the EOR label from the GR Restarter after reestablishing the connection with the GR Restarter. During this period, the GT Helper keeps alive the route of the GR Restarter. It will delete the rotue tagged with Stale after receiving the EOR lable or the StalePath timer expires. You can modify this value by using the **bgp graceful-restart stalepath-time** command.

7.20.2 Implementation of BGP GR

Implementation of BGP GR is not an independent process. All BGP peers are necessary to enable BGP GR capability for normal operation. Failed GR may cause a temporary route black hole or loop and affect the network operation. Consequently, it is recommended that you ensure the GR capability is negotiatied successfully by using the **show ip bgp neighbors** command. To enable BGP GR, exeucte the **bgp graceful-restart** command in BGP route configuration mode.

7.20.3 Configuring BGP GR Capability

BGP GR capability is an extended capability of the BGP protocol, which is disabled by default. When enabling GR, the BGP reestablishes the connection with its neighbor and negotiates GR capability. The GR is enabled only when both sides support GR capability.

To enable the GR capability, execute the following commands:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *500*	Enters BGP configuration mode.
Qtech(config-router)# **bgp graceful-restart**	Enables GR.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Shows the configuration.
Qtech # **write**	(Optional) Saves the configuration.

☑ All the BGP-enabled products support this command.

⚠ Caution

The **bgp graceful-restart** command does not take effect for established BGP connections. Namely, the BGP connection will not negotiate GR capability immediately when it is in Established status. In this case, you need to forcibly restart the peer to renegotiate the GR capability, for instance, **clear ip bgp 192.168.195.64**. This is to prevent the restart of neighboring relations for capability renegotiation when GR is enabled or disabled, as renegotiation may cause network oscillation. Therefore, you can decide whether to restart neighbors.

Supporting BGP GR capability does not mean a device can serve as the GR Restarter for graceful restart, which also depends on the hardware. The GR Restarter device of Qtech Networks needs to support dual-engine redundant hot backup.

7.20.4 Configuring BGP GR Timer

After the GR capability is enabled, the BGP automatically configures relevant timers with default values. By default, the Restart Timer is 120s, the Wait-For-EOR Timer is 120s and the StalePath Timer is 360s.

To configure these timers, execute the following commands:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *500*	Enters BGP configuration mode.
Qtech(config-router)# **bgp graceful-restart**	Enables GR capability.
Qtech(config-router)# **bgp graceful-restart restart-time** *150*	Sets the Restart Timer to 150s.
Qtech(config-router)# **bgp update-delay** *150*	Sets the Wait-For-EOR Timer to 150s.
Qtech(config-router)# **bgp graceful-restart stalepath-time** *400*	Sets the StalePath Timer to 400s.
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Shows the configuration.

Command	Function
Qtech # **write**	(Optional) Saves the configuration.

☑ All the BGP-enabled products support this command.

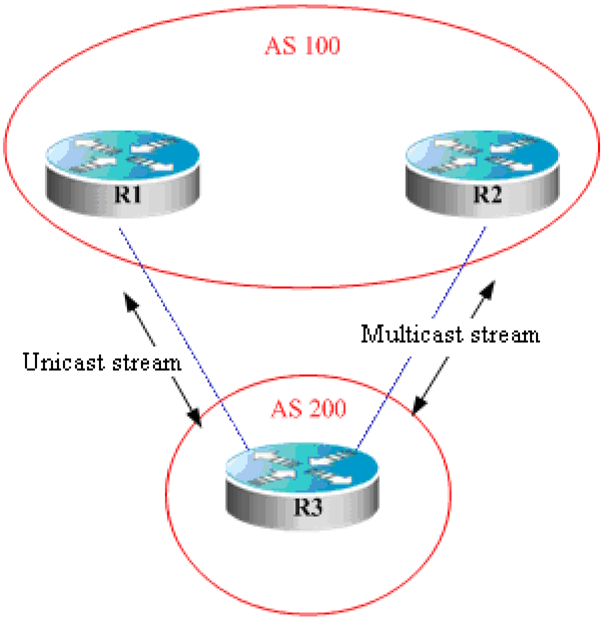
Caution

The restart time configured by using the **bgp graceful-restart restart-time** command should not exceed the Hold time of the BGP peer. Otherwise, the Hold Time will be used as the restart time and be notified to the peer for GR capability negotiation.

7.21 Configuring BGP Multicast

The BGP multicast route is used for multicast RFC check. In general, the multicast forwarding topology is similar to the unicast forwarding topology. You can design different multicast topologies by using BGP multicast, which is used for the multicast topology between ASs, as shown in the following figure.

Figure 2



There are two routers in AS100. In terms of design, unicast streams are sent to R1 and multicast streams to R2. In this case, MPBGP is required between R2 and R3.

Step 1: Enable BGP on R1, R2 and R3 and establish neighbors among them.

Take R3 as an example. Configures R1 and R2 as its BGP neighbors.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *200*	Enters BGP configuration mode with the AS number of 200.
Qtech(config-router)# **neighbor** *R2* **remote-as** *100*	Configures R2 as the BGP neighbor with the AS number of 100.
Qtech(config-router)# **neighbor** *R1* **remote-as** *100*	Configures R1 as the BGP neighbor with the AS number of 100.

Step 2: Since R3 does not need to transmit multicast routes with R1, disable the multicast address of R1.

Command	Function
Qtech(config-router)# **address-family ipv4 multicast**	Enters the IPv4 multicast address family configuration mode.
Qtech(config-router-af)# **no neighbor** *R1* **active**	Disable the multicast address of R1.

Step 3: Since R2 needs to transmit multicast rotues with R1, enable the multicast address of R2.

Command	Function
Qtech(config-router)# **address-family ipv4 multicast**	Enters IPv4 multicast address family configuration mode.
Qtech(config-router-af)# **neighbor** *R2* **active**	Enables the multicast address of R2.

Step 4: Import the routes that R3 needs to advertise to R2 in multicast address family mode.

Routes are imported by using **redistribute**, network advertising, and aggregate route publishing. They are configured in a multi-address family, for example:

Command	Function
Qtech(config-router)# **address-family ipv4 multicast**	Enters IPv4 multicast address family configuration mode.
Qtech(config-router-af)# **redistribute ospf** 1	Redistributes OSPF routes.

Caution

During the process of redistribution, the routes imported are unicast routes. For instance, the **redistribute ospf 1** command imports OSPF unicast routes. This is because that multicast routes depend on the egress of unicast routes for the establishment of a multicast spanning tree.

7.22 Configuring BGP Local AS

This function configures a local AS different from the real AS (rotuer BGP AS) for one peer, which is equivalent to virtualizing an AS. When the real AS changes, you still can establish a BGP connection without modifying the BGP configuration of the peer. Local AS applies to AS migration and converge of large networks without affecting the configurations of the devices in other interconencted ASs.

When establishing the BGP connection, the local device will advertise the local AS number to the peer in an OPEN message. The peer checks whether the AS number matches the local one and rejects the BGP connection if there is a difference. By default, the local AS of the BGP connection is the real BGP AS. With this function, the local device replaces the real AS with the configured one to establish a BGP connection.

By default, no peer is configured with Local AS. The Local AS of the peer is real AS over BGP. To configure a local AS for one peer, execute the following commands:

Command	**Function**
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *500*	Enters BGP configuration mode.
Qtech(config-router)# **neighbor** *192.168.195.64* **remote-as** *100*	Configures the peer.
Qtech(config-router)# **neighbor** *192.168.195.64* **local-as** *300*	Configures AS 300 as the local AS for the peer

The local AS function applies only to EBGP peers, instead of IBGP peers, confederation EBGP peers. Meanwhile, there are some limitations as described below:

- The local AS cannot be configured as the remote peer.
- Local AS cannot be configured for one member of the peer group.
- The local AS cannot be configured as the real BGP AS.
- The local AS cannot be configured as the AS number of the confederation if the device is a member of the confederation.

For details about the **neighbor** *peer-address* **local-as** *as-num* command, refer to the *Command Reference*.

7.23 Monitoring BGP

You can use the **Shows** commands to view the BGP route table, buffer and database. Execute the follwing commands in privileged EXEC mode:

Command	**Function**
Qtech# **show ip bgp**	Shows the information on all BGP routes.
Qtech# **show ip bgp** {*network* | *network-mask* } [longer-prefixes]	Shows the BGP routing information of the specified destination.
Qtech# **show ip bgp prefix-list** *prefix-list-name*	Shows the BGP routing information of the specified matching against the prefix list.
Qtech# show **ip bgp community** [**exact**] *community-number*	Shows the BGP routing information including the specified community.
Qtech# **show ip bgp community-list** *community-lister-number* [exact]	Shows the BGP routing information which matches against the specified community list.
Qtech# **show ip bgp filter-list** ***path-list***-*number*	Shows the BGP routing information which matches against the specified AS path list.
Qtech# **show ip bgp regexp** *as-regular-expression*	Shows the BGP routing information of the specified regular expression which matches against the AS path attribute.
Qtech# **show ip bgp dampening dampened-paths**	Shows the suppressed flap statistics information.
Qtech# **show ip bgp dampening flap-statistics**	Shows the flap statistics information of all routes with the flap record.

Command	Function
Qtech# **show ip bgp neighbors** [*address*] [**received-routes** | **routes** | **advertised-routes** | **received]**	Shows the information of the BGP peer.
Qtech# **show ip bgp summary**	Shows the configuration of the BGP router and the information about the peer.
Qtech# show **ip bgp peer-group** [*peer-group-name*]	Shows the configuration of the BGP peer group.

7.24 Protocol Independent Configuration

7.24.1 route-map Configuration

The BGP protocol follows the Route-map policy. For detailed configurations, refer to the Protocol Independent Configuration.

7.24.2 Regular Expression Configuration

The regular expression is a formula used to match the string based on a template. The regular expression is used to evaluate the text data and return a true or false value, that is to say, it determines whether the expression can describe this data correctly.

7.24.2.1 Description of Control Characters for Regular Expression

The BGP path attribute uses the regular expression. The following table describes the use of the special characters for the regular expression:

Characters	Signs	Special Functions
Period	.	Matched with any single character.
Asterisk	*	Matched with none or any sequence of the string.
Plus	+	Matched with one or any sequence of the string.
Interrogation Mark	?	Matched with none or one sign of the string.
Plus Sign	^	Matched with the starting of the string.
Dollar	$	Matched with the end of the string.
Underlining	_	Matched with the comma, bracket, the starting and end of the string and blank.
Square Brackets	[ ]	Matched with the single character within the specified scope.

7.24.2.2 Application Example of Regular Expression

Run the **show ip bgp** command on the device:

```
Qtech# show ip bgp
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Status Network                Next Hop      Metric  LocPrf  Path
------ ------------------    ---------------   --------  ---------  --------------------
*>  211.21.21.0/24    110.110.110.10  0      1000   200 300
*>  211.21.23.0/24    110.110.110.10  0      1000   200 300
*>  211.21.25.0/24    110.110.110.10  0      1000   300
*>  211.21.26.0/24    110.110.110.10  0      1000   300
*> 1.1.1.0/24        192.168.88.250  444          0   606
*> 179.98.0.0        192.168.88.250  444          0   606
*> 192.92.86.0       192.168.88.250  8883         0   606
*> 192.168.88.0      192.168.88.250  444          0   606
*> 200.200.200.0     192.168.88.250  777          0   606
```

Use the regular expression in the **show** command:

```
Qtech# show ip bgp regexp _300_
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Status Network            Next Hop            Metric  LocPrf  Path
------ ------------------  ---------------   --------  ---------  --------------------
*>  211.21.21.0/24  110.110.110.10  0      1000   200 300
*>  211.21.23.0/24  110.110.110.10  0      1000   200 300
*>  211.21.25.0/24  110.110.110.10  0      1000   300
*>  211.21.26.0/24  110.110.110.10  0      1000   300
```

7.25 BGP Load Protection Configuration

Too many BGP routes may overload a switch, especially a switch with a small memory size. BGP load protection can prevent unforeseen switch problems caused by switch resource usage.

7.25.1 Limiting BGP Routes

To limit BGP routes, configure the maximum number of routes in the BGP address-family mode. Configure the maximum number of routes learned from a BGP neighbor.

Use the following commands to configure the maximum number of routes learned from a BGP neighbor:

Command	Function
Qtech(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Qtech(config-router)#**neighbor** {*address* | *peer-group-name*} **remote**-as *as-num*	Configures the BGP neighbor.
Qtech(config-router)# **neighbor** {*address* | *peer-group-name*} **maximum-prefix** *maximum* [**threshold**] [**warning-only**]	Configures the maximum number of routes learned from the BGP neighbor.

Use the following commands to configure the maximum number of routes in the specified BGP address-family mode:

Command	Function
Qtech(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Qtech(config-router)#**address-family ipv4 unicast**	Enters the BGP ipv4 unicast address-family mode.
Or： Qtech(config-router)#**address-family ipv4** vrf *vrf-name*	Enters the BGP ipv4 VRF address-family mode.
Or： Qtech(config-router)#a**ddress-family vpnv4 unicast**	Enters the BGP VPNV4 address-family mode.
Qtech(config-router)# **maximum-prefix** *maximum*	Configures the maximum number of routes in the specified BGP address-family mode.

7.25.2 Configuring Overflow Memory-lack

BGP can be in the overflow state when the memory is insufficient. In OVERFLOW mode, BGP generates a default route to the NULL interface. A newly learned route will be discarded if it is not a default route in the current routing table. In general, the routes BGP learned in the overflow state are dropped, and the system memory stays in a steady state to protect the network from routing loops. In other words, BGP is safe and also preferred in OVERFLOW state.

Use the following commands to move BGP into the overflow state:

Command	Function
Qtech(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Qtech(config-router)# **overflow memory-lack**	Brings BGP into the overflow state when memory is running short.

Note

By default, BGP switches to OVERFLOW state automatically when the memory is running short. Use the **no overflow memory-lack** command for the BGP to exit the OVERFLOW state.

Caution

In OVERFLOW state, BGP supports the **clear bgp** { *addressfamily* | **all** } * command. Alternatively, you can disable and re-enable BGP to exit the OVERFLOS state. When the memory becomes sufficient, BGP exits the OVERFLOW state automatically.

7.26 BGP Configuration Examples

The following section lists BGP configurations.

7.26.1 Configuring BGP Neighbor

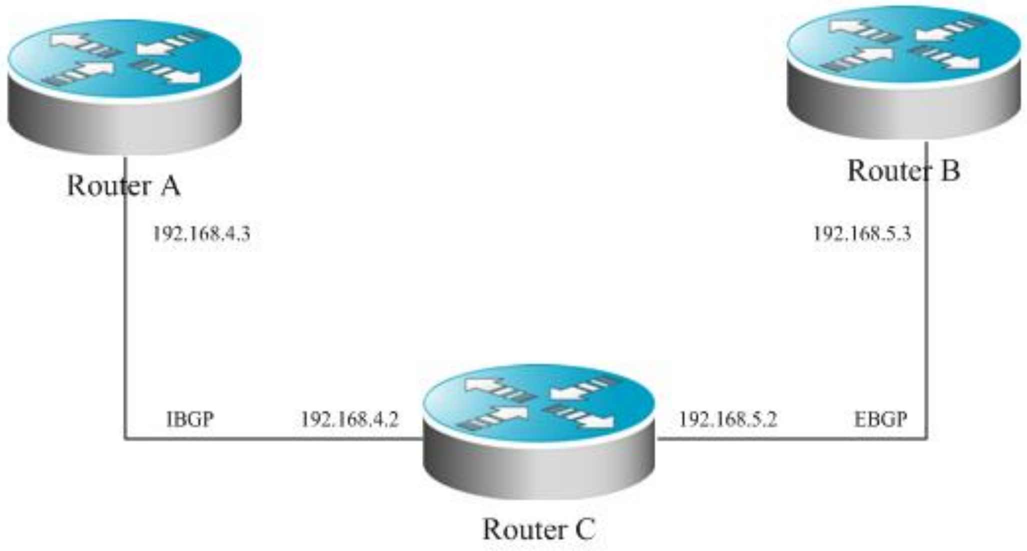
The following section shows how to configure a BGP neighbor. Use the **neighbor remote-as** command to configure the BGP neighbor. Configuration details are shown as follows:

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

Configure one IBGP peer 131.108.234.2 and two EBGP peers 131.108.200.1 and 150.136.64.19.

The following example shows how to configure the BGP neighbor. For the relationship among routers and the assignment of IP addresses, see the figure.

Figure 3



This example shows the BGP configuration of different routers:

Router A configuration:

```
!
router bgp 100
 neighbor 192.168.4.2 remote-as 100
```

Router B configuration:

```
!
router bgp 100
 neighbor 192.168.4.3 remote-as 100
 neighbor 192.168.5.3 remote-as 200
```

Router C configuration:

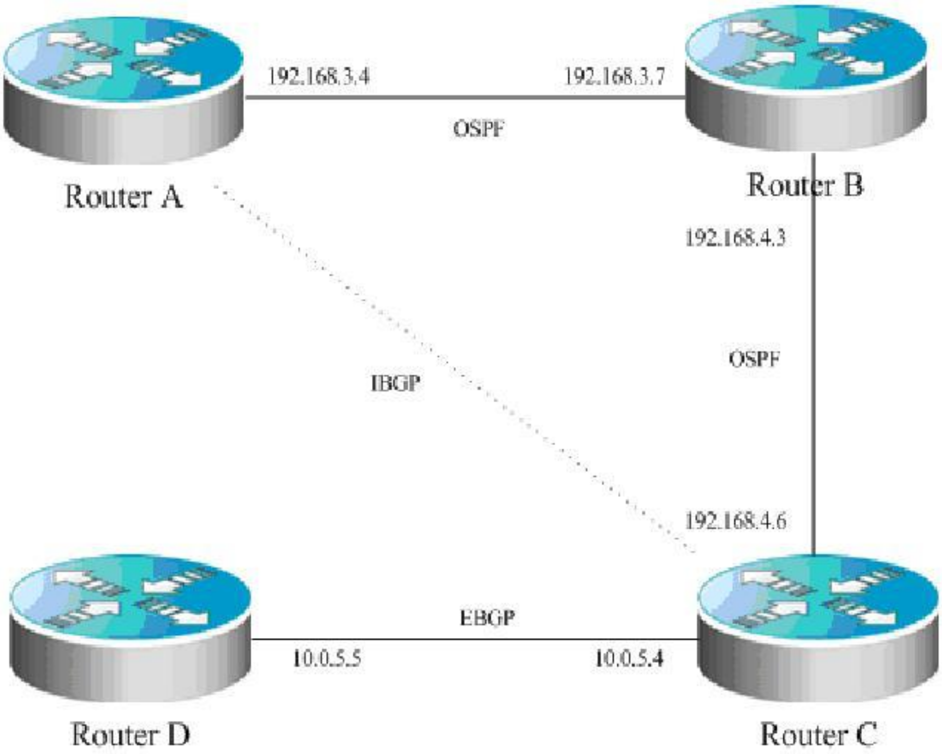
```
!
router bgp 200
 neighbor 192.168.5.2 remote-as 100
```

7.26.2 Configuring BGP Synchronization

Use the **synchronization** command to configure synchronization in BGP routing configuration mode, and use the **no synchronization** command to cancel the configured synchronization.

The following example shows the function of synchronization. The following figure illustrates the relationship between devices and the assignment of IP addresses:

Figure 4



In the figure, route p in router A is sent to router C based on the IBGP adjacency. If router C is configured with BGP synchronization, it is necessary for the router to wait for the IGP (this example uses the OSPF protocol) to receive the same routing information p, and send the route p to the EBGP neighbor, router D. If router C is configured asynchronously, it is not necessary for the BGP to wait for the IGP to receive the route p, and send the route p to the EBGP neighbor router D.

7.26.3 Configuring Neighbors to Use as-Path Filter

Configure the **as-path access-list** command for filtering first in configuration mode. Enter BGP route configuration mode after configuration, and use the **neighbor filter-list** command to apply the configured as-path access list among the BGP neighbors to filter AS paths.

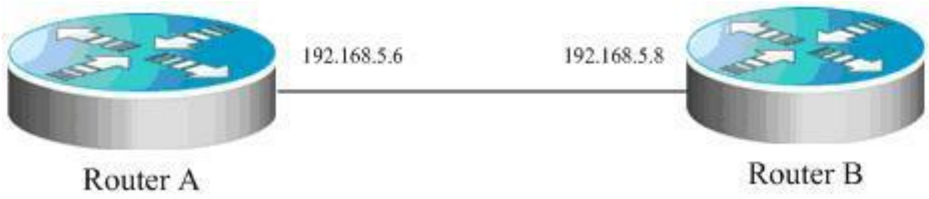
The configurations are detailed below:

```
router bgp 200
neighbor 193.1.12.10 remote-as 100
neighbor 193.1.12.10 filter-list 2 out
neighbor 193.1.12.10 filter-list 3 in
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```

This configuration indicates that only the routes permitted by the **as-path access-list** *2* can be advertised to the neighbor 193.1.12.10. The advertised routes from the neighbor 193.1.12.10 can be received only when they are permitted by the **as-path access-list** *3.*

The following figure provides a configuration eaxmple that shows the relationship and IP addresses of devices:

Figure 5



Do AS path-based filter on Router A.

The following example shows the configuraitons of different devices:

Router A configuration:

```
!
ip as-path access-list 4 deny ^300_
ip as-path access-list 4 permit .*
ip as-path access-list 5 deny ^450_65_
ip as-path access-list 5 permit .*
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.5.8 remote-as 200
 neighbor 192.168.5.8 filter-list 5 in
 neighbor 192.168.5.8 filter-list 4 out
```

Router B configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.5.6 remote-as 100
```

7.26.4 Configuring Route Aggregation

Use the **aggregate-address** command to configure an aggregated route in route configuration mode. When any route falls within the configured range, this aggregated route will become active.

The configuration is detailed as follows:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0
```

Configure one aggregate route:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0 as-set
```

The **as-path** segment of the aggregated route is an collection of **ASs**:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0 summary-only
```

The aggregated route is not advertised.

7.26.5 Configuring Confederation

When configuring a confederatin, you need to use the **bgp confederation identifier** command to configure the AS number for external connection, and use the **bgp confederation peers** command to configure confederation members.

The configuration is detailed as follows:

```
router bgp 6003
bgp confederation identifier 666
bgp confederation peers 6001 6002
neighbor 171.69.232.57 remote-as 6001
neighbor 171.69.232.55 remote-as 6002
neighbor 200.200.200.200 remote-as 701
```

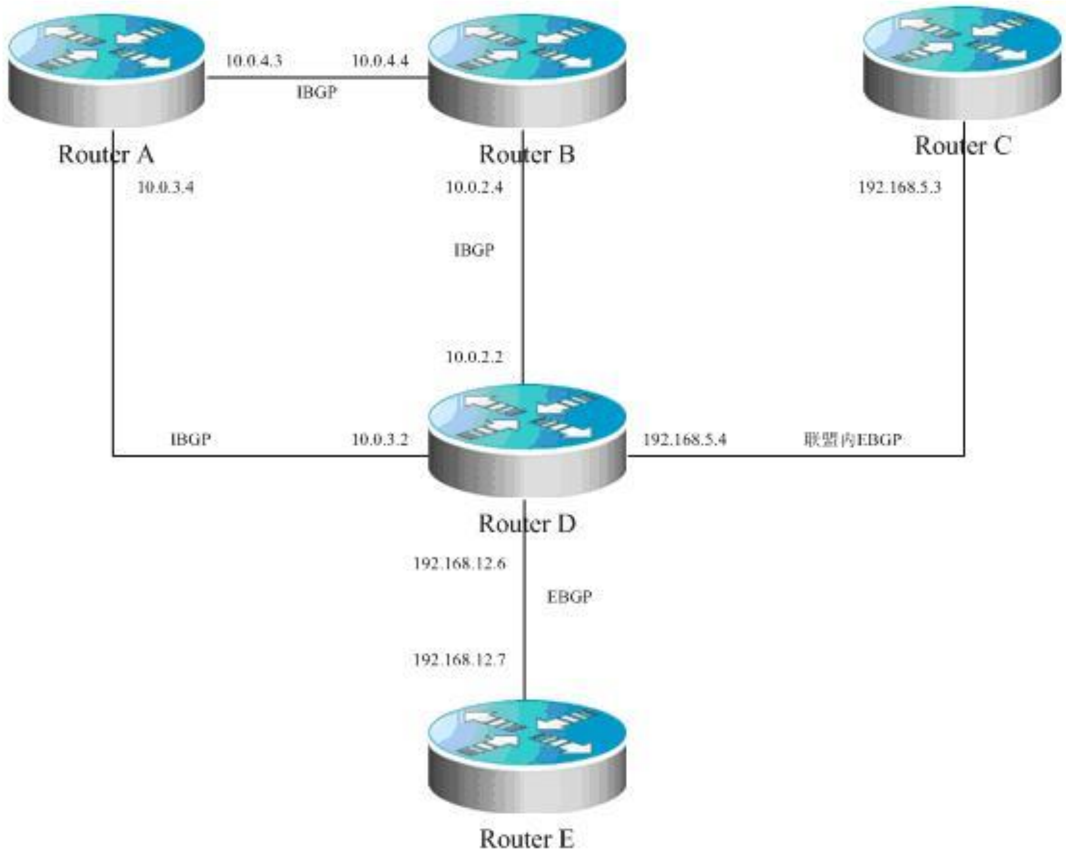
The configuration of peer 200.200.200.200 outside the confederation is shown as follows:

```
router bgp 701
neighbor 171.69.232.56 remote-as 666
neighbor 200,200,200,205 remote-as 701
```

For the configuration, the first device is in the confederation, while the second device is outside the confederation. Therefore, they are EBGP neighbors.

The following example shows their relastionship and IP addresses:

Figure 6



The following example shows the configurations of different devices:

Router A configuration:

```
!
router bgp 65530
 bgp confederation identifier 100
 bgp confederation peers 65531
 bgp log-neighbor-changes
 neighbor 10.0.3.2 remote-as 65530
 neighbor 10.0.4.4 remote-as 65530
```

Router B configuration:

```
!
router bgp 65530
 bgp confederation identifier 100
 bgp log-neighbor-changes
 neighbor 192.168.5.4 remote-as 65530
```

Router C configuration

```
!
router bgp 65531
 bgp confederation identifier 100
 bgp confederation peers 65530
 bgp log-neighbor-changes
 neighbor 10.0.3.2 remote-as 65530
 neighbor 10.0.4.4 remote-as 65530
```

Router D configuration:

```
!
```

```
router bgp 65530
 bgp confederation identifier 100
 bgp confederation peers 65531
 bgp log-neighbor-changes
 neighbor 10.0.2.4 remote-as 65530
 neighbor 10.0.3.4 remote-as 65530
 neighbor 192.168.5.3 remote-as 65531
 neighbor 192.168.12.7 remote-as 200
```

Router E configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.12.6 remote-as 100
```

7.26.6 Configuring Route Reflector

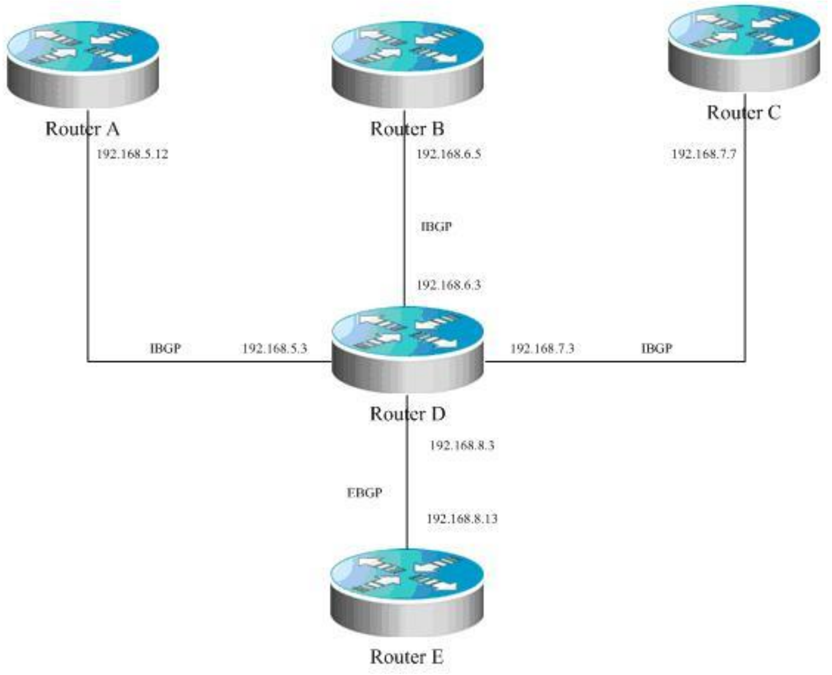
When a route reflector is configured, use the **bgp client-to-client reflection** command to enable the route reflection function on the device. If there are more than one route reflector within one cluster, use the **bgp cluster-id** command to configure the cluster ID of the reflector, and use the **neighbor route-reflector-client** command to add the peer to the client for route reflection.

The configuration is detailed as follows:

```
router bgp 601
bgp cluster-id 200.200.200.200
neighbor 171.69.232.56 remote-as 601
neighbor 200,200,200,205 remote-as 701
neighbor 171.69.232.56 route-reflector-client
```

The following example shows the relastionship between and IP addresses of the devices:

Figure 7



In this example, Router D is a route reflector. The following section shows the configurations of different devices:

Router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.5.3 remote-as 100
 neighbor 192.168.5.3 description route-reflector server
```

Router B configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.6.3 remote-as 100
 neighbor 192.168.6.3 description route-reflector server
```

Router C configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.7.3 remote-as 100
 neighbor 192.168.7.3 description not the route-reflector server
```

Router D Configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.5.12 remote-as 100
 neighbor 192.168.5.12 description route-reflector client
 neighbor 192.168.5.12 route-reflector-client
 neighbor 192.168.6.5 remote-as 100
 neighbor 192.168.6.5 description route-reflector client
 neighbor 192.168.6.5 route-reflector-client
 neighbor 192.168.7.7 remote-as 100
 neighbor 192.168.7.7 description not the route-reflector client
 neighbor 192.168.8.13 remote-as 200
```

Router E configuration:

```
!
router bgp 500
 bgp log-neighbor-changes
 neighbor 192.168.8.3 remote-as 100
```

7.26.7 Configuring Peergroup

This section uses the configuration of **peergroup** for IBGP and EBGP as an example.

7.26.7.1 Configuring IBGP peergroup

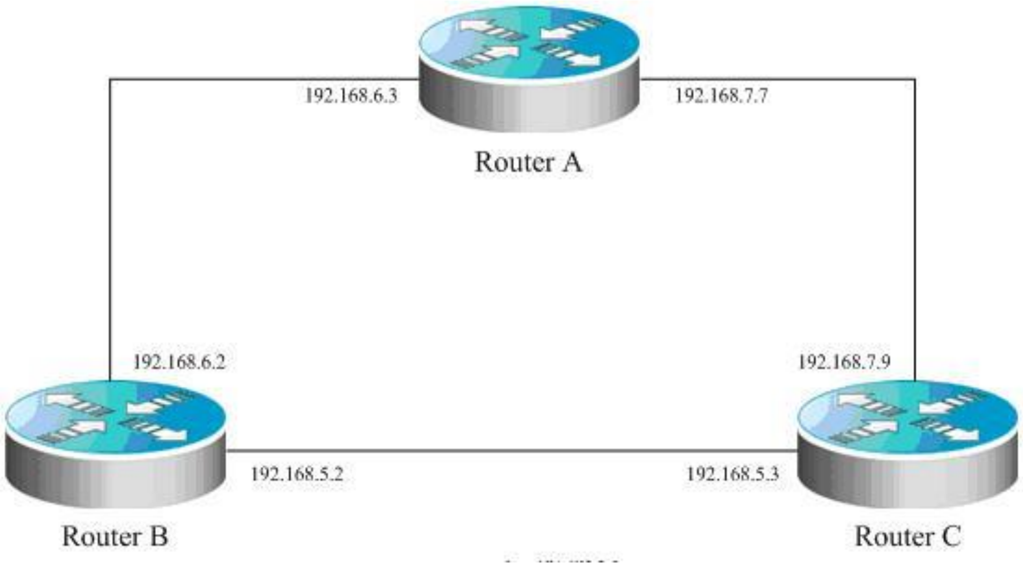
Use the **neighbor** *internal* **peer-group** command to create a peer group named *internal*, and configure a remote AS, and other options for the peer group. Use the **neighbor** *A.B.C.D* **peer-group** *internal* command to add peers A.B.C.D into the peer group.

The configuration commands are described as follows:

```
router bgp 100
neighbor internal peer-group
neighbor internal remote-as 100
neighbor internal update-source loopback 0
neighbor internal route-map set-med out
neighbor internal filter-list 1 out
neighbor internal filter-list 2 in
neighbor 171.69.232.53 peer-group internal
neighbor 171.69.232.54 peer-group internal
neighbor 171.69.232.55 peer-group internal
neighbor 171.69.232.55 filter-list 3 in
```

The following example shows the relastionship between and IP addresses of the devices:

Figure 8



Router A configuration

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-group peer-group
 neighbor ibgp-group description peer in the same as
 neighbor 192.168.6.2 remote-as 100
 neighbor 192.168.6.2 peer-group ibgp-group
 neighbor 192.168.6.2 description one peer in the ibgp-group
 neighbor 192.168.7.9 remote-as 100
 neighbor 192.168.7.9 peer-group ibgp-group
```

Router B configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-peer peer-group
 neighbor ibgp-peer remote-as 100
 neighbor ibgp-peer route-map ibgp-rmap out
 neighbor 192.168.5.3 peer-group ibgp-peer
 neighbor 192.168.5.3 route-map set-localpref in
 neighbor 192.168.6.3 peer-group ibgp-peer
```

Router C configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-group peer-group
 neighbor 192.168.5.2 remote-as 100
 neighbor 192.168.5.2 peer-group ibgp-group
 neighbor 192.168.7.7 remote-as 100
 neighbor 192.168.7.7 peer-group ibgp-group
```

7.26.7.2 Configuring EBGP peergroup

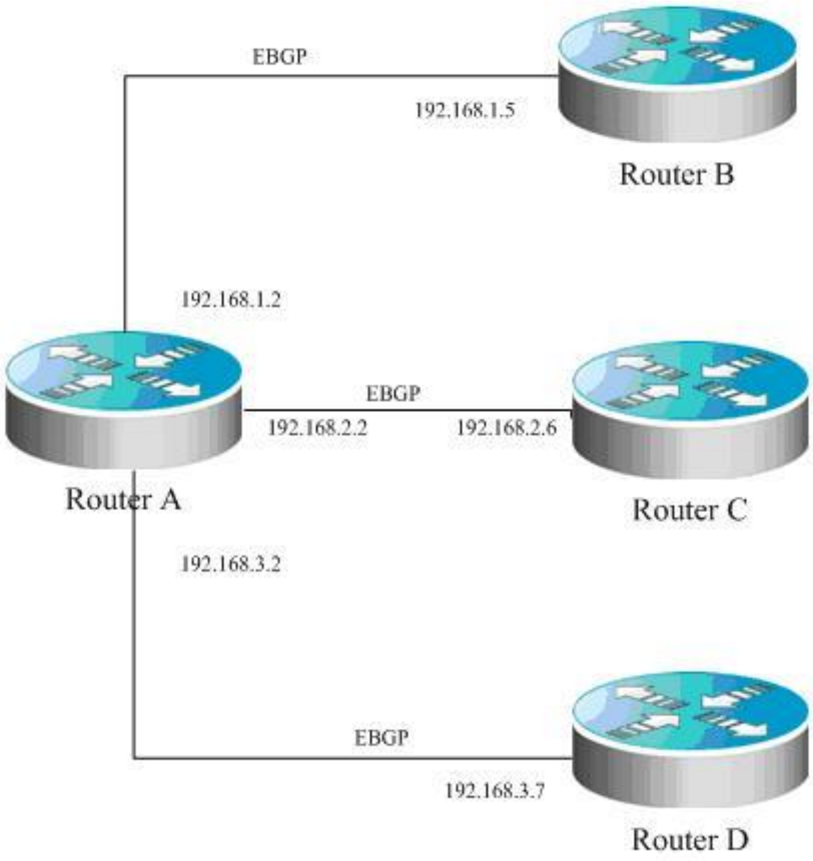
Use the **neighbor** *A.B.C.D* **remote-as** *num* command to configure an EBGP peer. Use the **neighbor** *external* **peer-group** command to create a peer group named **external**, and apply the **neighbor** *A.B.C.D* **peer-group** *external* command to add the peers A.B.C.D into the peer group *external*.

Here is an example of the specific configuration:

```
router bgp 100
neighbor external-peers peer-group
neighbor external-peers route-map set-metric out
neighbor external-peers filter-list 99 out
neighbor external-peers filter-list 101 in
neighbor 171.69.232.90 remote-as 200
neighbor 171.69.232.90 peer-group external-peers
neighbor 171.69.232.100 remote-as 300
neighbor 171.69.232.100 peer-group external-peers
neighbor 171.69.232.110 remote-as 400
neighbor 171.69.232.110 peer-group external-peers
neighbor 171.69.232.110 filter-list 400 in
```

The following figure shows the configuration of peer-group:

Figure 9



The figure illustrates the relationship between devices and the assignment of IP address.

Router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ebgp-group peer-group
 neighbor ebgp-group distribute-list 2 in
 neighbor ebgp-group route-map set-med out
 neighbor 192.168.1.5 remote-as 200
 neighbor 192.168.1.5 peer-group ebgp-group
 neighbor 192.168.2.6 remote-as 300
 neighbor 192.168.2.6 peer-group ebgp-group
 neighbor 192.168.2.6 distribute-list 3 in
 neighbor 192.168.3.7 remote-as 400
 neighbor 192.168.3.7 peer-group ebgp-group
!
```

Router B configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 100
!
```

Router C configuration:

```
!
router bgp 300
```

```
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 100
!
```

Router D configuration:

```
!
router bgp 400
 bgp log-neighbor-changes
 neighbor 192.168.3.2 remote-as 100
!
```

7.26.8 Configuring TCP MD5

Use the CLI command **neighbor password** to configure TCP MD5 for the BGP connection in BGP configuration mode.

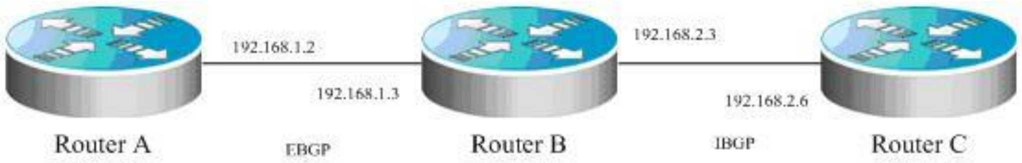
The configuration format is shown as follows:

```
router bgp 100
neighbor 171.69.232.54 remote-as 110
neighbor 171.69.232.54 password peerpassword
```

Configure the *password* of peer 171.69.232.54 as *peerpassword.*

The following figure shows the configuration of MD5 and IP addresses on different devices:

Figure 10



The AS of router A is 100, and the AS of router B and router C is 200. Router A establishes EBGP adjacency with router B and uses EBGP as the MD5 password. Router B establishes IBGP adjacency with router C and uses IBGP as the MD5 password.

router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.1.3 remote-as 200
 eighbor 192.168.1.3 password ebgp
!
```

Router B configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 100
 neighbor 192.168.1.2 password ebgp
 neighbor 192.168.2.6 remote-as 200
 neighbor 192.168.2.6 password ibgp
!
```

Router C configuration:

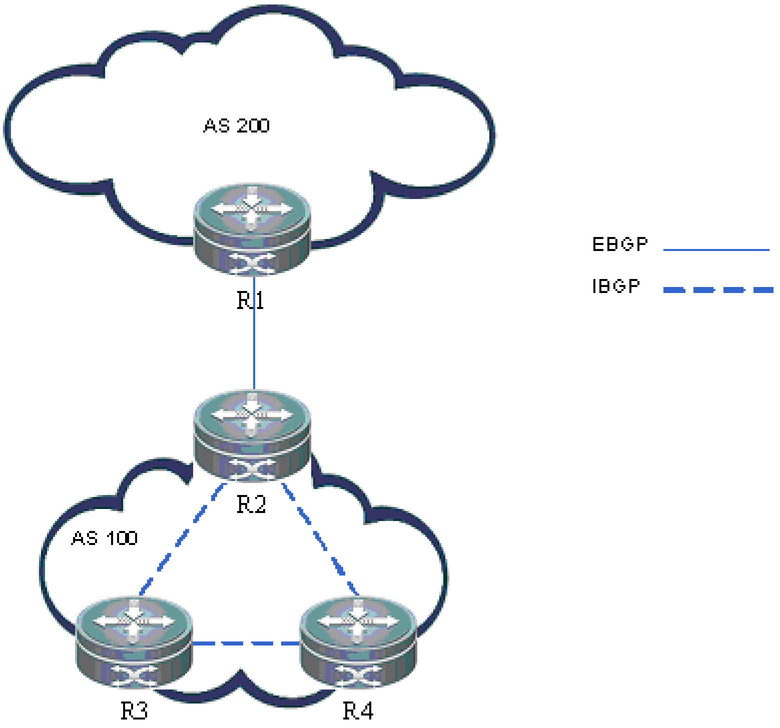
```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.2.3 remote-as 200
 neighbor 192.168.2.3 password ibgp
!
```

7.26.9 Configuring BGP GR

7.26.9.1 Networking Requirements

As shown in the following figure, R2 is the border device of AS100 and AS200. R1 is the access device of AS200. In AS100, R2, R3 run OSPF to offer IBGP connection for the BGP protocol. At the same time, IBGP connections are established between them. R2 establishes an EBGP conneciton with R1. R2, as the border device connecting AS100 and AS200, must be more reliable. R2 is configured to support dual-system redundant bakcup for continuous forwarding and graceful restart of routing protocols (OSPF and BGP in this example). The graceful restart of routing protocols involves adjacent devices. Hence, R1, 3 and R4 need to support the BGP GR capability, and R3 and R4 need to support the GR Helper of OSPF to support the OSPF GR capabiltiy. In this way, when one engine of R2 fails, the transmission of data is not interrupted and therefore reliability is enhanced.

Figure 11 BGP GR configuration example



7.26.9.2 Confiugration Precautions:

Before configuration, ensure that R2 can serve as the GR Restarter for graceful restart and the software on all devices support OSPF GR and BRP GR capability. If not, continuous data forwarding cannot be peformed when the backup engine takes over the work of the master engine in case of failures. Meanwhile, the BGP protocol depends on the BGP connection from OSPF. Hence, both the BGP and OSPF protocols should have GR enabled. Therefore, R2 must support OSPF GR.

14) R2 enables dual-eningle redundnat hot backup;

15) The software of all devices support OSPF GR and BGP GR capability

16) OSPF GR is enabled on R2

17) BGP GR is enabled on R2

18) BGP GR is enabled on neighbors to support the GR Helper of BGP

19) All BGP connections restart on R2 to negotitate GR capabitliy

7.26.9.3 Configuration Steps

20) Ensure that R2 enables dual-eningle redundnat hot backup

21) Ensure that the software of all devices supports OSPF GR and BGP GR capabilities.

Check that these devices support the configuration commands of BGP GR and OSPF GR. For details, refer to Step 3 and Step 4.

22) Enables OSPF GR on R2

```
Qtech(config)# router ospf 1
Qtech(config-router)# graceful-restart
```

23) Enable BGP GR on R2

```
Qtech(config)# router bgp 100
Qtech(config-router)# graceful-restart
```

24) Enable BGP GR on neighbors to support the GR Helper of BGP

```
Qtech(config)# router bgp 100
Qtech(config-router)# bgp graceful-restart
```

For BGP GR negotiation, both sides of the BGP connection must enable BGP GR. Hence, R2 needs to negotiate with its neighbors which serve as the GR Helper to assist BGP GR.

25) Restart all BGP connections on R2 to neotitate GR capabitliy

You must manually restart the BGP connection for GR capability renegotiation, because the **bgp graceful-restart** command cannot take effect immediately.

```
Qtech# clear ip bgp *
```

7.26.9.4 Configuration Check

For R2 to enable continuous data forwarding during engine handover, check the negotiation of BGP GR and OSPF GR configration.

26) Ensure that BGP GR can negitate with all neighbors.

```
 Qtech# show ip bgp neighbors
BGP neighbor is 192.168.195.183, remote AS 200, local AS 100, external link
Using BFD to detect fast fallover - BFD session state up
  BGP version 4, remote router ID 10.0.0.1
  BGP state = Established, up for 00:06:37
  Last read 00:06:37, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
Address family IPv4 Unicast: advertised and received
Graceful restart: advertised and received
  Remote Restart timer is 120 seconds
  Address families preserved by peer:
    None
```

Graceful restart: advertised and received means BGP GR negotiation of the BGP connection is successful. Ensure that BGP GR can negotiate with all BGP connections.

27) Ensure that OSPF GR is enabled on R2.

```
Qtech# show ip ospf
Routing Process "ospf 1" with ID 10.0.0.2
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583Compatibility flag isenabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
This router is an ASBR (injecting external routing information)
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
LsaGroupPacing: 240 secs
Number of incomming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 4. Checksum 0x0278E0
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 4
External LSA database is unlimited.
Number of LSA originated 6
Number of LSA received 2
Log Neighbor Adjency Changes : Enabled
Graceful-restart enabled
Graceful-restart helper support enabled
Number of areas attached to this router: 1
Area 0 (BACKBONE)
```

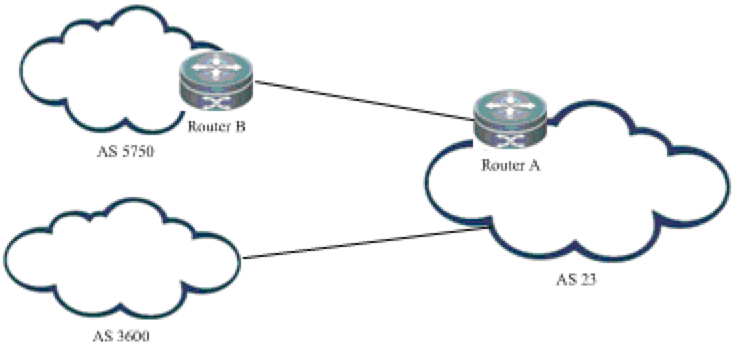
Graceful restart enabled means OSPF GR is enabled.

7.26.10 Configuring BGP Local AS

Networking requirements

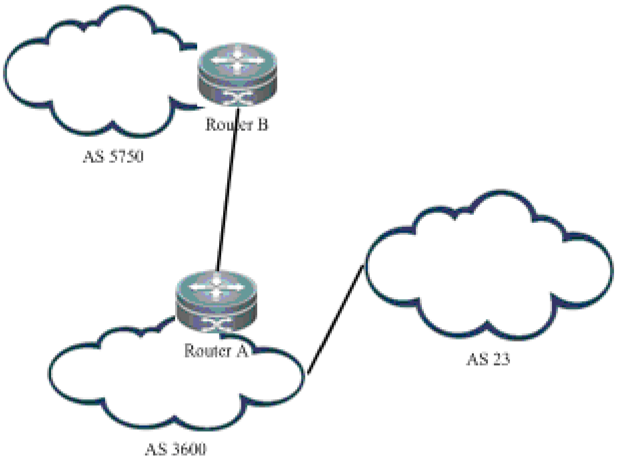
As shown in the following figure, Router A and its home network are located in AS 23, which connects AS 3600 through EBGP. The routing information of AS 5750 is transmitted to AS 3600 via AS 23.

Figure 12 Logical topology before AS migration



You must migrate Router A and its home network to AS 3600.

Figure 13 Logical topology after AS migration

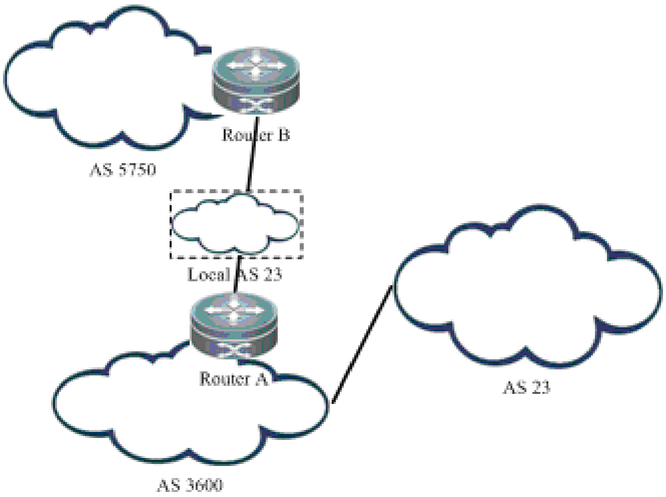


AS 23 and AS 3600 belong to one management domain. The configurations of these two ASs are modified after negotiation. At this point, Router A configures AS 3600 as the AS of the BGP protocol. In this case, you need to maintain the BGP connection between Router A and Router B, and modify related peer configuration on Router B of AS 5750. Sometimes Router B may not modify the configuration immediately. As a result, Router B cannot establish the BGP connection with Router A. On Router A, you can configure local AS for Router B to establish a BGP connection between them wihtout affecting the transmisison and calculation of routes.

7.26.10.1 Networking Topology

The following figure illustrates how to configure local AS for Router B.

Figure 14 Configuration of local AS



After configuration, a virtual AS 23 is set up between Router A and Router B. Router B considers that it is directly connected to AS 23 and can transmit routes to AS 23. This removes the need to modify Router B's configuration. When AS 3600 reaches an agreement with AS 5750 in terms of management, Router B can modify the remote AS of Router A as AS 3600 and Router A deletes the corresponding local AS for migration of network in different ASs.

7.26.10.2 Configuration Steps

28) Enter BGP configuration mode

```
Qtech-A(config)# router bgp 3600
```

29) Configures local AS for the peer

```
Qtech-A(config-router)# neighbor 57.50.1.1 local-as 23 no-prepend replace-as dual-as
```

30) Delete local AS after Router B modifies its configuration

```
Qtech-A(config-router)#no neighbor 57.50.1.1 local-as
```

7.26.10.3 Configuration Check

```
Use the show ip bgp neighbors command to verify the local AS used by a neighbor to
establish a BGP connection as follows:
Qtech-A#show ip bgp neighbors 57.50.1.1
BGP neighbor is 57.50.1.1,  remote AS 5750, local AS 23(using Peer's Local AS, no-
prepend, replace-as, dual-as), external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read         , hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
    open message:0 update message:0 keepalive message:0
    refresh message:0 dynamic cap:0 notifications:0
  Sent 0 messages, 0 notifications, 0 in queue
```

7.26.10.4 Detailed Configuration:

Router A configuration

```
router bgp 3600
neighbor 57.50.1.1 remote-as 5750
neighbor 57.50.1.1 local-as 23 no-prepend replace-as dual-as
neighbor 57.50.1.1 update-source loopback 0
neighbor 57.50.1.1 ebgp-multihop 255
```

Router B configuration

```
router bgp 5750
neighbor 36.0.1.1 remote-as 23
neighbor 36.0.1.1 update-source loopback 0
neighbor 36.0.1.1 ebgp-multihop 255
```

8 CONFIGURING BGP MCE

8.1 About BGP MCE

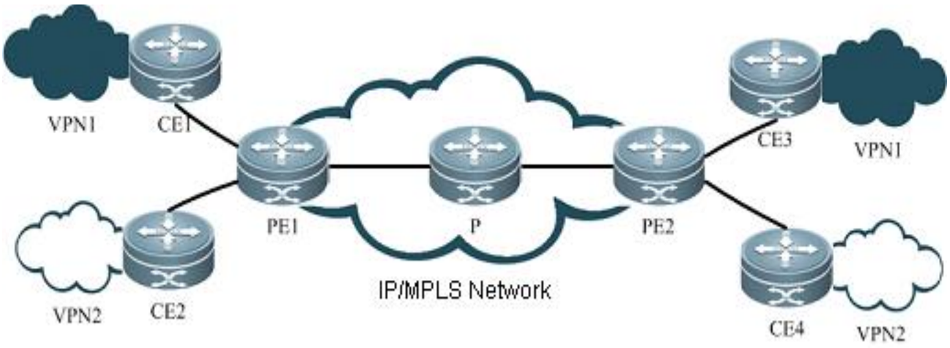
8.1.1 MCE Overview

MCE refers to Multi-CE. MCE enabled network devices can function as CEs of multiple VPN instances in a BGP/MPLS VPN network. This helps reduce the need for additional network equipment.

8.1.2 Working Principle of BGP MCE

With BGP/MPLS VPN, private network data can be securely transmitted in a public network through tunnels. However, in a typical BGP/MPLS VPN network, each VPN is connected to the PE through a CE, as shown in Figure 15:

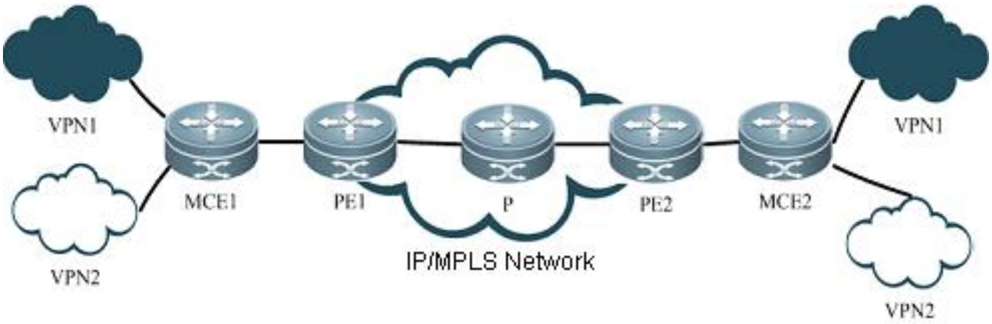
Figure 15 BGP/MPLS VPN network



As users' demand surges for service segmentation and security, a private network may be divided into multiple VPNs, and the users of different VPNs are usually isolated from each other. As a result, equipment and maintenance costs may be increased by assigning a CE for each VPN, while data security cannot be guaranteed by sharing one CE and using the same routing entry among multiple VPNs. MCE can balance data security and networking cost. By binding the VLAN interfaces of a CE device to the VPNs, you can create and maintain a routing table for each of the VPNs (Multi-VRF). In this way, packets of different VPNs in the private network can be isolated. Moreover, the PE enables the routes of each VPN to be advertised to the corresponding remote PE. As such, packets of each VPN can be transmitted securely through the public network.

The following example shows how the MCE maintains routing entries of multiple VPNs and how the MCE exchanges VPN routes with PEs.

Figure 16 MCE functions



As shown in Figure 16, two VPN sites on the left side (VPN1 and VPN2) are connected to the MPLS backbone through an MCE device. Users of VPN1 and VPN2 need to establish VPN tunnels with remote VPN1 and VPN2 users. MCE enables routing tables to be created for VPN1 and VPN2 individually on the MCE device. VLAN-interface 2 can be bound to VPN1, and VLAN-interface 3 can be bound to VPN 2. When receiving routing information, MCE determines the source of information based on the number of the interface receiving the information and then maintains the corresponding VPN routing table. Meanwhile, you need to bind the MCE-connecting interfaces on PE1 to the VPNs in the same way as those on the MCE device. The MCE device is connected to PE1 through a trunk, which permits packets of VLAN2 and VLAN3 carrying VLAN tags. In this way, PE1 can determine the home VPN of a received packet according to the VLAN tag and passes the packet to the corresponding tunnel.

How does MCE device accurately transmit private routing information of multiple VPN instances to PEs? This involves two steps: routing information exchange between MCE and VPN site, and between MCE and PE. There are several ways to exchange routing information, such as static route, RIP, OSPF, ISIS and BGP. If BGP routing protocol is used to exchange routing information, BGP MCE applies. Specifically, BGP MCE allows BGP protocol to support VRF and enable BGP routing information exchange under VRF. We need to configure the BGP peer for each VRF instance on MCE and introduce IGP routing information of corresponding VPN. As each VPN is generally in different ASs, EBGP is therefore used to advertise routes.

8.1.3 Protocol Specification

NA.

8.2 Default Configurations

The following table describes the default configurations of BGP MCE.

Function	Default setting
VRF instance	No VRF instance is created by default.
BGP-VRF binding	No BGP-VRF binding by default

The following products support BGP MCE: the RSR30, RSR50 and RSR50E series of routers.

8.3 Configuring BGP MCE

8.3.1 Configuring VRF Instance and Route-related Attributes

Before configuring the BGP MCE, VRF instance and route-related attributes must be configured first:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode
Qtech(config)# **ip vrf** *VRF1*	Creates a VRF named VRF1 and enters VRF mode.
Qtech(config-vrf)# **rd** *rd-value*	Configures VRF RD value, which is identified using XX:XX format, such as RD 1:100. 1 refers to the AS ID of backbone network, while 100 is a user-defined numerical value.
Qtech(config-vrf)# **route-target both** | **export** | **import**} *rt-value*	Configures the route export and import RT attribute of VRF.
Qtech(config-vrf)# {**export** | **import**} **map** *map*	Configures the route map for import and export routes to support policy-based filtering of import and export routes.
Qtech(config-vrf)# **exit**	Exits VRF mode and enters global configuration mode
Qtech(config)# **interface vlan *2***	Enters VLAN 2 interface configuration mode.
Qtech(config-if)# **ip vrf forwarding** *VRF1*	Associates interface with VRF instance of VRF2
Qtech(config-if)# **ip address 172.16.25.18 255.255.255.0**	Configures IP address for VLAN 2
Qtech(config-if)# **end**	Returns to privileged EXEC mode
Qtech # **show running-config**	Verifies the configurations
Qtech # **write**	(Optional) Saves configurations.

8.3.2 Configuring BGP Route Exchange between MCE and VPN Site

To use the BGP protocol to exchange routing information between MCE and VPN sites, you need to bind BGP to the corresponding VRF instance on MCE, and configure site device as EBGP neighbor, as shown below:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode
Qtech(config)# **router bgp *23***	Enables the BGP protocol and enters BGP routing process mode
Qtech(config-router)# **address-family ipv4 vrf** *VRF1*	Enters the IPv4 address family configuration mode of VRF1
Qtech(config-router-af)# **neighbor** 172.16.25.57 **remote-as** 65531	Configures EBGP neighbor and learns routing information advertised by VPN site through BGP.
Qtech(config-router-af)# **redistribute** *ospf 1*	Introduces the routing information of remote VPN as advertised by PE. We assume that MCE and PE exchange routing information through OSPF protocol.
Qtech(config- router-af))# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves configurations.

BGP protocol must also be enabled on CE devices at a VPN site, allowing the site to exchange routing information with MCE devices through BGP protocol.

8.3.3 Configuring BGP Route Exchange between MCE and PE

To use the BGP protocol to exchange routing information between MCE and PE, bind BGP to the corresponding VRF instance on MCE, and configure PE device as EBGP neighbor, as shown below:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp *23***	Enables BGP protocol and enter BGP routing mode.
Qtech(config-router)# **address-family ipv4 vrf** *VRF1*	Enters IPv4 address family configuration mode of VRF1.
Qtech(config-router-af)# **neighbor** 172.16.25.157 **remote-as** 65532	Configures EBGP neighbor and study the routing information advertised by PE through BGP.
Qtech(config-router-af)# **redistribute** *ospf 1*	Introduces the routing information of local VPN. We assume that MCE and local VPN site exchange routing information through OSPF protocol.
Qtech(config- router-af))# **end**	Returns to privileged EXEC mode.
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves configurations.

BGP protocol must also be enabled on the PE device, and MCE must be configured as EBGP neighbor, allowing PE to exchange routing information with the MCE device through BGP protocol.

8.3.4 Displaying Configurations

The "show" commands used in BGP MCE are similar to the "show" commands used in ordinary BGP. The following information is displayed: neighbor state, routing information and neighbor summary.

Command	Function
Qtech # **show ip vrf**	Displays all VRF summary information on the device.
Qtech # **show ip vrf detail [*VRF1*]**	Displays the detailed configurations of all VRFs or a specified VRF.
Qtech # **show ip vrf interfaces** [*VRF1*]	Displays the interface binding information and state of all VRFs or a specified VRF.
Qtech# show **ip bgp vrf** *VRF1* [**summary**| **neighbors**| *A.B.C.D*]	Displays the summary information, detailed information, specific routing information, and all routing information of BGP neighbor under VRF1. Similar to those "show" commands used in ordinary BGP, other sub-commands are not discussed herein.
Qtech# **show bgp vpnv4 unicast** [**all** | **rd** *rd* | **vrf** *vrf-name*] [**neighbors** | **summary** | *A.B.C.D*]	This command is similar to the above command, but the routes displayed are different: "all" will display all vpn routes, "rd" will display vpn routes with a specified RD value, and "vrf" will display vpn routes under a specified VRF.

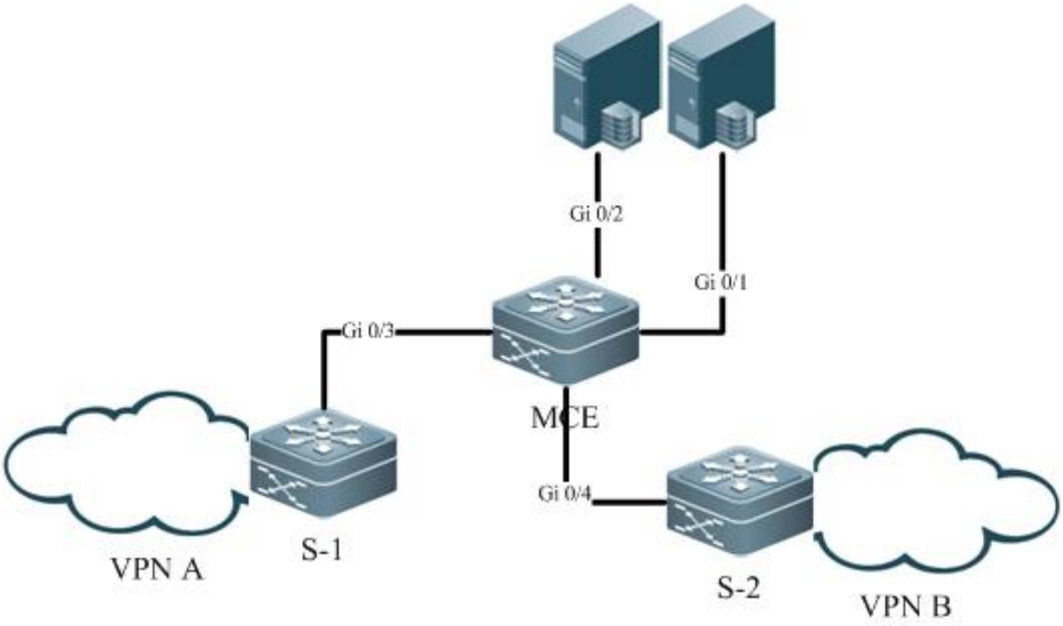
8.4 Typical BGP MCE Configuration Examples

8.4.1 Networking Requirements

A company needs to isolate the networks of two subsidiaries: A and B, and expects both subsidiaries to access the resource servers at the same time. OSPF protocol operates on the networks of subsidiary A and subsidiary B. An MCE device is used to isolate A and B. The device is directly connected with multiple resource servers.

8.4.2 Network Topology

Figure 17 Network topology of BGP MCE



S-1 is the convergence device on the network of subsidiary A; S-2 is the convergence device on the network of subsidiary B. S-1 and S-2 are both connected with the MCE device, with connecting interfaces belonging to different VRF instances and running the OSPF routing protocol. Gi 0/1 and Gi 0/2 of the MCE is directly connected with resource servers, and belong to another separate VRF instance.

8.4.3 Configuration Tips

We assume that OSPF protocol is running normally on VPN A connected directly with S-1 and VPN B connected directly with S-2. The following configurations are related to the MCE device.

31) Configure VRF instances and associate with interfaces to allow network isolation;

32) Configure OSPF routing protocol and associate with VRF, so that MCE can learn the routes to respective subsidiaries;

33) Configure BGP routing protocol and import OSPF routes and directly connected routes, allowing route exchange between different VRFs;

34) Configure the import and export route attributes of VRF instance, so that both subsidiaries can access the resource servers;

8.4.4 Configuration Steps

35) Configures VRF instances and associate with interfaces;

# Create three VRF instances: VRF1, VRF2 and VRF3

```
Qtech# config terminal
```

```
Qtech(config)# ip vrf VRF1
Qtech(config-vrf)# rd 100:1
Qtech(config-vrf)# exit
Qtech(config)# ip vrf VRF2
Qtech(config-vrf)# rd 100:2
Qtech(config-vrf)# exit
Qtech(config)# ip vrf VRF3
Qtech(config-vrf)# rd 100:3
Qtech(config-vrf)# exit
```

# Bind Gi0/1 and Gi0/2 to VRF3, Gi0/3 to VRF1 and Gi0/4 to VRF2

```
Qtech(config)#interface gigabitEthernet 0/1
Qtech(config-GigabitEthernet 0/1)#ip vrf forwarding VRF3
Qtech(config-GigabitEthernet 0/1)#ip address 10.1.1.1 255.255.255.0
Qtech(config)#interface gigabitEthernet 0/2
Qtech(config-GigabitEthernet 0/2)#ip vrf forwarding VRF3
Qtech(config-GigabitEthernet 0/2)#ip address 10.1.2.2 255.255.255.0
Qtech(config)#interface gigabitEthernet 0/3
Qtech(config-GigabitEthernet 0/3)#ip vrf forwarding VRF1
Qtech(config-GigabitEthernet 0/3)#ip address 172.16.25.18 255.255.255.0
Qtech(config)#interface gigabitEthernet 0/4
Qtech(config-GigabitEthernet 0/4)#ip vrf forwarding VRF2
Qtech(config-GigabitEthernet 0/4)# ip address 192.168.25.18 255.255.255.0
```

36) Configure OSPF routing protocol and associate with VRF;

# Create OSPF 1 and OSPF 2 and associate with VRF1 and VRF2

```
Qtech# config terminal
Qtech(config)# router ospf 1 vrf VRF1
Qtech(config-router)# network 172.16.25.0 0.0.0.255 area 0
Qtech(config-router)# exit
Qtech(config)# router ospf 2 vrf VRF2
Qtech(config-router)# network 192.168.25.0 0.0.0.255 area 0
Qtech(config-router)# exit
```

37) Configure BGP protocol and import routes

# Configure BGP protocol and import OSPF routes to BGP VRF instance

```
Qtech# config terminal
Qtech(config)# router bgp 100
Qtech(config-router)# address-family ipv4 vrf VRF1
Qtech(config-router-af)# redistribute ospf 1
Qtech(config-router-af)# exit
Qtech(config-router)# address-family ipv4 vrf VRF2
Qtech(config-router-af)# redistribute ospf 2
```

# Import the directly connected routes of VRF3 to BGP VRF3 instance

```
Qtech(config-router)# address-family ipv4 vrf VRF3
Qtech(config-router-af)# redistribute connect
```

38) Configure the import and export route attributes of VRF instance

# Configure the import route attribute of VRF1 as 100:3 and export attribute as 100:1

```
Qtech(config)# ip vrf VRF1
Qtech(config-vrf)# route-target import 100:3
Qtech(config-vrf)# route-target export 100:1
Qtech(config-vrf)# exit
```

# Configure the import route attribute of VRF2 as 100:3 and export attribute as 100:2

```
Qtech(config)# ip vrf VRF2
Qtech(config-vrf)# route-target import 100:3
Qtech(config-vrf)# route-target export 100:2
Qtech(config-vrf)# exit
```

# Configure the import route attribute of VRF3 as 100:1 and 100:2 and export attribute as 100:3

```
Qtech(config)# ip vrf VRF2
Qtech(config-vrf)# route-target import 100:1 100:2
Qtech(config-vrf)# route-target export 100:3
Qtech(config-vrf)# exit
```

8.4.5 Verification

Execute the following steps to verify configurations:

39) Verify the state of interfaces bound to VRF. Execute the **show ip vrf interface** command to verify interface binding information and the interface state.

40)

```
Qtech#sh ip vrf interfaces
Interface             IP-Address      VRF                     Protocol
GigabitEthernet 0/1  10.1.1.1        VRF3                   up
GigabitEthernet 0/2  10.1.2.1        VRF3                   up
GigabitEthernet 0/3  172.16.25.18   VRF1                   up
GigabitEthernet 0/4  192.168.25.18  VRF2                   up
```

41) Verify whether OSPF protocol bindings are correct and whether OSPF protocol runs normally. Execute the **show ip ospf** command to verify whether the OSPF instance is properly bound to VRF;

42) Verify whether the routes imported by BGP instance are correct. Execute the **show ip bgp vrf** or **show bgp vpnv4 unicast** command to verify whether the imported routes are correct, as shown below:

```
Qtech#sh ip bgp vrf VRF3
BGP table version is 1, local router ID is 10.14.219.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S
Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop             Metric  LocPrf      Weight Path
*> 10.1.1.0/24    0.0.0.0              0         32768       ?
*> 10.1.2.0/24    0.0.0.0              0         32768       ?
Total number of prefixes 2
```

43) Execute the **show ip bgp vrf** command to verify whether routes of other VRFs have been properly imported to local VRF. Execute the **show ip route vrf** command to verify whether routes are correct.

```
Qtech#sh ip bgp vrf VRF3
BGP table version is 1, local router ID is 10.14.219.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S
Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop             Metric  LocPrf      Weight Path
*> 10.1.1.0/24    0.0.0.0              0         32768       ?
*> 10.1.2.0/24    0.0.0.0              0         32768       ?
*> 172.16.22.0/24 0.0.0.0              0         32768       ?
*> 172.16.23.0/24 0.0.0.0              0         32768       ?
*> 172.16.25.0/24 0.0.0.0              0         32768       ?
*> 192.168.22.0     0.0.0.0              0         32768       ?
*> 192.168.23.0     0.0.0.0              0         32768       ?
*> 192.168.25.0     0.0.0.0              0         32768       ?
Total number of prefixes 8
Qtech#sh ip route vrf VRF1
Routing Table: VRF1
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
```

```
Gateway of last resort is no set
C    10.1.1.0/24 is directly connected, GigabitEthernet 0/1
C    10.1.1.1/32 is local host.
C    10.1.2.0/24 is directly connected, GigabitEthernet 0/2
C    10.1.2.1/32 is local host.
B    172.16.22.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    172.16.23.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    172.16.25.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    192.168.22.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
B    192.168.23.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
B    192.168.25.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
```

9 CONFIGURING BGP 4-OCTET AS

9.1 About 4-octet AS

9.1.1 Overview

A traditional AS number consists of two octets within the range of 1-65535. The AS number defined by RFC4893 consists of four octets falling within the range of 1-4294967295 to ease the burden of AS number resources. According to RFC5396, a 4-octet AS number supports two representation formats: asplain and asdot+. The two representations take the same format. Specifically, the 4-octet AS number will be represented using decimal value. The asdot+ representation contains ([high order 2 octets.] low order 2 octets). The high order 2 octets are not displayed if the value is 0. In other words, the AS number of 65536 in asplain format will be represented as 1.0 in asdot+ format. In addition, the AS number of 65534 in asplain format will be represented as 65534 in asdot+ format (without displaying the 0 value).

9.1.2 Working Principle

The 4-octet AS number requires a BGP connection between an old bgp speaker supporting only 2-octet AS number and a new bgp speaker supporting 4-octet AS number. If the autonomous system for the new bgp speaker uses a 4-octet AS number, the old bgp speaker must use the reserved AS number of 23456 to replace the 4-octet AS number of new bgp speaker while creating a neighbor. In the packets sent from the new bgp speaker to the old bgp speaker, 23456 will replace the 4-octet AS number in the domain of "My Autonomous System". Meanwhile, in the UPDAT packets sent to the old bgp speaker, 23456 will replace the 4-octet AS number found in AS-PATH and AGGREGATOR attributes. These packets also carry the true 4-octet AS number reserved in the optional transitive attributes of AS4-PATH and AS4-AGGREGATOR. Therefore, the true AS-PATH attribute and AGGREGATOR attribute can be restored when this route reaches the next new bgp speaker.

In other cases, the true AS number of peer side is directly used to create a neighbor.

9.1.3 Protocol Specification

RFC 4893

RFC 5396

9.2 Default Configurations

By default, BGP protocol is not enabled. After BGP protocol is enabled, the decimal value is used by default to represent 4-octet AS numbers.

9.3 Configuring BGP 4-octet AS

9.3.1 Configuring BGP Instance with 4-octet AS Number

Command	Function
Qtech # **configure terminal**	Enters global configuration mode
Qtech(config)# **router bgp 65538**	Enables BGP protocol and configure device AS number as 65538
Qtech(config)# **router bgp 1.2**	Uses asdot+ format 1.2 to represent four-octet AS number of 65538
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **write**	(Optional) Saves configurations.

9.3.2 Configuring the Display Format of 4-octet AS Number

By default, the asplain format is used to display a 4-octet AS number. You can also configure the display format as asdot+. Meanwhile, after changing the display format of a 4-octet AS number, the 4-octet AS number in regular expression will be matched using asdot+ format.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode
Qtech(config)# **router bgp 65538**	Enables BGP protocol and configures the device AS number as 65538
Qtech(config-router)# **bgp asnotation dot**	Use the asdot+ format to display 4-octet AS number, namely 1.2
Qtech(config-router)# **end**	Returns to privileged EXEC mode.
Qtech # **clear ip bgp** *	Resets BGP protocol for re-matching the regular expression.
Qtech # **write**	(Optional) Saves configurations.

After executing the **bgp asnotation dot** command, you must execute the **clear ip bgp *** command to reset BGP protocol, so that the regular expression can be rematched.

9.3.3 Displaying Configurations

Execute the **show** command to view the configuration of the 4-octet AS number. This command is similar to the **show** command used in BGP mode.

Command	Function
Qtech # **show ip bgp summary**	Displays the connection state of all BGP neighbors.

9.4 Typical BGP 4-Octet AS Configuration Examples

9.4.1 Interconnection between 4-octet AS and 2-octet AS

9.4.1.1 Networking Requirements

44) A BGP connection is established between the router supporting 2-octet AS number and the router supporting 4-octet AS number (using 2-octet AS number);

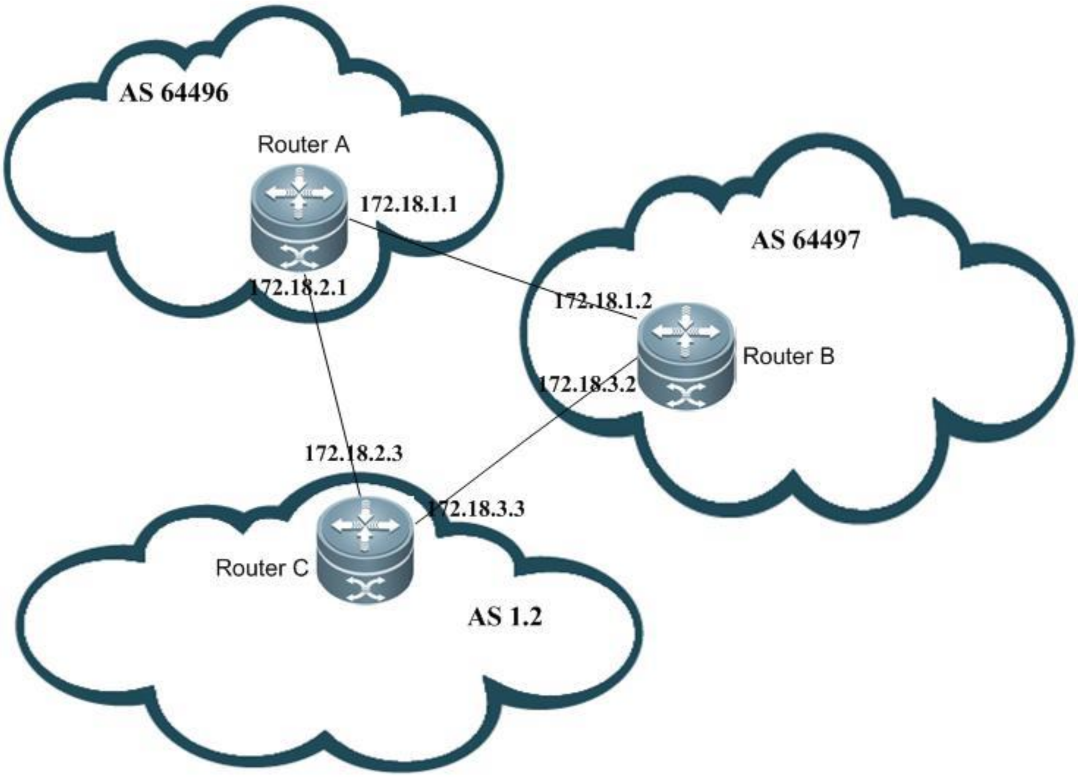
45) A BGP connection is established between the router supporting 2-octet AS number and the router supporting 4-octet AS number (using 4-octet AS number);

46) A BGP connection is established between routers supporting 4-octet AS number, with one router using 2-octet AS number and the other router using 4-octet AS number.

9.4.1.2 Network Topology

As shown in the figure below, Router A, Router B and Router C are edge routers of three autonomous systems, and BGP connections have been established between them. Router A only supports 2-octet AS numbers; Router B and Router C support 4-octet AS numbers. The autonomous system of Router A uses a 2-octet AS number of 64496; the autonomous system of Router B uses a 2-octet AS number of 64497; the autonomous system of Router C uses a 4-octet number of 1.2.

Figure 18 BGP 4-Octet AS configuration



9.4.1.3 Configuration Tips

47) Router A cannot recognize the 4-octet AS number of 1.2 used by the autonomous system of Router C. When a neighbor is created, the reserved AS number of 23456 must replace 1.2 during the configuration of remote-as.

48) Although Router B supports 4-octet AS numbers, it still uses a 2-octet AS number. Therefore, the AS number of the peer can be used as remote-as while neighbor interconnection is created between Router A and Router B.

49) Router B can recognize the 4-octet AS number used by the autonomous system of Router C. The AS number of the peer can be used as remote-as while a neighbor is created.

9.4.1.4 Configuration Steps

1. Router A

```
Qtech# conf t
Qtech(config)# router bgp 64496
Qtech(config-router)# neighbor 172.18.1.2 remote-as 64497
Qtech(config-router)# neighbor 172.18.2.3 remote-as 23456
```

2. Router B

```
Qtech# conf t
Qtech(config)# router bgp 64497
Qtech(config-router)# neighbor 172.18.1.1 remote-as 64496
Qtech(config-router)# neighbor 172.18.3.3 remote-as 1.2
```

# Use "bgp asnotation dot" command to change the display format of 4-octet AS number

```
Qtech(config-router)# bgp asnotation dot
Qtech(config-router)# end
Qtech# clear ip bgp *
```

3. Router C

```
Qtech# conft
Qtech(config)# router bgp 1.2
Qtech(config-router)# neighbor 172.18.2.1 remote-as 64496
Qtech(config-router)# neighbor 172.18.3.2 remote-as 64497
```

9.4.1.5 Verification

50) Display the state of neighbor connection on Router A:

```
Qtech# show ip bgp summary
BGP router identifier 172.18.1.1, local AS number 64496
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Stadt
172.18.1.2     4       64497       7       7        1    0    0 00:03:04      0
172.18.2.3     4       23456       4       4        1    0    0 00:00:15      0
```

51) Display the state of neighbor connection on Router B:

```
Qtech# show ip bgp summary
BGP router identifier 172.18.3.2, local AS number 64497
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Stadt
172.18.1.1     4       64496       7       7        1    0    0 00:03:04      0
172.18.3.2     4       65538       4       4        1    0    0 00:01:18      0
```

After executing "bgp notation dot" command, the following information will be displayed:

```
Qtech# show ip bgp summary
BGP router identifier 172.18.3.2, local AS number 64497
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Stadt
172.18.1.1     4       64496       7       7        1    0    0 00:00:04      0
172.18.3.2     4         1.2       4       4        1    0    0 00:00:16      0
```

52) Display the state of neighbor connection on Router C:

```
Qtech# show ip bgp summary
BGP router identifier 172.18.3.3, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Stadt
172.18.2.1     4       64496       7       7        1    0    0 00:00:15      0
172.18.3.2     4       65597       4       4        1    0    0 00:01:19      0
```

10 CONFIGURING THE BGP MDT ADDRESS FAMILY

10.1 About the MDT Address Family

When using PIM-SSM to create Default-MDT during multicast VPN network configuration, configure a BGP MDT address family. Through routing based on the MDT address family, PE can discover other PE addresses and initiate the grating of SPT to other PEs (Configuration steps are detailed in "MD-SCG.doc").

10.2 Default Configurations

No address family is configured.

10.3 Configuring MDT address family

10.3.1 Configuring VRF Instance and Route-related Attributes

Before configuring the MDT address family, VRF instance and route-related attributes must be configured:

Command	Function
Qtech # **configure terminal**	Enters global configuration mode
Qtech(config)# **ip vrf** *VRF*	Creates a VRF named VRF1 and enters VRF mode.
Qtech(config-vrf)# **rd** *rd-value*	Configures VRF RD value, which is identified using XX:XX format, such as RD 1:100. 1 refers to the AS ID of backbone network, while 100 is a numerical value specified by the user.
Qtech(config-vrf)# **route-target** {**both** | **export** | **import**} *rt-value*	Configures route export and import RT attribute of VRF.
Qtech(config-vrf)# {**export** | **import**} **map** *map*	Configures the route map for import and export routes, allowing policy-based filtering of import and export routes.
Qtech(config-vrf)# **mdt default** *group-address*	Configures MDT group address of VRF.
Qtech(config-vrf)# **exit**	Exit VRF mode and enter global configuration mode
Qtech(config)# **interface** *IFNAME*	Enters interface configuration mode
Qtech(config-if)# **ip vrf forwarding** *VRF*	Associates interface with VRF instance
Qtech(config-if)# **ip address** *ip-address mask*	Configures an IP address for the interface
Qtech(config-if)# **end**	Returns to privileged EXEC mode
Qtech # **show running-config**	Verifies the configurations.
Qtech # **write**	(Optional) Saves configurations.

10.3.2 Configuring MDT Address Family

The following part describes the steps of configuring an MDT address family:

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *asn-num*	Creates BGP and enters BGP configuration mode.
Qtech(config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configures BGP session.
Ruiije(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Configures the router to use interface address as the source address when MP-IBGP session is established. Usually, a Loopback interface address is used as the source address.
Ruiije(config-router)# **address-family ipv4** mdt	Enters MDT address family.
Qtech(config-router-af)# **neighbor** *ip-address* **activate**	Activates the route to exchange MDT address family on BGP session.
Qtech(config-router-af)# **neighbor** *ip-address* **next-hop-self**	Changes the next-hop route. This command can be executed on ASBR in OptionB.

10.3.3 Displaying Configurations

The BGP MDT address family can be viewed by executing the **show bgp ipv4 mdt** commands, as listed in the following table:

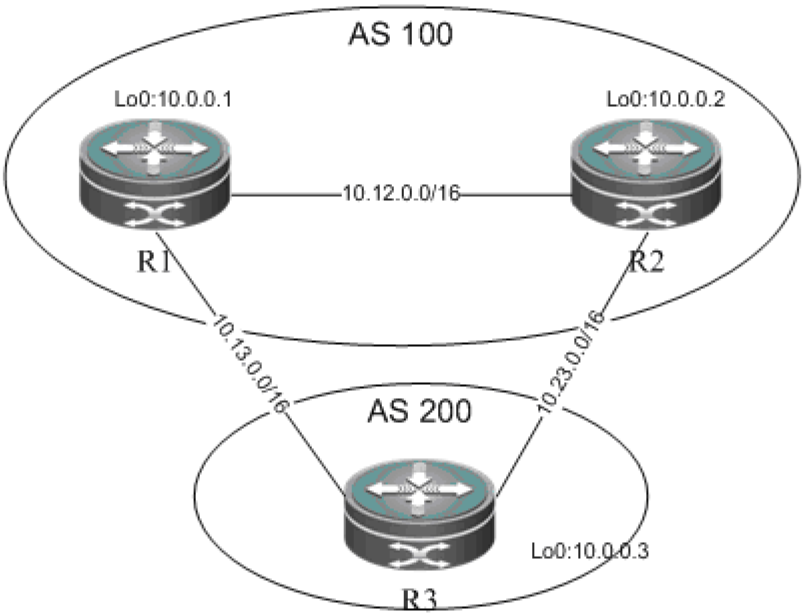
Command	Function
Qtech # **show bgp ipv4 mdt all** [*ip-address* | neighbor [*ip-address*] | **summary**]	Displays all routes under all RDs, a specified route, neighbor information and summary information of the MDT address family.
Qtech # **show bgp ipv4 mdt rd** *rd* [*ip-address*]	Displays all routes or a specified route under a specified RD of the MDT address family.

10.4 Typical Configuration Examples

10.4.1 Networking Requirements

R1 and R2 belong to the same AS. R3 belongs to another AS. Multicast VPN must be established between them, and BGP is used to transmit information about the MDT address family.

10.4.2 Network Topology



R1 and R2 belong to AS100. An IBGP connection is established between R1 and R2 to transmit routes of the MDT address family. R3 belongs to AS200 and establishes EBGP connections with R1 and R2 to transmit the MDT address family.

10.4.3 Configuration Tips

53) Configure VRF instances and associate with interfaces for network isolation;

54) Configure BGP routing protocol to advertise routes of the MDT address family

10.4.4 Configuration Steps

55) Configure VRF instances and associate with interfaces;

- R1

# Create a VRF instance named "VRF1"

```
Qtech# config terminal
Qtech(config)# ip vrf VRF1
Qtech(config-vrf)# rd 100:1
Qtech(config-vrf)# route-target both 123:123
Qtech(config-vrf)# mdt default 232.1.1.1
Qtech(config-vrf)# exit
```

# Associate Gi0/1 with VRF1

```
Qtech(config)# interface gigabitEthernet 0/1
Qtech(config-GigabitEthernet 0/1)# ip vrf forwarding VRF1
Qtech(config-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
Qtech(config-GigabitEthernet 0/1)# exit
R2 and R3 are configured to be the same as R1.
```

56) Configure the BGP routing protocol to advertise routes of the MDT address family

- R1

# Configure the MDT address family

```
Qtech# configure terminal
Qtech(config)# router bgp 100
```

# Configure R2 and R3 as BGP neighbors

```
Qtech(config-router)# neighbor 10.0.0.2 remote-as 100
Qtech(config-router)# neighbor 10.0.0.2 update-source loopback 0
Qtech(config-router)# neighbor 10.13.0.3 remote-as 200
```

# Activate R2 and R3 under the MDT address family

```
Qtech(config-router)# address-family ipv4 mdt
Qtech(config-router-af)# neighobr 10.0.0.2 activate
Qtech(config-router-af)# neighobr 10.13.0.3 activate
```

# Activate R2 and R3 under the VPNv4 address family

```
Qtech(config-router)# address-family vpnv4
Qtech(config-router-af)# neighobr 10.0.0.2 activate
Qtech(config-router-af)# neighobr 10.13.0.3 activate
```

# Bind VRF to BGP

```
Qtech(config-router)# address-family ipv4 vrf VRFi
Qtech(config-router)# exit
R2 and R3 are configured to be the same as R1.
```

10.4.5 Verification

Take the following steps to verify configurations:

57) Verify the state of interfaces bound to VRF. Execute the **show ip vrf interface** to verify interface binding information and state.

```
Qtech#show ip vrf interfaces
Interface             IP-Address      VRF                   Protocol
GigabitEthernet 0/1  10.1.1.1        VRF1                  up
```

58) Ensure MDT routes exist in BGP protocol, as shown below:

```
Qtech#show bgp ipv4 mdt all
BGP table version is 1, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S
Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric  LocPrf      Weight Path
Route Distinguisher: 100:1
*> 10.0.0.1/32    0.0.0.0               0         32768       ?
*>i10.0.0.2/32    10.0.0.2              0         100         ?
*> 10.0.0.3/32    10.13.0.3             0                   200 ?
```

```
Total number of prefixes 3
```

11 CONFIGURING BGP MULTI-PATH LOAD BALANCING

11.1 Understanding BGP Multi-Path Load Balancing

11.1.1 Overview

Multi-path load balancing means data packets are equally forwarded from a number of paths to the same network, and multiple next hops are present in the routing table. Based on the type of equivalent route, BGP multi-path load balancing comes under the following two categories:

- EBGP load balancing: through routes from EBGP neighbors.
- IBGP load balancing: through routes from IBGP neighbors.

Caution The protocol currently does not support load balancing between IBGP and EBGP routes.

Currently, IPv4 and IPv6 protocol stacks support multi-path load balancing, a maximum number of 32 equivalent next hops. The BGP does not limit the number of equivalent routes. This also applies to IBGP and EBGP load balancing.

11.1.2 Working Principle

BGP selects the route with the highest priority from multiple routes that are contained in the BGP routing table and destined to the same network. If several routes have the same priority and are all optimal, BGP will select the only one based on comparison and advertises the route to the forwarding plane for data flow control. When multi-path load balancing is enabled, BGP will list the routes with the same priority as the only optimal route as equivalent routes and advertise the optimal route and its equivalent routes to the forwarding plane for load balancing. Equivalent routes have the same priority and basic attributes.

11.1.3 Protocol Specifications

N/A

11.2 Default Configuration

BGP load balancing is disabled.

11.3 Configuring BGP Multi-Path Load Balancing

11. 3. 1 Configuring EBGP Load Balancing

Command	**Function**
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *as-number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).
Qtech(config-router)# **maximum-paths ebgp** *number*	Sets the number of equivalent routes that support EBGP multi-path load balancing. *number* ranges from 1 to 32.
Qtech(config-router)# **end**	Returns to privilege mode.
Qtech # **write**	(Optional) Saves configuration.

11. 3. 2 Configuring IBGP Load Balancing

Command	**Function**
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *as-number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).

Qtech(config-router)# **maximum-paths ibgp** *number*	Sets the number of equivalent routes that support IBGP multi-path load balancing. *number* ranges from 1 to 32.
Qtech(config-router)# **end**	Returns to privilege mode.
Qtech # **write**	(Optional) Saves configuration.

11.3.3 Configuring AS-PATH Loose Comparison

By default, equivalent routes must have equal AS-PATH. This requirement is too strict in some cases. For load balancing, AS-PATH loose comparison is recommended. Under the AS-APTH loose comparison mode, equivalent routes only need to have equal AS-PATH and AS-PATH in addition to meeting other criteria.

Command	Function
Qtech # **configure terminal**	Enters global configuration mode.
Qtech(config)# **router bgp** *number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).
Qtech(config-router)# **bgp bestpath as-path multipath-relax**	Enables BGP AS-PATH loose comparison.
Qtech(config-router)# **maximum-paths ibgp** *number*	Sets the number of equivalent routes that support IBGP multi-path load balancing. *number* ranges from 1 to 32.
Qtech(config-router)# **end**	Returns to privilege mode.
Qtech # write	(Optional) Saves configuration.

11.3.4 Checking Configuration

Use the **show** command to view information about equivalent routes.

Command	Function
Qtech # **show ip bgp**	Displays BGP routing information.
Qtech # **show ip route**	Checks information in the core routing table.

11.4 Typical Configuration Examples of BGP Multi-Path Load Balancing

11.4.1 Configuring IBGP Non-Equivalent Load Balancing

11.4.1.1 Networking Requirements

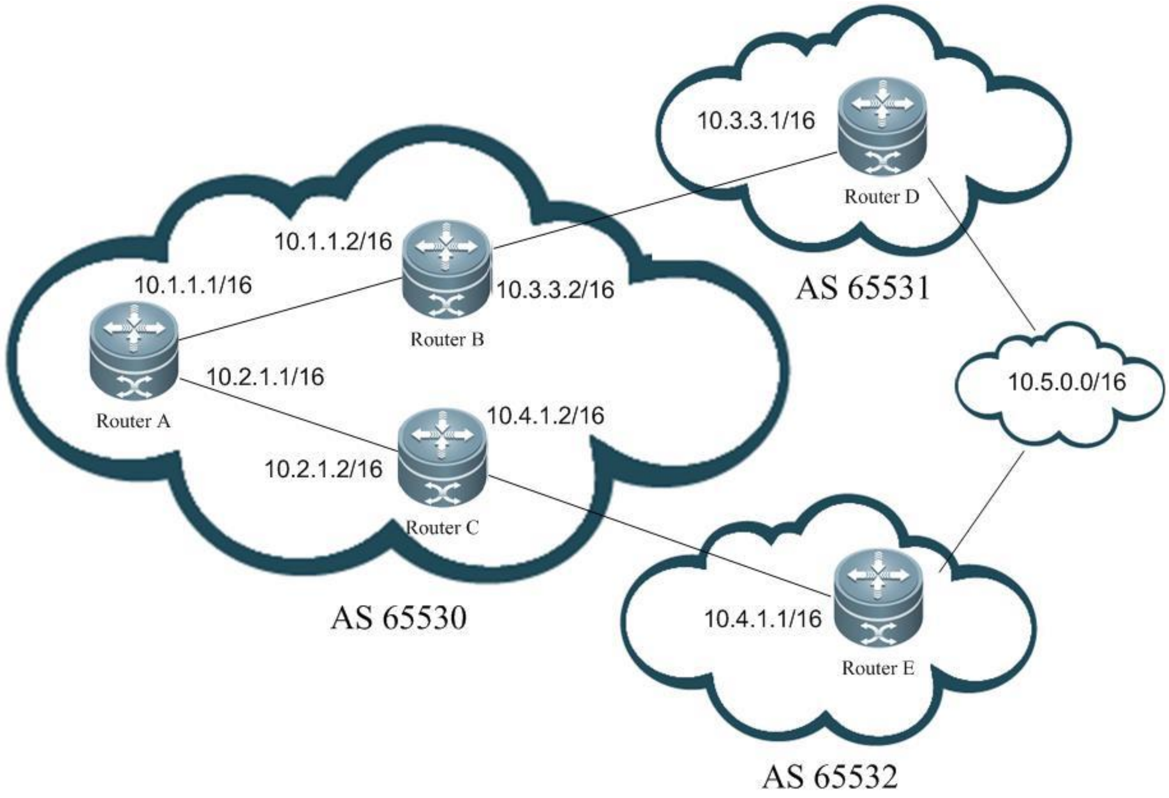
59) Achieves load balancing based on routes learned from IBGP neighbors;

60) Supports BGP AS-PATH loose comparison.

11.4.1.2 Network Topology

As shown in the following figure, Routers A, B and C belong to the same AS numbered 65530. Routers D and E belong to AS 65531 and 65532, which are linked by a BGP connection. AS 65531 and AS 65532 contain the route 10.5.0.0/16 with the same prefix and send the route to AS 65530. Router A learn the route 10.5.0.0/16 from Router B and Router C.

Figure 19 Configuring BGP ECMP



11.4.1.3 Configuration Precautions

61) Enables IBGP load balancing on Router A, and AS-PATH loose comparison.

62) Routers B and D， Routers C and E are connected to EBGP neighbors through a single hop.

63) Routers B and C consider Router A as IBGP neighbors.

11.4.1.4 Configuration Steps

64) Configuration on Router A

```
Qtech# conf t
Qtech(config)# interface fastEthernet 0/0
Qtech(config-if-FastEthernet 0/0)# ip address 10.1.1.1 255.255.0.0
Qtech(config-if-FastEthernet 0/0)# exit
Qtech(config)# interface fastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)# ip address 10.2.1.1 255.255.0.0
Qtech(config-if-FastEthernet 0/1)# exit
Qtech(config)# ip route 10.3.0.0 255.255.0.0 10.1.1.2
Qtech(config)# ip route 10.4.0.0 255.255.0.0 10.2.1.2
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 10.1.1.2 remote-as 65530
Qtech(config-router)# neighbor 10.2.1.2 remote-as 65530
Qtech(config-router)# bgp maximum-paths ibgp 2
Qtech(config-router)# bgp bestpath as-path multipath-relax
```

65) Configuration on Router B

```
Qtech# conf t
Qtech(config)# interface fastEthernet 0/0
Qtech(config-if-FastEthernet 0/0)# ip address 10.1.1.2 255.255.0.0
Qtech(config-if-FastEthernet 0/0)# exit
Qtech(config)# interface fastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)# ip address 10.3.1.2 255.255.0.0
Qtech(config-if-FastEthernet 0/1)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 10.1.1.1 remote-as 65530
Qtech(config-router)# neighbor 10.3.1.1 remote-as 65531
```

66) Configuration on Router C

```
Qtech# conf t
Qtech(config)# interface fastEthernet 0/0
Qtech(config-if-FastEthernet 0/0)# ip address 10.2.1.2 255.255.0.0
Qtech(config-if-FastEthernet 0/0)# exit
Qtech(config)# interface fastEthernet 0/1
Qtech(config-if-FastEthernet 0/1)# ip address 10.4.1.2 255.255.0.0
Qtech(config-if-FastEthernet 0/1)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 10.2.1.1 remote-as 65530
Qtech(config-router)# neighbor 10.4.1.1 remote-as 65532
```

67) Configuration on Router D

```
Qtech# conf t
Qtech(config)# interface fastEthernet 0/0
Qtech(config-if-FastEthernet 0/0)# ip address 10.3.1.1 255.255.0.0
Qtech(config-if-FastEthernet 0/0)# exit
Qtech(config)# interface loopback 1
Qtech(config-if)#ip address 10.5.1.1 255.255.0.0
Qtech(config-if-FastEthernet 0/1)# exit
Qtech(config)# router bgp 65531
Qtech(config-router)# neighbor 10.3.1.2 remote-as 65530
Qtech(config-router)# redistribute connected
```

68) Configuration on Router E

```
Qtech# conf t
Qtech(config)# interface fastEthernet 0/0
Qtech(config-if-FastEthernet 0/0)# ip address 10.4.1.1 255.255.0.0
Qtech(config-if-FastEthernet 0/0)# exit
Qtech(config)# interface loopback 1
Qtech(config-if)#ip address 10.5.1.2 255.255.0.0
Qtech(config-if-FastEthernet 0/1)# exit
Qtech(config)# router bgp 65532
Qtech(config-router)# neighbor 10.4.1.2 remote-as 65530
Qtech(config-router)# redistribute connected
```

11.4.1.5 Checking Configuration

69) Check the status of neighbor connection on Router A:

```
Qtech#show ip bgp summary
BGP router identifier 10.2.1.1, local AS number 65530
BGP table version is 9
2 BGP AS-PATH entries
0 BGP Community entries
3 BGP Prefix entries (Maximum-prefix:4294967295)
Neighbor        V   AS      MsgRcvd   MsgSent   TblVer   InQ   OutQ   Up/Down    
State/PfxRcd
172.16.23.140   4   65530   29        25        8        0     0      00:18:48   2
172.16.23.141   4   65530   24        21        8        0     0      00:17:58   2
Total number of neighbors 2
```

70) Check BGP routes on Router A.

```
Qtech#show ip bgp
BGP table version is 9, local router ID is 10.2.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop   Metric   LocPrf   Weight   Path
*>i10.3.0.0/16    10.3.1.1   0        100      0        65531 ?
*>i10.4.0.0/16    10.4.1.1   0        100      0        65532 ?
* i10.5.0.0/16    10.3.1.1   0        100      0        65531 ?
*>i               10.4.1.1   0        100      0        65532 ?
Total number of prefixes 3
```

71) Check BGP route 10.5.0.0 on Router A:

```
Qtech#show ip bgp 10.5.0.0
BGP routing table entry for 10.5.0.0/16
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  65532
    10.4.1.1 from 10.2.1.2 (172.16.24.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath, best
      Last update: Mon Mar 21 03:45:14 2011
  65531
    10.3.1.1 from 10.1.1.2 (172.16.25.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath
      Last update: Mon Mar 21 03:45:14 2011
```

72) Check routes in Router A’s core routing table:

```
Qtech#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    10.1.0.0/16 is directly connected, FastEthernet 0/0
C    10.1.1.1/32 is local host.
C    10.2.0.0/16 is directly connected, FastEthernet 0/1
C    10.2.1.1/32 is local host.
S    10.3.0.0/16 [1/0] via 10.1.1.2
S    10.4.0.0/16 [1/0] via 10.2.1.2
B    10.5.0.0/16 [200/0] via 10.3.1.1, 00:27:56
                 [200/0] via 10.4.1.1, 00:27:56
```

12 CONFIGURING BGP/MPLS VPN

Please refer to the section "BGP/MPLS L3VPN Configuration" in the "MPLS Configuration Guideline" for details.

13 CONFIGURING BGP/MVPN

Please refer to the section "Multicast VPN Configuration" in the "Multicast VPN Configuration Guideline" for details.

14 CONFIGURING BGP FAST-REROUTE

14.1 BGP Fast-Reroute Overview

14.1.1 IP Fast-Reroute Overview

With high-speed development of IP technologies and application of various complex services, the requirements for network security and stability become increasingly higher. Especially, certain real-time services (audios and videos) are sensitive to network running status and may be largely affected by unstable networks. Therefore, more and more focus and importance are attached to network reliability. With these requirements, the IP Fast-Reroute function comes into being. It is intended to use a backup link to maintain data forwarding during route platform convergence after a faulty link is detected, in order to achieve the ideal targets of "zero delay" and "zero loss" in packet forwarding.

14.1.2 Working Principle of BGP Fast-Reroute

According to the description in RFC 4271, BGP uses a routing policy to select a best route for forwarding local data and advertises the route to neighbors. After the BGP Fast-Reroute function is used, BGP selects a backup route for each best route. After BFD Fast-Reroute detects that the master link is faulty, it switches the data to the originally calculated backup link for forwarding. After route convergence is completed, data is switched to the best route re-calculated for forwarding. In this way, BGP Fast-Reroute can avoid route disconnection due to a link fault before BGP route convergence is completed.

14.2 Default Configuration

BGP Fast-Reroute is disabled by default.

14.3 Configuring BGP Fast-Reroute

Enable BGP Fast-Reroute by using the following steps:

	Command	Description
Step 1	Qtech# **configure terminal**	Enters global configuration mode.
Step 2	Qtech(config)# **router bgp** *as-number*	Enters BGP routing configuration mode.
Step 3	Qtech(config-router)# **address-family ipv4 unicast**	Enters the BGP IPv4 address family.
Step 4	Qtech(config-router-af)# **bgp fast-reroute**	Enables BGP Fast-Reroute.
Step 5	Qtech(config-router-af)#**exit**	Returns to BGP routing configuration mode.
Step 6	Qtech(config-router)#**neighbor** *peer-address* **fall-over bfd**	Configures a BFD session with a BGP neighbor.
Step 7	Qtech(config-router)#**exit**	Returns to global configuration mode.
Step 8	Qtech(config)#**bfd bind bgp peer-ip** *ip-address* **interface** *interface-type interface-index* **source-ip** *ip-address*	(Optional) Configure a BFD session when it is impossible to detect failure of the master link through a BFD session with a BGP neighbor. It is not recommended to use the two commands for configuring the BFD session at the same time.
Step 9	Qtech(config)# **end**	Returns to privileged EXEC mode.
Step 10	Qtech # **show running-config**	Verifies the configuration.
Step 11	Qtech # **write**	(Optional) Saves the configuration.

😐 **Product Compatibility**	All products that support BGP and IP Fast-Reroute support the above configuration commands.

⚠
Caution

BGP Fast-Reroute is subject to the following constraints:
1) BGP Fast-Reroute is supported only in the IPv4 Unicast and IPv4 VRF address families of BGP.
2) Only one backup route can be generated and the next hop of the backup route cannot be the same as that of the preferred route.
3) A backup next hop cannot be generated for an ECMP route.
4) In BGP IPv4 VRF configuration mode, BGP Fast-Reroute has a lower priority than VPN Fast-Reroute. That is, if VPN Fast-Reroute is enabled in VRF mode, BGP Fast-Reroute takes effect only when VPN Fast-Reroute fails to calculate a backup route.

14.4 Configuration Example

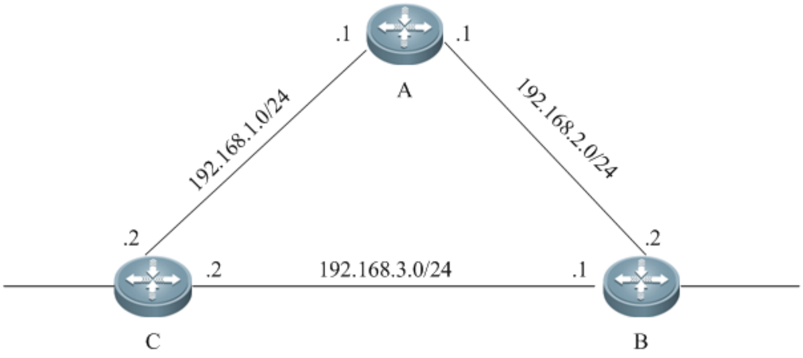
14.4.1 Configuring EBGP Fast-Reroute

14.4.1.1 Networking Requirements

As shown in the following figure, the three routers A, B and C belong to different autonomous areas (ASs). They are interconnected to each other through BGP. It is required that when the link between B and C is faulty, traffic can be fast switched to the link between C and A, and the traffic path is C→A→B.

14.4.1.2 Network Topology

Figure 14-1 EBGP Fast-Reroute



14.4.1.3 Configuration Steps

73) Configure router A.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.2.1 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 100
Qtech(config-router)# neighbor 192.168.1.2 remote-as 300
Qtech(config-router)# neighbor 192.168.2.2 remote-as 200
Qtech(config-router)# redistribute connect
```

74) Configure router B.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.3.1 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.2.2 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 200
Qtech(config-router)# neighbor 192.168.3.2 remote-as 300
Qtech(config-router)# neighbor 192.168.3.2 fall-over bfd
Qtech(config-router)# neighbor 192.168.2.1 remote-as 100
Qtech(config-router)# redistribute connect
```

75) Configure router C.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 192.168.1.2 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface fastEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 192.168.3.2 255.255.0.0
Qtech(config-if-GigabitEthernet 0/2)# bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 300
Qtech(config-router)# neighbor 192.168.1.1 remote-as 100
Qtech(config-router)# neighbor 192.168.3.1 remote-as 200
Qtech(config-router)# neighbor 192.168.3.1 fall-over bfd
Qtech(config-router)# address-family ipv4 unicast
Qtech(config-router-af)# bgp fast-reroute
Qtech(config-router-af)# redistribute connect
```

14.4.1.4 Verification

76) Display the neighbor connection status on router C.

```
Qtech# show ip bgp summary
BGP router identifier 10.10.10.10, local AS number 300
BGP table version is 12
4 BGP AS-PATH entries
0 BGP Community entries
3 BGP Prefix entries (Maximum-prefix:4294967295)

Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.1.1     4         100      76      77       12    12    0 00:59:27        3
192.168.3.1     4         200      30      30       12    12    0 00:19:03        3

Total number of neighbors 2
```

77) Display the BGP route on router C.

```
Qtech# show ip bgp
BGP table version is 12, local router ID is 10.10.10.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale, b - backup entry
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric      LocPrf      Weight Path
*  192.168.1.0      192.168.3.1              0                       0 200 ?
*                   192.168.1.1              0                       0 100 ?
*>                  0.0.0.0                  0                   32768     ?
*> 192.168.2.0      192.168.3.1              0                       0 200 ?
*b                  192.168.1.1              0                       0 100 ?
*  192.168.3.0      192.168.3.1              0                       0 200 ?
*                   192.168.1.1              0                       0 100 200 ?
*>                  0.0.0.0                  0                   32768     ?

Total number of prefixes 3
```

78) Display the BGP route 192.168.2.0 on router C.

```
Qtech# show ip bgp 192.168.2.0
BGP routing table entry for 192.168.2.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  192.168.1.1
  200
    192.168.3.1 from 192.168.3.1 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Last update: Tue Oct  5 00:26:52 1971

  100
    192.168.1.1 from 192.168.1.1 (44.44.44.44)
      Origin incomplete, metric 0, localpref 100, valid, external, backup
      Last update: Mon Oct  4 23:46:28 1971
```

79) Display the route in the core routing table on router C.

```
Qtech# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, GigabitEthernet 1/9
C    192.168.1.2/32 is local host.
B    192.168.2.0/24 [20/0] via 192.168.3.1, 00:21:39
C    192.168.3.0/24 is directly connected, GigabitEthernet 1/11
C    192.168.3.2/32 is local host.
```

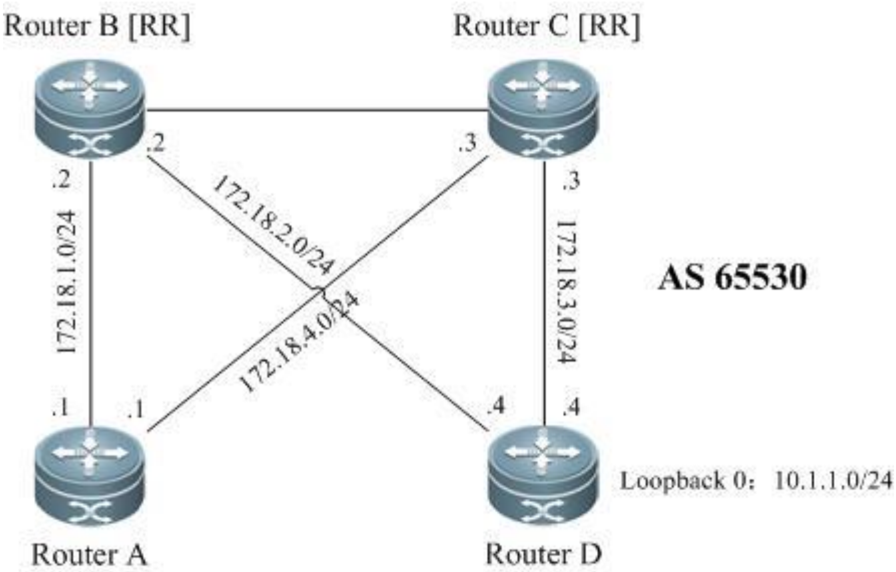
14.4.2 Configuring IBGP Fast-Reroute

14.4.2.1 Networking Requirements

As shown in the following figure, the router routers A, B, C and D belong to the same AS. B and C serve as the route reflectors and establish IBGP neighbors with A and D respectively. The route 10.1.1.0/24 advertised by D will be reflected to A through B and C at the same time. Normally, A prefers the link A->B->D to reach the network segment 10.1.1.0/24. It is required that when the link A->B->D is faulty, traffic can be fast switched to the link A->C->D.

14.4.2.2 Network Topology

Figure 14-2 IBGP Fast-Reroute



14.4.2.3 Configuration Steps

80) Configure router A.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 172.18.1.1 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 172.18.4.1 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 172.18.1.2 remote-as 65530
Qtech(config-router)# neighbor 172.18.4.3 remote-as 65530
Qtech(config-router)# bgp fast-reroute
Qtech(config-router)# exit
Qtech(config)# bfd bind bgp peer-ip 172.18.2.4 interface GigabitEthernet 0/1 source-ip
172.18.1.1
```

81) Configure router B.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 172.18.1.2 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 172.18.2.2 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 172.18.1.1 remote-as 65530
Qtech(config-router)# neighbor 172.18.2.4 remote-as 65530
Qtech(config-router)# address-family ipv4 unicast
Qtech(config-router-af)# neighbor 172.18.1.1 route-reflector-client
Qtech(config-router-af)# neighbor 172.18.2.4 route-reflector-client
Qtech(config-router-af)# end
```

82) Configure router C.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 172.18.4.3 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 172.18.3.3 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 172.18.4.1 remote-as 65530
Qtech(config-router)# neighbor 172.18.3.4 remote-as 65530
Qtech(config-router)# address-family ipv4 unicast
Qtech(config-router-af)# neighbor 172.18.4.1 route-reflector-client
Qtech(config-router-af)# neighbor 172.18.3.4 route-reflector-client
Qtech(config-router-af)# end
```

83) Configure router D.

```
Qtech# conf t
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if-GigabitEthernet 0/1)# ip address 172.18.2.4 255.255.255.0
Qtech(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Qtech(config-if-GigabitEthernet 0/1)# exit
Qtech(config)# interface GigabitEthernet 0/2
Qtech(config-if-GigabitEthernet 0/2)# ip address 172.18.3.4 255.255.255.0
Qtech(config-if-GigabitEthernet 0/2)# exit
Qtech(config)# interface loopback 0
Qtech(config-if-Loopback 0)# ip address 10.1.1.1 255.255.255.0
Qtech(config-if-Loopback 0)# exit
Qtech(config)# router bgp 65530
Qtech(config-router)# neighbor 172.18.2.2 remote-as 65530
Qtech(config-router)# neighbor 172.18.3.3 remote-as 65530
Qtech(config-router)# network 10.1.1.0 mask 255.255.255.0
Qtech(config-router)# exit
Qtech(config)# bfd bind bgp peer-ip 172.18.1.1 interface GigabitEthernet 0/1 source-ip
172.18.2.4
```

14.4.2.4 Verification

84) Display the neighbor connection status on router A.

```
Qtech# show ip bgp summary
BGP router identifier 10.1.1.2, local AS number 300
BGP table version is 12
0 BGP AS-PATH entries
0 BGP Community entries
1 BGP Prefix entries (Maximum-prefix:4294967295)

Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.18.1.2      4       65530      76      77       12    12    0 00:59:27        1
172.18.4.3      4       65530      30      30       12    12    0 00:19:03        1

Total number of neighbors 2
```

85) Display the BGP route on router A.

```
Qtech# show ip bgp
BGP table version is 12, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale, b - backup entry
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric      LocPrf      Weight Path
*>i10.1.1.0         172.18.2.4               0         100           0     i
*bi                 172.18.3.4               0         100           0     i

Total number of prefixes 3
```

86) Display the BGP route 10.1.1.0 on router A.

```
Qtech# show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    172.18.2.4 (metric 10) from 172.18.1.2 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Last update: Tue Oct  5 00:26:52 1971

  Local
    172.18.3.4 (metric 20) from 172.18.4.3 (44.44.44.44)
      Origin incomplete, metric 0, localpref 100, valid, external, backup
      Last update: Mon Oct  4 23:46:28 1971
```

87) Display the route in the core routing table on router A.

```
Qtech# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    172.18.1.0/24 is directly connected, GigabitEthernet 1/1
C    172.18.1.1/32 is local host.
C    172.18.4.0/24 is directly connected, GigabitEthernet 1/2
C    172.18.4.1/32 is local host.
B    10.1.1.0/24 [200/0] via 172.18.2.4, 00:21:39
```

15 CONFIGURING IS-IS

15.1 Understanding IS-IS Protocol

15.1.1 Overview

IS-IS (Intermediate System-to-Intermediate System) is a routing protocol defined in ISO10589. It was initially a dynamic routing protocol designed by ISO for CLNP (Connectionless Network Protocol). With IP getting more and more popular, IETF enables IS-IS to support IP in RFC1195 and develops IS-IS into Integrated IS-IS. After years of development, Integrated IS-IS has become a scalable, robust and easy-to-use IGP protocol, which is applicable to IP and ISO CLNS based dual-environment network.

As a link-state protocol, IS-IS has certain features shared by link-state protocols. It discovers and maintains adjacencies by sending Hello packets, and advertises its own link state by sending LSP (Link State PDU) to its neighbors. IS-IS has a two-level hierarchy (level 1 and level 2 routing), with all devices at the same level having the same LSDB, which stores the LSP generated by all devices at the same level. In this way, all devices at the same level are aware of the network topology of their level, and each device uses Dijkstra SPF algorithm to optimize route calculation, select path and achieve fast convergence.

15.1.2 Hierarchical Structure of IS-IS Network

Figure 1

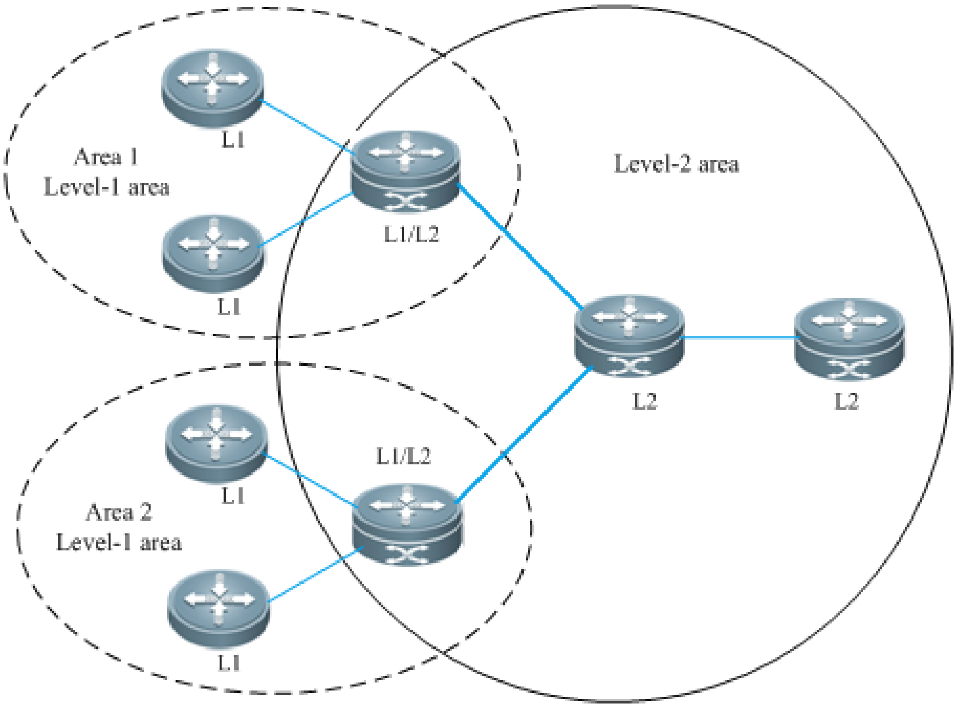


Figure 1 IS-IS hierarchy

This network is divided into Level-1 and Level-2. All nodes for exchanging information among devices in the same area form Level-1. All intra-area devices are aware of the network topology of entire area and carry out Inter-area data exchange. Level-1-2 devices are the boundary devices to connect different areas. Inter-area connection is achieved by connecting Level-2 devices, while the boundary devices of respective areas jointly form a backbone network (Level-2). Inter-area data exchange is carried out at level-2.

Level-1 devices only concern about the topology structure of the local area, including all nodes and next-hop devices reaching these nodes in the local area. Level-1 device accesses other areas through the Level-2 device, and forwards data packets in the destination network outside the area to the closest Level-2 device.

15.1.3 Address Encoding of IS-IS Protocol

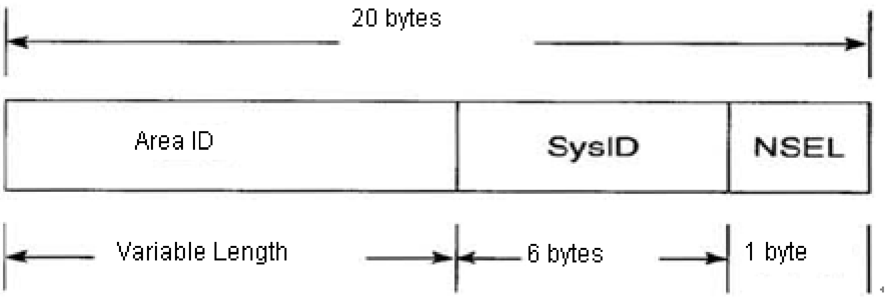


Figure 2 NET address format

IS-IS protocol address is called NET, which can be divided into three parts: Area address, System ID and NSAP selector. The total length of NSAP address ranges from 8 to 20 bytes.

The length reserved for area address is variable. The area ID is the length of route domain, and is fixed in the route domain. The length of area address ranges from 1 to 13 bytes.

The length of System ID is 6 bytes, and is unique in the autonomous system.

NSAP is the network selector, and is sometimes called SEL, with length being 1 byte. In IS-IS, SEL is usually set to 00 to represent the routing device.

15.1.4 IS-IS Packet Types

There are three types of packets:

- Link-state PDUs (LSP)
- IS-IS Hello PDUs

Sequence number PDUs (SNP)
Link-state PDUs (LSP) are used to advertise link-state logs within the area. They can be divided into: Level 1 Link State PDU and Level 2 Link State PDU. LSP will only be flooded at its own level.

IS-IS Hello PDUs (IIH PDU) are used to maintain adjacencies. Hello PDUs will send multicast MAC address to detect whether IS-IS is operated in other systems.

Sequence number PDUs (SNP) can be divided into CSNP and PSNP.

Complete sequence number PDU (CSNP) is used to synchronize LSDB. In a broadcast network, DIS will send CSNP packets once every 10 seconds by default. In a point-to-point network, CSNP packets will only be sent once after adjacency is formed.

Partial sequence number PDU (PSNP) is also used to synchronize LSDB.

15.1.5 DIS

DIS: Designated IS, the designated routing device in on the broadcast network, equivalent to the DR in OSPF.

Pseudonode: The pseudonode is generated by DIS and establishes contacts with all devices in the network.

DIS will model the multi-access link as a pseudonode to create pseudonode LSPs. All routing devices on the local network contact with the pseudonode, and no direct contact between them is allowed. The broadcast subnet and NBMA network are regarded as a pseudonode externally. All non-DIS devices on the network will report their link state to the DIS, which will report the link state on behalf of all ISs on the entire network. The reason to elect DIS is the same as the reason to elect DR in OSPF: to reduce unnecessary adjacencies and routing information exchange.

DIS is created through election. The DIS election in IS-IS is pre-emptive, which is different from DR in OSPF.

The result of DIS election can be controlled by configuring the "Priority" of interface. The one with the highest "Priority" value will be elected.

15.1.6 TLVs Supported by IS-IS

Currently, Qtech IS-IS supports the following TLV codes:

TLV CODE	Description
Code=1	Area addresses
Code=2	Priority level information of IS neighbor
Code =3	ES neighbors
Code=6	MAC address of IS neighbor
Code=8	Padding
Code=9	LSP entries
Code=10	Authentication information
Code=14	Buffer size of source LSP
Code=22	Extended IS reachability
Code=128	IP internal reachability information
Code=129	Protocols supported
Code =130	IP external reachability information
Code=131	IDRP information
Code=132	IP interface address
Code=133	Authentication information
Code=135	Extended IP reachability TLV
Code=137	Dynamic host name
Code = 211	Graceful Restart
Code=232	IPV6 interface
Code =236	IPV6 IP Reachability TLV
Code =240	P2P 3-way handshake TLV

15.1.7 LSP Fragments Extension

IS-IS informs devices of link-state information by flooding LSP packets. The size of LSP packets is restricted by link MTU and cannot be extended. When the information to be informed exceeds the size of a LSP packet, IS-IS will create LSP fragments to carry new link-state information. According to the ISO standard, the LSP fragment is recognized by the 1-byte LSP Number. Therefore, the maximal number of LSP fragments produced by an IS-IS node is 256.

There are several reasons causing 256 fragments not enough:

- New TLV or Sub-TLV extended by new application, such as TE
- Constant expansion of network scale
- Informing routes with smaller chip or redistributing other routes to IS-IS.

When LSP fragments are filled, subsequent routing information and neighbor information will be discarded directly. There will be network anomaly, such as routing blackhole or routing loops. LSP fragments need to be extended to carry more link-state information to ensure normal operation of the network.

Definitions of fragments extension are listed as follows:

- Normal system-id: It refers to the current system ID defined by ISO, which is used to form adjacency and learn routes. “Nornal” differentiates this kind of system-id from additional system-id produced by fragments extension.
- Additional system-id: It is configured by the administrator and used to extend LAP, in comparison with normal system-id. Additional system-id does not appear in Hello packets for adjacency formation. Except that, additional system-id adopts the same rules as normal system-id, for example, it must be unique and cannot be repeated in the whole intra-area.
- Originating System: It refers to the routing device running the IS-IS protocol. It is in comparison with the virtual system identified by the additional system ID.
- Virtual System/Vritual IS: It refers to the system identified by additional system-id, which is used to generate extended LSP. RFC proposes this notion and differentiate it from the originating system. Every virtual system generates up to 256 LSP fragment packets. The administrator can configure several additional system IDs , which represent virtual systems, to generate more LSP fragment packets to meet the demand.
- Original LSP: The LSP packet is generated by the originating system. The system-id is normal system-id.
- Extended LSP: The LSP packet is generated by the virtual system. The system-id is additional system-id.

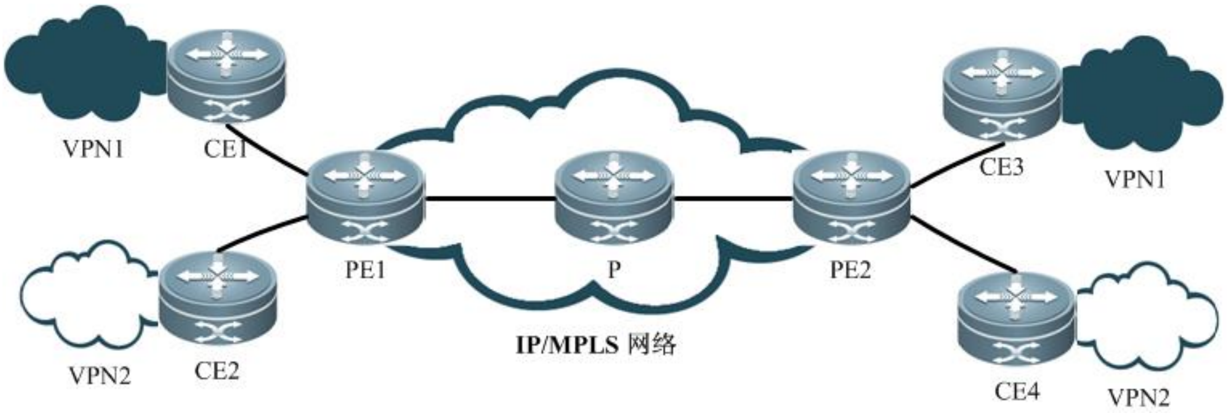
IS-IS can inform devices of more link-state information with extended LSP by setting additional system-id and enabling fragments extension. Every virtual system can be regarded as a virtual routing device which establishes adjacency with the originating system. The metric between them is 0. Extended LSP is the LSP packet released by the neighbor of originating system, namely the virtual system,

15.1.8 IS-IS VRF

VRF is short for VPN Routing and Forwading. It is mainly used to perform local routing, segregate data packets and address routing conflicts caused by VPNs using the same prefix. Most IPv4 and IPv6 VPNs are MPLS VPN. Combined with MPLS’s advantage in service quality and security guarantee, MPLS VPN has become the preferred solution to enable interconnection among branches of enterprises and industries in different areas.

The following figure is a typical VRF networking application, which is to enable VPN segregation control by configuring VRF on PE devices.

Figure 1-3 Enabling VPN segregation control by configuring VRF on PE devices



As figure 1-3 shows, two site users (CE1 and CE3) under VPN1 should be able to visit each other. Two site users (CE2 and CE4) under VPN2 should be able to visit each other. VPN1 and VPN2 should not be able to visit each other for two reasons:

- The two VPNs belong to different users or departments. Mutual visit is prevented for security’s sake.
- There may be the same IP address on VPN1 and VPN2.

CE is used to connect the user network to PE and exchange VPN routing information with PE: release local routes to PE and learn remote site routes from PE.

PE is used to learn routing from directly connected CE and exchange learned VPN routes with other PEs through BGP. The PE device is responsible for the access of VPN business.

The device P is a device that is not directly connected with CE on the operator network. The device is only required to support MPLS forwarding and cannot sense VPN.

The IS-IS routing protocol runs between PE and CE to enable VRF-based routes learning. PE and CE only learn routes within the same VPN to enable VPNs segregation control.

15.1.9 IS-IS Definitions

- ES: End System refers to non-router devices, such as host.
- IS: Intermediate System refers to router devices, the basic unit sending routing information and generating routes in the IS-IS protocol.
- ES-IS: End System-to-Intermediate System, an OSI protocol that defines how end systems (ES) and intermediate systems (IS) learn about each other.
- Domain: routing domain. In one routing domain, a group of ISs will exchange routing information through the same routing protocol.
- Area: a routing sub-domain. One routing domain can be divided into multiple areas.
- CSNP: Complete sequence number PDU, sent by DIS every 10 seconds on the broadcast network to synchronize link state.
- PSNP: Partial sequence number PDU, sent on the point-to-point link to acknowledge receipt of an LSP or on the broadcast network to request an LSP.
- ENPA: attached subnet point that provides subnet services.
- CLNP: Connectionless Network Protocol, an IP-alike OSI protocol to transmit data and error messages in the network layer.
- CNLS: Connectionless Network Service is the solution to unreliable connection which doesn't need the circuit to be established before data transmission.
- DIS: Designated Intermediate System, which is similar to the DR in OSPF. It floods LSPs to other devices on the LAN. Unlike OSPF, DIS forms adjacencies with other devices which also form adjacencies between each other.
- Hello: This packet is used to establish and maintain adjacencies.
- LSP: Link-state PDU, which is similar to the LSA in OSPF, but LSP doesn't rely on TCP/IP protocol information. There are different LSPs for different routes, such as L1 LSP and L2 LSP.
- NSEL: NSAP selector, sometimes also called SEL. It identifies a network service user, and is similar to the TCP/UDP port for upper-layer service in the IP protocol. In IS-IS, SEL is usually set to 00 to imply the routing device.
- NSAP: Network Service Access Point is the complete address for CLNS packets, including OSI address and high-level process, with structure containing area ID, System ID and SEL. A NSAP address with SEL being 00 implies the NET entity. It is similar to the combination of IP address and IP protocol number.
- SNPA: Sub-network Point of Attachment provides physical link and network layer services, and is similar to the MAC address in IP and DLCI, WAN and HDLC in FR.
- L1 router: The router inside an area. It only accepts relevant information from the local area. In order to reach other areas, a default route to the closest L2 must be saved in L1.
- L2 router: the trunk router between different areas. L1 cannot be directly connected with L2.
- L1/L2 router: The boundary router used to connected L1 router and L2 router, containing the databases of both L1 router and L2 routers. It is similar to the ABR in OSPF.
- Pseudonode: The identifier of broadcast subnet of LAN. Pseudonode makes broadcast media a virtual routing device, while every router acts as its interface. DIS manages the adjacency between router and pseudonode.
- NET: network entity title, a part of OSI address describing the area and system ID.
- Circuit: Circuit is the term for interface in IS-IS. NSAP and NET represent the entire device, while circuit represents the interface. For a point-to-point interface, the circuit ID is 1 byte long. For example, the circuit ID is 0x00 in HDLC; in a broadcast network such as LAN, the circuit ID is generally 7 bytes long combining the System ID, such as 1921.6800.0001.01.

To learn more details about IS-IS, please refer to ISO 10589 and RFC 1195.

15.2 Configuring IS-IS

The default configurations of IS-IS are given below:

Function	Default setting
Network interface	Interface metric: 10 Advertised Hello interval: 10 seconds Advertised CSNP interval: 10 seconds Minimal interval for LSP transmission: 33 milliseconds LSP retransmission interval: 5 seconds Hello multiplier number: 3 Priority for routing node election: 64 Circuit type: Level-1-2 Authentication password: none
System type	Level-1-2

Default route	Level-1 default route: enabled Level-2 default route: disabled.
LSP authentication password	NA
IS-IS GR	IS-IS GR Restarter: Disabled IS-IS GR Helper: Enabled
Summary-address	Undefined
Overload flag bit	Undefined
LSP checksum error	Enabled
Adjacency change logging	Disabled
LSP refresh interval	900 seconds
LSP lifetime	1200 seconds

15.2.1 Enabling IS-IS

Unlike other routing protocols, you need to first create an IS-IS routing process and specify the interfaces on which IS-IS shall be enabled.

15.2.1.1 Creating IS-IS Routing Process

Command	Function
Qtech(config)# **router isis** [*tag*]	Starts IS-IS routing process, with tag being the name of IS-IS process.
Qtech(config-router)# net areaAddress. SystemId.00	Configures the NET address of IS-IS.

- Create IS-IS route process:
- To run IS-IS routing protocol, first create IS-IS routing process in global configuration mode; you can also add "Tag" behind "router isis". This Tag refers to the name of IS-IS routing process. You can also choose not to configure the name of IS-IS routing process. You can configure different IS-IS routing processes by adding different Tags.
- Configure IS-IS protocol's system ID and area address:
- System ID is the only identifier of IS in an autonomous system. Therefore, System ID must be unique in the entire autonomous system. In IS-IS, each area can have one or multiple area addresses, and generally only one area address is needed. Area repartition can be done by configuring multiple area addresses. When configuring multiple area addresses for one IS, the System ID must be identical.

Example:

```
Qtech(config-router)# net 49.0001.0000.0000.0001.00
```

In the above configuration command, the area address is 49.0001 and System ID is 0000.0000.0001.Dots in the numbers are for your convenience only.

Caution

Level-1 IS routing nodes in the same area must be configured with the same area address. Currently, the core routing table will not be sensitive to the IS-IS process generating the routing table.

Caution

By default, CPU protection is enabled on the switch. For packets corresponding to each destination group address of IS-IS (AllISSystems, AllL1ISSystems, AllL2ISSystems), there will be default limit in number when sent to the CPU. For example, the default limit is 400pps. If there are many adjacencies or if Hello packets are sent at short intervals, the IS-IS packets received by the switch may exceed the default limit, leading to the continual oscillation of adjacencies. In such a case, the limit for IS-IS packets must be raised by configuring such global commands of **cpu-protect type isis-is pps, cpu-protect type isis-l1is pps and cpu-protect type isis-l2is pps**.

15.2.1.2 Configuring IS-IS Protocol on the Interface

After global IS-IS protocol is enabled, you need to configure IS-IS protocol on the interface.

Use the following command to configure IS-IS protocol on the interface.

Command	**Function**
Qtech(config-if)# ip router isis [tag]	Enables IPv4 IS-IS on the specified interface, with "tag" being the name of IS-IS process.
Qtech(config-if)# **ipv6 router isis** [*tag*]	Enables IPv6 IS-IS on the specified interface, with "tag" being the name of IS-IS process.

⚠ Caution

When configuring IP address, the IP address must be in the same network segment as the IP address of adjacent interface.

⚠ Caution

If the IP address is not in the same network segment as the IP address of adjacent interface, the adjacency cannot be established.

⚠ Caution

If the interface needs to join the specified IS-IS process, the Tag name of this IS-IS process must be added after "ip router isis".

⚠ Caution

By configuring the no ip routing command in global configuration mode, IS-IS will disable IPv4 routing function on all interfaces, namely "no ip router isis [tag]" will be executed automatically on all interfaces, while other IS-IS configurations will remain unchanged.

⚠ Caution

When you configure the IPv6 address, the local link address will be configured by default

⚠ Caution

If the IPv6 address or the IPv6 address of the adjacent interface does not have the local link address, the adjacency cannot be established.

⚠ Caution

If you want to add the interface to the designated IS-IS process, attach the Tag of this IS-IS process to the end of **ipv6 router isis.**

⚠ Caution

If the **no ipv6 unicast-routing** command is executed in global configuration mode, IS-IS will disable IPv6 routing on all interfaces. Namely, the **no ipv6 router isis** [*tag*] command is executed automatically on all interfaces while other IS-IS configurations remain unchanged.

⚠ Caution

In order to avoid routing blackholes on the network where IPv4 and IPv6 coexist, if protocols supported by two devices or interfaces are not the same, adjacency will not be set up. In this case, please check whether the network topology has any problem. If there is no problem with the network topology and there are no routing blackholes, configure different instances to perform IPv4 and IPv6 routes learning.

15.2.2 Configuring IS-IS Hello Packets

15.2.2.1 Configuring the Advertised Hello Interval

IS-IS will periodically send Hello packets on the interface, while routing deviceswill discover and maintain adjacencies through the reception and sending of Hello packets. Complete the following configuration in interface configuration mode to set the Hello packet broadcast interval:

Command	Function
Qtech(config-if)#**isis hello-interval** { **interval** | **minimal** } [**level-1** | **level-2**]	Configures the interval for sending Hello packets on the interface, in the range of 1 to 65535 seconds.

Use the command to change the interval for sending Hello packets. DIS in broadcast network will send Hello packets at an interval which is three times shorter than non-DIS. If IS is elected as DIS on this interface, the interface will send Hello packets every 3.3 seconds by default.

If the key word minimal is used, then the holdtime in Hello packets will be set to 1, and hello interval will be calculated based on the hello-multiplier. For example, if hello-multiplier is configured to 3 and the isis hello-interval minimal command is configured at the same time, the value of hello-interval shall be 1s/3 (333ms).

⚠ Caution

By default, CPU protection is enabled on the switch. For packets corresponding to each destination group address of IS-IS (AllISSystems, AllL1ISSystems, AllL2ISSystems), there will be default limit in number when sent to the CPU. For example, the default limit is 400pps. If there are many adjacencies or if Hello packets are sent at short intervals, the IS-IS packets received by the switch may exceed the default limit, leading to the continual oscillation of adjacencies. In such a case, the limit for IS-IS packets must be raised by configuring such global commands of cpu-protect type isis-is pps, cpu-protect type isis-l1is pps and cpu-protect type isis-l2is pps.

15.2.2.2 Configuring Hello Multiplier Number

IS-IS will periodically send the Hello packet on the interface, and advertise the adjacency hold time of the IS device in the header of the Hello packet. Neighbors will update adjacencies based on the holdtime field in the Hello packet header. The holdtime value in the Hello packet header equals to the hello-interval value multiplies the hello-multiplier value.

Complete the following configuration in interface configuration mode to set the Hello packet holdtime multiplier:

Command	Function
Qtech(config-if)# **isis hello-multiplier multiplier-number** [**level-1** | **level-2**]	Configures Hello multiplier number on the interface.

Use the command to change the holdtime multiplier of Hello packets and the holdtime. The holdtime of Hello packets can also be changed by changing hello-interval or changing both of them.

15.2.2.3 Configuring Hello packet failure number

IS-IS protocol maintains relationships with adjacent routers by sending and receiving Hello packets. When the local router fails to receive a specific number of Hello packets from peers continuously, adjacent routers will be considered failed. The default number is three. Complete the following configuration in interface configuration mode to set the failure number of Hello packets:

Command	Function
Qtech(config-if)# **isis hello-multiplier** *multiplier-number* [**level-1** | **level-2**]	Sets the number of failure Hello packets on the interface.

15.2.3 Configuring IS-IS LSP

15.2.3.1 Configuring LSP Minimal Transmission Interval

Complete the following configuration to set the minimum interval for sending LSP packets continuously by IS-IS on the interface:

Command	Function
Qtech(config-if)# **isis lsp-interval** *interval*	Configures the minimal interval (1-4294967295 milliseconds) for sending LSPs on the interface.

15.2.3.2 Configuring LSP Retransmission Interval

On a point-to-point link, if the local router fails to receive any reply after sending LSPs for a while, it will assume that the LSPs sent formerly are lost or discarded. To ensure the reliability the LSP sending, the local routing device will retransmit the same LSPs. Complete the following configuration in interface configuration mode to set the packets retransmission interval:

Command	Function
Qtech(config-if)# **isis retransmit-interval** *interval*	Configures the interval (1-65535 seconds) for retransmit LSPs on the point-to-point link.

15.2.3.3 Configuring LSP Refresh Interval

To ensure that each network node can maintain the latest LSP, LSP will periodically refresh the current LSP, and such interval is called LSP refresh interval. With this mechanism, LSPs can remain synchronized in the entire area. Complete the following configuration in IS-IS protocol configuration mode to set the LSP refreshing frequency:

Command	Function
Qtech(config-router)# **lsp-refresh-interval** *interval*	Configures LSP refresh interval (1-65535 seconds).

⚠ Caution The lsp-refresh-interval shall be less than the max-lsp-lifetime.

15.2.3.4 Configuring LSP Lifetime

In LSP, there is a field value called LSP lifetime. When the routing device generates LSP, it will set the maximal lifetime in the field for this LSP. When the LSP is received by another routing decvice, the lifetime will decrease gradually, and the old LSP will be replaced if a new LSP is received. If no refreshed LSP is received and the LSP lifetime has decreased to 0, it will still be kept in the link-state database for 60 seconds. If no refreshed LSP is received within the 60 seconds, this LSP will be deleted from LSDB. With this mechanism, LSPs can remain synchronized in the entire area. Complete the following configuration in IS-IS protocol configuration mode to set the lifetime of LSPs generated by the router:

Command	Function
Qtech(config-router)#**max-lsp-lifetime value**	Configures LSP lifetime (1-65535 seconds).

⚠ Caution The max-lsp-lifetime must be greater than lsp-refresh-interval.

15.2.3.5 Configuring LSP Fragments Expansion

The LSP fragments expansion function is enabled by setting additional system ID and enabling fragments expansion . Execute the following commands in IS-IS routing process configuration mode:

Command	Function
Qtech(config-router)# **lsp-fragment-extend** [**level-1 | level-2**] [**compatible rfc3786**]	Enables fragments extension.
Qtech(config-router)#**virtual-system** *system-id*	Sets additional system ID.

15.2.4 Configuring IS-IS SNP

15.2.4.1 Configuring the Advertised CSNP Interval

Complete Sequence Number PDUs (CSNP) are packets sent by DIS in the broadcast network to maintain link-state database synchronization. CSNPs are also periodic broadcast packets. Complete the following configuration in interface configuration mode to set the CSNP packet broadcasting interval:

Command	Function
Qtech(config-if)# **isis csnp-interval** *interval* [**level-1** | **level-2**]	Configures the interval (0-65535 seconds) for sending CSNP packets on the interface.

Configure this command to change the interval for sending CSNP packets. By default, DIS on the broadcast network will send CSNP packets every 10 seconds.

For P2P interface network, CSNP packets will only be sent when adjacency is just established; if the interface is set mesh-groups, you can set sending CSNP packets periodically.

If csnp-interval is set to 0, no CSNP will be sent.

If **mesh-group** is required on the IS-IS interface, you need to configure a non-zero interval for sending VSNP packets to synchronize LSP by using the **isis csnp-interval** command so as to ensure complete LSP synchronization among adjacencies in the network.

15.2.5 Configuring IS-IS Level Type

IS-IS protocol supports two-level hierarchy, so as to manage route selection and achieve expandable route selection. Every level only maintains the topology structure of local area.

You can execute "is-type" command in IS-IS router configuration mode to configure IS-IS Level, or execute "circuit-type" command in the interface configuration mode to configure IS-IS Level of this interface. The default is-type and circuit-type are Level-1-2. If these two commands are configured simultaneously, the corresponding interface will only send Level PDUs with is-type being same as circuit-type.

15.2.5.1 Configuring System Type

You can configure the level of existing routing devices, which can be divided into Level-1 router (intra-area routing device), Level-2 (inter-area routing device) and Level-1-2 router (both an intra-area routing device and an inter-area routing device). If is-type is configured to Level-1 or Level-2-only, the IS-IS process will only process data at this level. Complete the following configuration in IS-IS protocol configuration mode to set Level of the router:

Command	Function

Qtech(config-router)# **is-type** { **level-1** | **level-1-2** | **level-2-only** }	Configures system type.

15.2.5.2 Configuring the Interface Circuit Type

You can configure the type of the interface circuit. Complete the following configuration in interface configuration mode to set the type of the interface circuit:

Command	Function
Qtech(config-if)# **isis circuit-type** { **level-1** | **level-1-2** | **level-2-only** }	Configures the interface circuit type.

If the circuit-type of "Level-1" or "Level-2-only" is configured, then IS-IS will only send PDUs of the same level.

15.2.5.3 Configuring IS-IS Authentication

You can configure IS-IS authentication to improve the security of IS-IS network. You can configure authentication for IS-IS in the different level ranges, which include IS-IS interface, IS-IS area and IS-IS route domain.

The interface authentication functions during adjacency formation. If two IS-IS devices are configured with different interface authentication passwords, the adjacency won't be formed, thus avoiding unauthorized or unauthenticated IS-IS devices from joining an IS-IS network in which authentication is required. The interface authentication password is encapsulated in the Hello packets.

IS-IS domain authentication and routing domain authentication are used to authenticate LSP, CSNP and PSNP packets, so as to avoid unauthorized or unauthenticated routing information from entering IS-IS link-state database. The authentication password is encapsulated in the corresponding LSP, CSNP and PSNP packets.

Currently, the following two kinds of authentication methods are provided: plain text authentication and MD5 authentication. The approach of plain text authentication can only guarantee limited security as the password carried by packets can be seen directly. The approach of MD5 authentication will provide better security as the password carried by packets has been encrypted using MD5 algorithm.

15.2.6 Configuring Interface Authentication

Configuring Interface Plain-text authentication

You can configure plain-text authentication password for IS-IS interface. The authentication password will be encapsulated in Hello packets sent on the interface; when Hello packets are received, consistency of the password will be examined.

IS-IS interface authentication should be configured in interface configuration mode. You can use the following command to configure interface plain-text authentication:

Command	Function
Qtech(config)# **interface** *interface-name* Qtech(config-if)# **isis password** *password* [**send-only**] [**level-1** | **level-2**]	Configures plain-text authentication password for Hello packets transmitted on the interface. When the send-only is specified, the authentication password is only applicable to the sent Hello packets rather than the received Hello packets. When Level is not specified, the authentication password configured is by default applicable to every Level. When configuring this command, if "isis authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS interface authentication, but this command has a lower priority level. To configure this command, you need to delete "isis authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS interface:

Command	Function

Qtech(config-if)# **isis authentication mode text** [**level-1** | **level-2**]	Use this command to specify the mode of IS-IS interface authentication. When Level is not specified, the authentication mode configured is by default applicable to every Level. When configuring this command, if "isis password password [level-1|level-2]" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS interface authentication, but this command has a higher priority level.
Qtech(config-if)# **isis authentication key-chain name-of-chain** [**level-1** | **level-2**]	Configures the key chain used by IS-IS interface authentication. When Level is not specified, the key chain configured is by default applicable to every Level. This command must be configured together with "isis authentication mode" command in order to achieve IS-IS interface authentication. Neither of them can be omitted.
Qtech(config-if)# **isis authentication send-only** [**level-1** | **level-2**]	(Optional) The IS-IS interface authentication can only apply to the packets sent. No authentication will be performed on packets received. When Level is not specified, the send-only authentication mode configured is by default applicable to every Level. This command can be used to avoid network oscillation caused by the failure in temporary authentication. Before deploying IS-IS authentication for the entire network, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute "no isis authentication send-only" command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Interface Encryption Authentication

You can configure encryption authentication password for IS-IS interface. The authentication password will be encapsulated in Hello packets sent on the interface; when Hello packets are received, consistency of the password will be examined.

IS-IS interface authentication should be configured in interface configuration mode. You can use the following commands to configure interface encryption authentication:

Command	Function
Qtech(config-if)# **isis authentication mode md5** [**level-1** | **level-2**]	Sets the IS-IS interface authentication mode. When Level is not specified, the set authentication mode applies to all Levels. If the isis password password [level-1 | level-2] command is previously configured, it will be covered by this command. The two commands can both configure IS-IS interface authentication. This command has a higher priority.
Qtech(config-if)# **isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**]	Sets key-chain used for IS-IS interface authentication. When Level is not specified, the set key-chain applies to all Levels. This command must be used together with the isis authentication mode command to authenticate IS-IS interface.

Qtech(config-if)# **isis authentication send-only** [**level-1** | **level-2**]	(Optional) The IS-IS interface authentication can only apply to the packets sent. No authentication will be performed on packets received. When Level is not specified, the set authentication and password apply to all Levels. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire network, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no isis authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

15.2.6.1 Configuring Area Authentication

Configuring Area Plain-text Authentication

You can configure plain-text authentication password for IS-IS area. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in the area (Level-1); when the packets are received, consistency of the password will be examined.

IS-IS area authentication must be configured in IS-IS process mode. You can use the following command to configure IS-IS area plain-text authentication:

Command	Function
Qtech(config-router)# **area-password** *password* [**send-only**]	Configures area (Level-1) plain-text authentication password. When send-only is specified, the authentication password is only applicable to the sent packets rather than the received packets. When configuring this command, if "authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS area authentication, but this command has a lower priority level. To configure this command, you need to delete "authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS area:

Command	Function
Qtech(config-router)# **authentication mode text level-1**	Use this command to specify the mode of IS-IS area authentication. When configuring this command, if "area-password password" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS area authentication, but this command has a higher priority level.
Qtech(config-router)# **authentication key-chain** *name-of-chain* **level-1**	Configures the key chain used by IS-IS area authentication. This command must be configured together with "authentication mode" command in order to perform IS-IS area authentication. Neither of them can be omitted.

Qtech(config-router)# **authentication send-only level-1**	(Optional) The IS-IS area authentication can only apply to the packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication. Before deploying IS-IS authentication for the entire area, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute "no authentication send-only" command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Area Encryption Authentication

You can configure encryption authentication password for IS-IS area. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in the area (Level-1); when the packets are received, consistency of the password will be examined.

IS-IS area authentication must be configured in IS-IS process mode. You can use the following commands to configure IS-IS area encryption authentication:

Command	Function
Qtech(config-router)# **authentication mode md5 level-1**	Sets the IS-IS area authentication mode. If the area-password password command is previously configured, it will be covered by this command. The two commands can both configure IS-IS area authentication. This command has a higher priority.
Qtech(config-router)# **authentication key-chain** *name-of-chain* **level-1**	Sets key-chain used for IS-IS area authentication. This command must be used together with the authentication mode command to authenticate IS-IS area authentication.
Qtech(config-router)# **authentication send-only level-1**	(Optional) The IS-IS area authentication can only apply to packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire area, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

15.2.6.2 Configuring the Routing Domain Authentication

Configuring Routing Domain Plain-text Authentication

You can configure plain-text authentication password for IS-IS routing domain. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in Level-2; when the packets are received, consistency of the password will be examined.

IS-IS routing domain authentication must be configured in IS-IS process mode. You can use the following command to configure IS-IS routing domain plain-text authentication:

Command	Function
Qtech(config-router)# domain-password password [send-only]	Configure routing domain (Level-2) plain-text authentication password. When send-only is specified, the authentication password is only applicable to the sent packets rather than the received packets. When configuring this command, if "authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS routing domain authentication, but this command has a lower priority level. To configure domain-password, you need to delete "authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS routing domain:

Command	Function
Qtech(config-router)# **authentication mode text level-2**	Use this command to specify the mode of IS-IS routing domain authentication. When configuring this command, if "domain-password password" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS routing domain authentication, but this command has a higher priority level.
Qtech(config-router)# **authentication key-chain** *name-of-chain* **level-2**	Configures the key chain used by IS-IS routing domain authentication. This command must be configured together with authentication mode command in order to achieve IS-IS routing domain authentication. Neither of them can be omitted.
Qtech(config-router)# **authentication send-only level-2**	(Optional) The IS-IS routing domain authentication can only apply to the packets sent. Packets received will not be authenticated. This command can be used to avoid network oscillation caused by temporary authentication failure. Before deploying IS-IS authentication for the entire routing domain, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

15.2.6.3 Configuring Routing Domain Encryption Authentication

You can configure encryption authentication password for IS-IS routing domain. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in Level-2; when the packets are received, consistency of the password will be examined.

IS-IS routing domain authentication must be configured in IS-IS process mode. You can use the following commands to configure IS-IS routing domain encryption authentication:

Command	Function
Qtech(config-router)# **authentication mode md5 level-2**	Sets the IS-IS routing domain authentication mode. If the domain-password password command is previously configured, it will be covered by this command. The two commands can both configure IS-IS routing domain authentication. This command has a higher priority.

Qtech(config-router)# **authentication key-chain** *name-of-chain* **level-2**	Sets key-chain used for IS-IS routing domain authentication. This command must be used together with the authentication mode command to authenticate IS-IS routing domain authentication.
Qtech(config-router)# **authentication send-only level-2**	(Optional) The IS-IS routing domain authentication can only apply to packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire routing domain, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

15.2.7 Configuring IS-IS GR

IS-IS Graceful Restart (IS-IS GR) guarantees continuous data forwarding during the process of protocol restart. Currently, the high-end products of Qtech can support IS-IS GR during main/standby switchover, so as to guarantee continuity of key services.

15.2.7.1 Operating Mechanism of IS-IS GR

- IS-IS GR Realization Standard

RFC5306: Restart Signaling for IS-IS

- RFC5306 operating mechanism

RFC5306 defines requirements, operating methods and issues to be noticed when executing GR; successful GR depends on two principles: 1) the network topology maintains stable; 2) the node maintains non-stop data forwarding during IS-IS protocol restart.

There are two roles in GR: Restarter and Helper. Accordingly, IS-IS GR can be functionally divided into IS-IS GR Restart Capability and IS-IS GR Help Capability. Devices with GR Restart Capability can send GR requests and proactively execute graceful restart, while devices with GR Help Capability can receive GR requests and help the neighbor to execute graceful restart. The process of GR starts with the sending of GR requests by Restarter. The neighbor devices will enter Help mode after receiving such GR requests and assist Restarter to rebuild link-state database while maintaining the adjacencies with Restarter. The main operating mechanism is shown below:

When proceeding with IS-IS GR, the device will advertise its neighbors to maintain their adjacencies, so that the other devices on the network will not perceive the network change. The topological relationships will remain unchanged, and the neighbors will not recalculate routes and update the forwarding table. On the other hand, the link-state database will be synchronized and restored under the aid of neighbors, so that routes and forwarding table remain unchanged after GR, ensuring continuity of data forwarding.

During graceful restart of Restarter, the following steps will be involved:

- GR Restarter advertises the GR Helpers of such restart

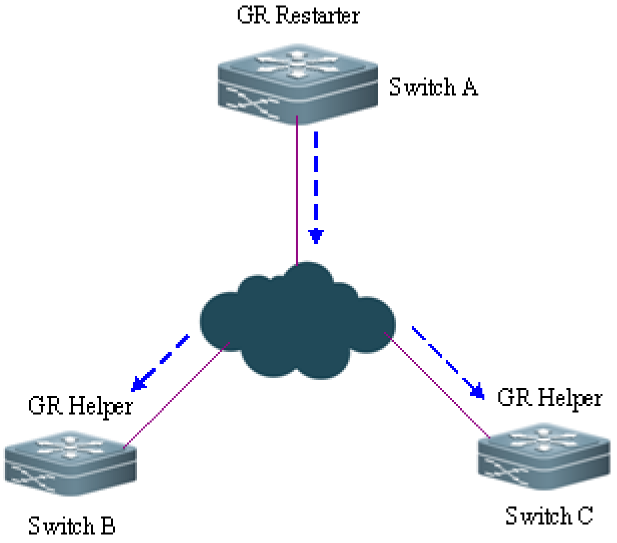


Figure 3 Restart of Restarter by advertising

As shown in Fig 3, Switch A is GR Restarter, Switch B and Switch C are GR Helpers of Switch A. Switch A sends GR requests to all its neighbors, as it needs to maintain the adjacencies during the process of GR. After receiving such requests, all neighbors will maintain the adjacencies with GR Restarter during the GR time (GR grace-period) advertised by the Restarter and send GR replies to Restarter.

- Restart of GR Restarter

As shown in Fig 4, when the GR Restarter proceeds with IS-IS restart, its IS-IS interface will undergo the process from Down to Up. Since the Helper is aware of the protocol restart state of Restarter, it will maintain its adjacency with GR Restarter and the routes acquired from GR Restarter during GR Time.

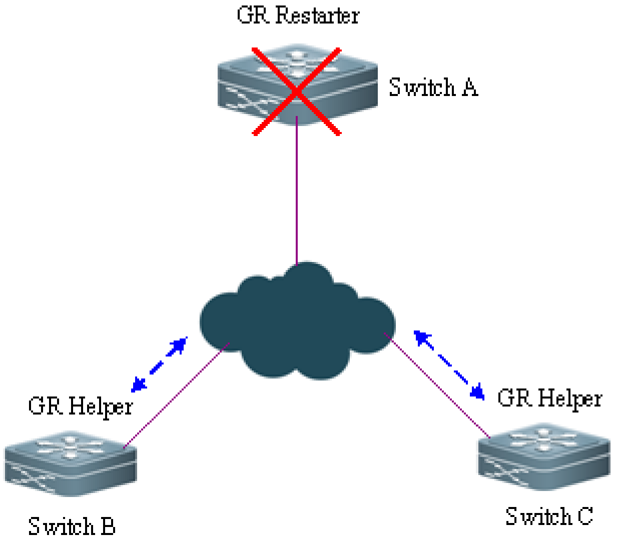


Figure 4 Restart of Restarter

- GR Restarter synchronizes with GR Helper and acquires the topology and routing information

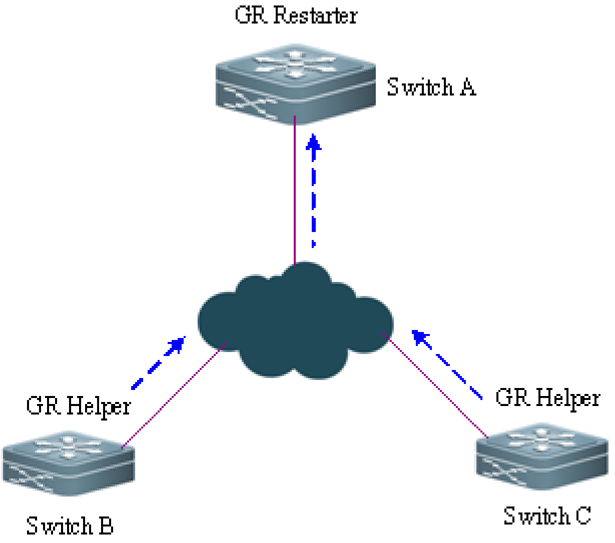


Figure 5 Database synchronization

As shown in Fig 5, after IS-IS protocol restart, GR Restarter will synchronize with GR Helper to acquire topology or routing information, and recalculate its own routing table accordingly. During this process, the forwarding table will not be updated by the routing table.

- GR Restarter completes database synchronization and graceful restart. All devices enter into IS-IS standard protocol interaction state.

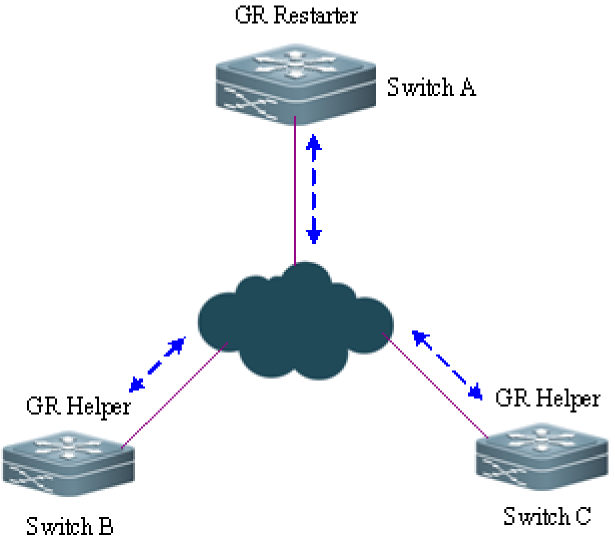


Figure 6 Completion of graceful restart

As shown in Fig 6, Restarter has completed data synchronization and all devices have entered into IS-IS standard protocol interaction state. By this time, the forwarding table will be updated by the routing table of Restarter and invalid entries will be deleted. Since the network maintains stable and Restarter has perfectly restored to the state before restart (completing graceful restart), its routing and forwarding tables will remain unchanged after the restart.

15.2.7.2 Use of IS-IS GR

GR of routing protocols is usually used to improve the system's reliability in the system that supports separation of the control panel and forwarding panel, thus realizing continued forwarding. IS-IS GR Restart capability depends on products.

When IS-IS GR Restarter capability is enabled, configure the IS-IS adjacency holdtime to no less than 40 seconds on devices with multiple management boards to ensure graceful restart of IS-IS triggered by switch of management boards. It can be achieved by configuring the isis hello-interval and isis hello-multiplier commands. If the holdtime value is less than 40 seconds, the holdtime value in the Hello packet header is set to 40 seconds by default.

The IS-IS GR Help capability only depends on software version. If the software supports IS-IS, the device is equipped with IS-IS GR Help capability.

15.2.7.3 Configuring IS-IS GR Restarter

To enable IS-IS graceful restart GR Restart capability, you must configure the graceful-restart command to enable graceful restart:

Command	Definition
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router isis**	Opens IS-IS and enters IS-IS configuration mode.
Qtech(config-router)# **graceful-restart grace-period seconds**	(Optional) Configures the restart cycle GR Time (default value: 300 seconds).
Qtech (config-router)# **end**	Returns to privileged mode.
Qtech # **show isis graceful-restart**	Verifies the configuration.
Qtech # **write**	(Optional) Save the configuration.

15.2.7.4 Configuring IS-IS GR Helper

IS-IS GR Help capability is enabled by default. You can also disable GR Help. The following example shows how to disable GR Help capability and re-enable it:

Command	**Definition**
Qtech # **configure terminal**	Enters global configuration mode.
Qtech (config)# **router isis**	Opens IS-IS and enters IS-IS configuration mode.
Qtech(config-router)# **graceful-restart helper disable**	Disables IS-IS GR Restarter capability on the neighbor of Restarter. The capability is enabled by default.
Qtech (config-router)# **no graceful-restart helper disable**	Re-enables IS-IS GR Help capability and restores it to the default action.
Qtech (config-router)# **end**	Returns to privileged mode.
Qtech # **show isis graceful-restart**	Verifies the configuration.
Qtech # **write**	(Optional) Save the configuration.

15.2.8 Configuring Linkage between IS-IS and BFD

The IS-IS protocol detects neighbors through Hello packets. After BFD detection is enabled with IS-IS, BFD session is established for the UP neighbors to monitor the neighbor status, Once the BFD neighbor is DOWN, IS-IS performs immediate convergence. The convergence time is reduced to 1s from 30s(by default, IS-IS Hello packets sending interval is 10s on a point-to-point network, and the failure time of the neighbor device is three times of the interval, that is, 30s),

In normal cases, BFD send detecting packets to detect link state with intervals in milliseconds. When the link gets abnormal, for example, the link is disconnected, BFD can detect link anomaly quickly and inform IS-IS to delete neighbors and neighbors-reachable information in LSP packets. IS-IS performs routing calculation again to generate new a route, avoiding the abnormal link and achieving fast convergence, With the introduction of new technologies such as Multi-Service Transport Platform (MSTP), link is congestion-prone in peak periods of data communication. In congestion, BFD can detect link anomaly quickly and inform IS-IS to delete neighbors and neighbors-reachable information in LSP packets. Besides, BFD perform the link switch to avoid congestion. As the IS-IS neighbor detects that the interval to send Hello packets is 10s and the timeout period is 30s. When BFD detects anomaly, the router can receive IS-IS and establish IS-IS adjacency relation. The route restores to the congested link and performs BFD detection again. BFD repeats the process of detecting link anomaly and performing link switch, making the route switched to either the congested link or other links and causing congestion.

Anti-congestion is enabled to avoid routing congestion caused by link congestion. Thus in link congestion, the IS-IS neighbor remains but the neighbor-reachable information is deleted in LSP packets. The route is switched to the non-congested link. After the link restores to normal, or rather non-congested, the neighbor-reachable information in LSP packets is restored and the route is switched back, avoiding routing congestion.

When IS-IS enables anti- congestion, both the **bfd all-interfaces** [**anti-congestion**] and the **bfd up-dampening** commands must be configured on the interface. Configuring only one command may cause ineffective anti- congestion or other network anomalies.

Command	Definition
Qtech(config-router)# **bfd all-interfaces** [**anti-congestion**]	Enables linkage between IS-IS and BFD on all interfaces.
Qtech(config-if)# **isis bfd** [**disable** | **anti-congestion**]	Enables or disables linkage between IS-IS and BFD on the interface.

Note

The BFD session needs to be set on the interface before configuring IS-IS with BFD.

Note

When the interface is configured with the **bfd up-dampening** command, the **bfd all-interfaces** [**anti-congestion**] command must be enabled if IS-IS is used with BFD on the interface.

Note

The **bfd all-interfaces** [**anti-congestion**] command must be configured together with the **bfd up-dampening** command on the interface.

Note

IP routing may cause inconsistency between the specified interface and the actual outbound interface of BFD packets, therefore the BFD session cannot be established.

Note

If the specified interface is not the actual inbound interface of BFD packets, the BFD session cannot be established.

15.2.9 Configuring IS-IS OVERLOAD

If the OVERLOAD flag in a non-virtual LSP packet, IS-IS neighbors can be notified for not using a local node as a transit device.

Command	Function
Qtech(config-router)# **set-overload-bit** [**on-startup** *seconds*] [**suppress** { [**interlevel**] [**external**] }] [**level-1** | **level-2**]	Sets the OVERLOAD flag.

The OVERLOAD flag is used in the following three cases:

- Device overload

If the overload (such as memory insufficiency and CPU full load) occurs on a local IS-IS node, the local routing table will be incomplete or contain no resource transit data. At this time, you can set the OVERLOAD flag in the LSP packet to notify the neighbors for not using the local node as a transit device.

In this case, the configuration does not carry the keyword on-startup, and the user must manually set or cancel the OVERLOAD flag. The user must manually cancel the OVERLOAD flag when the local IS-IS node recovers. Otherwise, the local IS-IS node will always stay in the OVERLOAD state.

- Instant black hole

In the scenario described in RFC3277, an instant black hole (instant blocked route) may occur after the IS-IS node is restarted because the IS-IS convergence speed is faster than that of BGP.

In this case, the configuration must carry the keyword on-startup. The IS-IS node automatically set or cancel the OVERLOAD flag based on the configuration. After the on-startup option is selected, an instant black hole occurs on the IS-IS node after it is restarted. Once a new neighbor relationship is created, the LSP packet carrying the OVERLOAD flag is sent for notifying the neighbor that there is an instant black hole on the local node and therefore do not use the local node as a transit device. When the specified period times out, the IS-IS node immediately sends the LSP packet where the OVERLOAD flag is canceled. This LSP packet notifies the neighbor that the instant black hole does not exist on the local node any more and therefore the neighbor can use the local node as a transit device.

- Requiring that the IS-IS node does not forward true data

The user may need the local IS-IS node to access the production network for experiments or other functions instead of forwarding true data on the network. At this time, the user can set the OVERLOAD flag in the LSP packet to notify the neighbors for not using the local node as a transit device.

In this case, the configuration does not carry the keyword on-startup, and the user must manually set or cancel the OVERLOAD flag. The user can configure the suppress function as required to suppress routing information carried in the LSP packet in the OVERLOAD state. If internal and external routes are suppressed, only the local directly-connected route is notified.

Configuring IS-IS VRF

Make sure VRF is configured before you binding the IS-IS instance to a VRF. To configure an IS-ISv6 neighbor, multi-protocol VRF must be configured and IPv6 must be enabled.

⚠ Caution The following restrictions should be noted for IS-IS and VRF bindings: 1. The IS-IS instances bound to a same VRF must be configured with different system IDs. While the IS-IS instances bound to different VRFs can share a system ID. 2. One IS-IS instance can be bound to only one VRF, and one VRF can be bound to multiple IS-IS instances. 3. When the VRF bound to an IS-IS instance changes, all the interfaces associated with the instance and the redistribution in routing process configuration mode will be removed accordingly.

Execute the following command to configure IS-IS VRF:

Command	Function
Qtech(config-router)# **vrf** *vrf-name*	Binds an IS-IS instance to a VRF.

15.2.10 Configuring IS-IS SNMP

By default, SNMP software can perform MIB on the first IS-IS instance displayed by the system. If you want to perfom MIB on another instance, specify it manually.

Execute the following command in IS-IS routing process configuration mode to bind the instance used for IS-IS MIB operation:

Command	Function
Qtech(config-router)# **enable mib-binding**	Binds the current instance to perform MIB.

There are 18 types of IS-IS packets. Based on different features, they are divided into several sets and each set includes several types of IS-IS TRAP packets. Enable IS-IS TRAP globally in global configuration mode (with the **snmp-server enable traps isis** command), specify the host receiving TRAP packets, and use this command to specify the types of IS-IS TRAP packets allowed to be sent in IS-IS routing process configuration mode. Then IS-IS packets can be transmitted.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)#**snmp-server enable traps isis**	Enables IS-IS TRAP globally.
Qtech(config)#**snmp-server host** 10.1.1.1	Configures global SNMP host and receives IS-IS TRAP packets.
Qtech(config)#**router isis**	Enters IS-IS routing pricess configuration mode
Qtech(config-router)# **enable traps all**	Allows all IS-IS TRAP packets to be sent to the host 10.1.1.1.

15.2.11 Configuring Other IS-IS Parameters

15.2.11.1 Configuring IS-IS Interface Metric

You can configure the interface metric, which must be configured in interface configuration mode as follows:

Command	Function
Qtech(config-if)# **isis metric metric** [**level-1**| **level-2**]	Configures the metric for the interface. This value is only effective when metric-style includes narrow mode.
Qtech(config-if)# **isis wide-metric metric** [**level-1** | **level-2**]	Configures the wide-metric for the interface. This value is only effective when metric-style includes wide mode.

15.2.11.2 Configuring the Priority Level of the Specified Routing Node

In the broadcast network, IS-IS needs to elect a designed routing node (DIS) among all routing nodes. The designated router will then create Pseudonode and generate Pseudonode LSP. In the broadcast network, the DIS is elected by priority, and the user can configure different priority values for different Levels. You can set different priority for different Levels. Complete the following configuration in interface configuration mode to set router priority for election:

Command	Function
Qtech(config-if)# **isis priority value** [**level-1** | **level-2**]	Configures the priority for designated router election on the interface.

⚠ Caution

The no isis priority command is used to restore the default priority no matter whether the parameter is followed. If you want to modify the configured priority, you can either use isis priority command with parameter specified to overwrite the configured command directly, or configure a new parameter after restoring the priority to the default value.

15.2.11.3 Configuring to Generate a Default Route

By default, L2 routers don't generate a default route. Execute the following command in IS-IS protocol configuration mode to generate a default route:

Command	Function
Qtech(config-router)# **default-information originate** [*route-map map-name*]	Generates a Level-2 default route and publishes through LSP. If the route-map option is specified, the default route can be generated only when the condition in the route-map is matched.

15.2.11.4 Configuring Convergent Route

You can create a convergent route to represent a group of routes in the routing table. The process is called route convergence. One convergent route can include multiple routes in a Level. The interface metric of the convergent route is the smallest one of all routes. Complete the following configuration in IS-IS protocol configuration mode to set the route convergence:

Command	Function
Qtech(config-router)#**summary-address ip-address net-mask** [**level-1** | **level-2** | **level-1-2**]	Sets convergent route.

Route convergence in IS-ISv6 protocol should be set in IS-ISv6 protocol configuration mode:

Command	Function

Qtech(config-router)# **address-family ipv6 unicast** Qtech(config-router-af)# **summary-prefix** *ipv6-prefix* / *prefix-length* [**level-1** | **level-2** | **level-1-2**]	Sets IS-IS IPv6 convergent route.

15.2.11.5 Configuring to Ignore LSP Authentication and Verification Errors

When local IS-IS receives LSP packet, the LSP packet needs to be verified and calculated. In addition, the calculation result is compared with the verification in the LSP packet. That is, the received LSP packet needs to be verified and authenticated. By default, if the calculation result is different from the verification in the LSP packet, the LSP packet is discarded and not processed. If you run the ignore-lsp- errors command to ignore the verification error, the LSP packet is processed normally even an error is verified. The configuration to ignore LSP authentication and errors must be performed in IS-IS protocol configuration mode:

Command	**Function**
Qtech(config-router)#**ignore-lsp-errors**	Configures to Ignore LSP authentication and verification errors.

15.2.11.6 Configuring to Open Adjacent Event Output Switch

To log events when the IS-IS adjacency changes, you need to open the adjacent event output switch. Complete the following configuration in IS-IS protocol configuration mode to set the adjacent event output switch:

Command	**Function**
Qtech(config-router)# **log-adjacency-changes**	Enable the logging of IS-IS adjacency change.

15.2.11.7 Configuring Route Redistribution

Route redistribution can redistribute one routing protocol's routing information to another routing protocol. Route redistribution must be configured in IS-IS configuration mode or IS-IS address-family ipv6 mode:

⚠ Caution

In case there are IS-IS Level-2 instances, all IS-IS Level-1 routes will by default be automatically redistributed into IS-IS Level-2 in these instances. Of course, you can also disable the redistribution from Level-1 into Level-2 by executing "no redistribute isis [tag] level-1 into level-2". You can also filter the redistributed Level-1 routes by executing "redistribute isis [tag] level-1 into level-2 {distribute-list access-list-name| route-map route-map-name}".

⚠ Caution

2. Configure no redistritbue {bgp | ospf <1-65535> | rip | connected | static} to disable protocol redistribution. If no redistribute is followed by any other parameter, it means that this parameter is restored to the default setting instead of disabling protocol redistribution. For example: no redistribute bgp will disable bgp redistribution, while no redistribute bgp route-map aa will disable route-map aa filtering during redistribution instead of disabling bgp redistribution.

⚠ Caution

3. In the old version software developed by some manufacturers, after configuring metric-type as external, the metric of redistributed route will be added by 64 during route calculation, and the route will be selected according to metric value. This has violated the protocol. In actual applications, the external route may have higher priority than the internal route. During the intercommunication with such manufacturers, if this problem exists, relevant configurations of devices can be adjusted (such as metric or metric-type) to ensure internal routes have higher priority than the external routes.

15.3 Monitoring and Maintaining IS-IS

View IS-IS's link status database and transmission and reception of various packets and calculation of SPF through the following configuration and operation to confirm maintenance of IS-IS route.

Command	Function
Qtech# **show isis** [*tag*] **database** [*FLAGS* | *LEVEL* | *LSPID*]	Displays IS-IS link-state database.
Qtech# **show isis** [*tag*] **neighbors** [**detail**]	Displays IS-IS neighbors.
Qtech# **show isis** [*tag*] **virtual-neighbors**	Displays IS-IS neighbors in virtual system.
Qtech# **show isis** [*tag*] **interface** [*interface-type interface-number*]	Displays IS-IS interface information.
Qtech# **show isis** [*tag*] **topology** [**l1** | **l2** | **level-1** | **level-2**]	Displays IS-IS connection topology.
Qtech# **show isis** [*tag*] **ipv6 topology** [**l1** | **l2** | **level-1** | **level-2**]	Displays IS-IS IPv6 unicast topology.
Qtech# **show isis** [*tag*] **counter**	Displays various statistics of IS-IS.
Qtech# **show isis** [*tag*] **hostname**	Displays the mapping relation between the device hostname and System ID.
Qtech# **show isis** [*tag*] **mesh-groups**	Displays the mesh group configurations on each interface.
Qtech# **show isis** [*tag*] **graceful-restart**	Displays IS-IS GR status.
Qtech# **show isis** [*tag*] **protocol**	Displays the IS-IS protocol.

For detailed explanation of commands, please refer to *IS-IS Commands*.

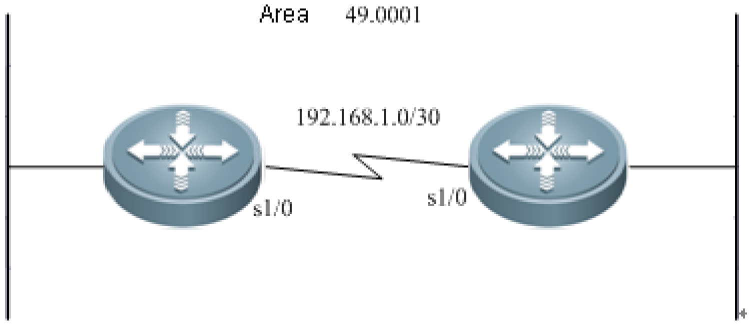
15.4 IS-IS Configuration Examples

15.4.1 IS-IS Point-to-Point Serial Link Configuration Example

- Requirement

The connection layout and IP address distribution are shown in Fig 7. A point-to-point network is configured between Device A and Device B.

Figure 7 IS-IS point-to-point serial link configuration



- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring the Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 10.1.1.1 255.255.255.0
Qtech(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring the Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 10.1.1.2 255.255.255.0
Qtech(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0003.00
```

Configuring the Ethernet port

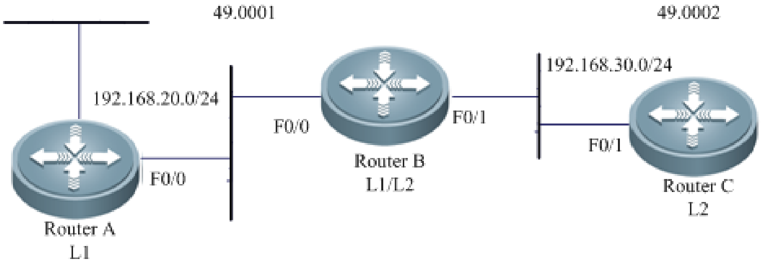
```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 10.1.1.3 255.255.255.0
Qtech(config-if)# ip router isis
```

15.4.2 IS-IS Broadcast Multipoint Link Configuration Example

- Requirement

The connection layout and IP address distribution are shown in Fig 8. Device A, Device B and Device C are interconnected through Ethernet, running the IS-IS routing protocol. Device A is Level-1 node, Device B is Level1-2 node and Device C is Level-2 node. It is required that Hello packets between Device A and Device B, Level-1 LSP and SNP packets adopt plaintext authentication, Hello packets between Device B and Device C, Level-2 LSP and SNP packets adopt MD5 encryption authentication,

Figure 8 IS-IS broadcast multipoint link configuration



- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 10.1.1.1 255.255.255.0
Qtech(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 10.1.1.2 255.255.255.0
Qtech(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0003.00
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 10.1.1.3 255.255.255.0
Qtech(config-if)# ip router isis
```

15.4.3 IS-IS Authentication Configuration Example

- Requirement

Three devices are interconnected through Ethernet and run IS-IS routing protocol. The connection layout and IP address distribution are shown in Fig 9. Device A is a Level-1 router, Device B is a Level-1-2 router, and Device C is a Level-2 router. Hello packets exchanged between Device A and Device B shall be subject to plain-text authentication, and Level-1 LSP and SNP packets shall be subject to plain-text authentication. Hello packets exchanged between Device B and Device C shall be subject to MD5 encrypted authentication, and Level-2 LSP and SNP packets shall be subject to MD5 encrypted authentication.

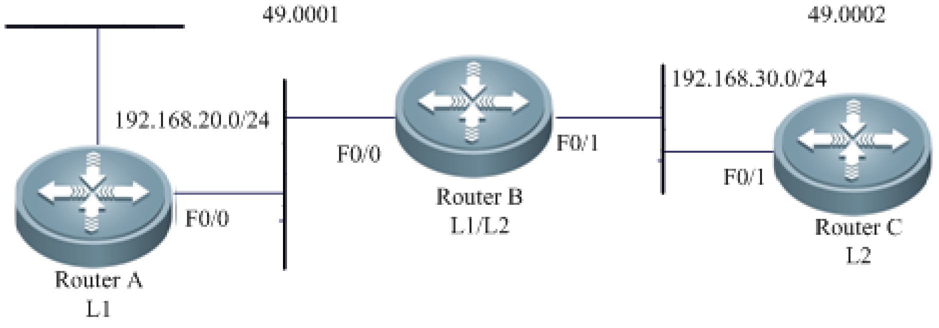


Figure 9 IS-IS authentication configuration

Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
Qtech(config-router)# is-type level-1
Qtech(config-router)# area-password aa
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 192.168.20.1 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config-if)# isis password cc
```

Device B:

Configuring the key chain used by IS-IS authentication:

```
Qtech(config)# key chain kc1
Qtech(config-keychain)# key 1
Qtech(config-keychain-key)# key-string aa
Qtech(config)# key chain kc2
Qtech(config-keychain)# key 1
Qtech(config-keychain-key)# key-string bb
Qtech(config)# key chain kc3
Qtech(config-keychain)# key 1
Qtech(config-keychain-key)# key-string cc
```

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
Qtech(config-router)# authentication mode text level-1
Qtech(config-router)# authentication key-chain kc1
Qtech(config-router)# authentication mode md5 level-2
Qtech(config-router)# authentication key-chain kc2
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 192.168.20.2 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config-if)# isis authentication mode text
Qtech(config-if)# isis authentication key-chain kc3
Qtech(config)# interface FastEthernet 0/1
Qtech(config-if)# ip address 192.168.30.2 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config-if)# isis authentication mode md5
Qtech(config-if)# isis authentication key-chain kc3
```

Device C:

Configuring the key chain used by IS-IS authentication:

```
Qtech(config)# key chain kc2
Qtech(config-keychain)# key 1
Qtech(config-keychain-key)# key-string bb
Qtech(config)# key chain kc3
Qtech(config-keychain)# key 1
Qtech(config-keychain-key)# key-string cc
```

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0002.0000.0000.0002.00
Qtech(config-router)# is-type level-2
Qtech(config-router)# authentication mode md5 level-2
Qtech(config-router)# authentication key-chain kc2
```

Configuring Ethernet interface

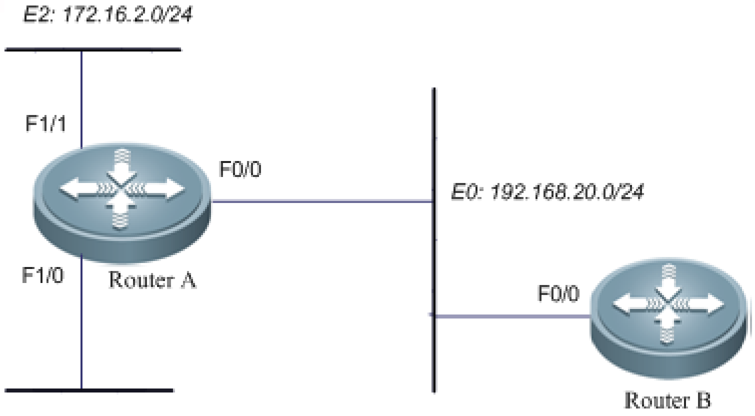
```
Qtech(config)# interface FastEthernet 0/1
Qtech(config-if)# ip address 192.168.30.3 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config-if)# isis authentication mode md5
Qtech(config-if)# isis authentication key-chain kc3
```

15.4.4 IS-IS Route Summary

- Requirement

Two devices are connected through Ethernet. The IP address distribution and device layout are shown in Fig 10.

Figure 10 IS-IS route summary configuration



Requirement

1. Two devices run IS-IS route protocol.

2. Configure Router A, so that Router A only advertises the route of 172.16.0.0/22 instead of routes of 172.16.1.0/24 and 172.16.2.0/24.

- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
Qtech(config-router)# summary-address 172.16.0.0/16 level-1-2
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 192.168.20.1 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config)# interface FastEthernet 1/0
Qtech(config-if)# ip address 172.16.1.1 255.255.255.0
Qtech(config-if)# ip router isis
Qtech(config)# interface FastEthernet 1/1
Qtech(config-if)# ip address 172.16.2.1 255.255.255.0
Qtech(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ip address 192.168.20.2 255.255.255.0
Qtech(config-if)# ip router isis
```

Execute show ip route on Device B to see only one summary address:

```
Qtech(config)# show ip route
i L1    172.16.0.0/16 [115/20] via 192.168.20.1, FastEthernet0/0
```

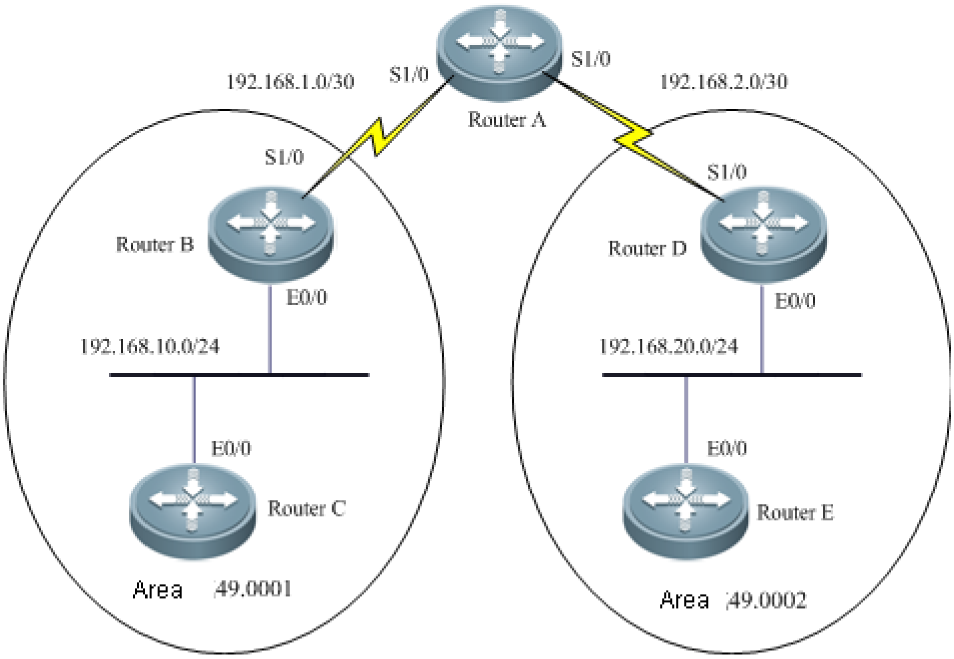
Caution If Level is no specified when summary-address is used, only Level-2 routes will be summarized by default.

15.4.5 IS-IS Level Configuration Example

- Requirement

See Fig 7 for allocation of IP addresses and connection of devices. P2P serial link connection is deployed between Device A and Device B and C respectively; Ethernet connection is deployed between Device B and Device C; Ethernet connection is deployed between Device D and Device E.

Figure 11 IS-IS level configuration



You need to configure IS-IS area route summary on Router A. Area route summary can only be configured on an area border device.

- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 50.0001.0000.0000.0001.00
Qtech(config-router)# is-type level-2-only
```

Configuring serial link interface

```
Qtech(config)# interface Serial 1/0
Qtech(config-if)# ip address 192.168.1.1 255.255.255.252
Qtech(config-if)# ip router isis
Qtech(config)# interface Serial 1/1
Qtech(config-if)# ip address 192.168.2.1 255.255.255.252
Qtech(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.10.1 255.255.255.0
Qtech(config-if)# ip router isis
```

Configuring serial link interface

```
Qtech(config)# interface Serial 1/0
Qtech(config-if)# ip address 192.168.1.2 255.255.255.252
Qtech(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0003.00
Qtech(config-router)# is-type level-1
```

Configuring Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.10.2 255.255.255.0
Qtech(config-if)# ip router isis
```

Device D:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0002.0000.0000.0004.00
```

Configuring Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.20.1 255.255.255.0
Qtech(config-if)# ip router isis
```

Configuring serial link interface

```
Qtech(config)# interface Serial 1/0
Qtech(config-if)# ip address 192.168.2.2 255.255.255.252
Qtech(config-if)# ip router isis
```

Device E:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0002.0000.0000.0005.00
Qtech(config-router)# is-type level-1
```

Configuring Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ip address 192.168.20.2 255.255.255.0
Qtech(config-if)# ip router isis
```

15.4.6 IS-ISv6 Simplest Configuration

- Requirement

The connection layout and IPv6 address distribution are shown in Fig 12. Device A and Device B are interconnected through Ethernet.

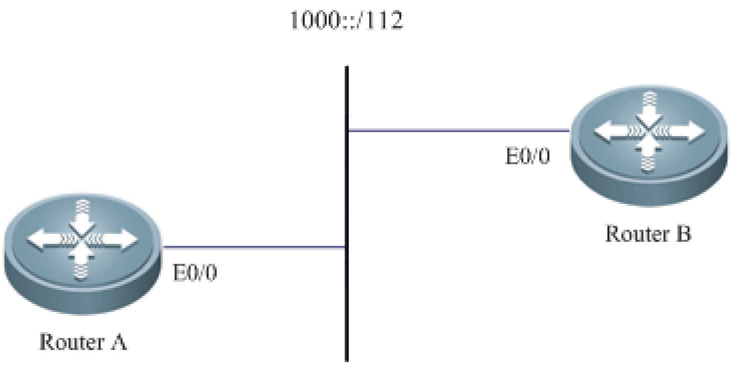


Figure 12 IS-ISv6 configuration

- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring Ethernet interface

```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ipv6 address 1000 ::1/112
Qtech(config-if)# ipv6 router isis
```

Device B:

Configuring IS-IS routing protocol

```
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

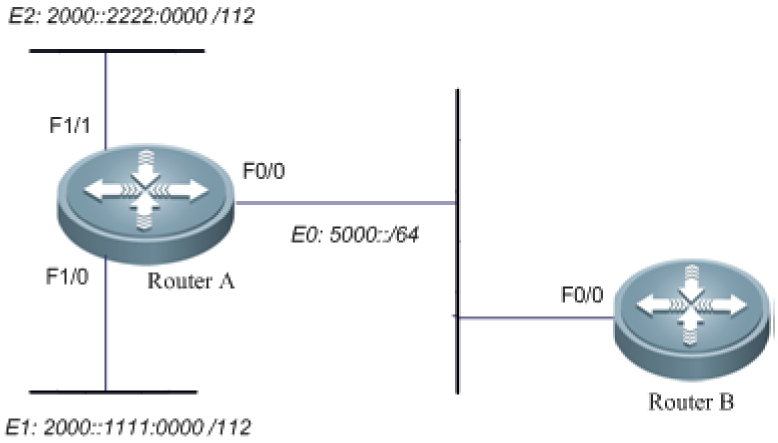
```
Qtech(config)# interface GigabitEthernet 0/0
Qtech(config-if)# ipv6 address 1000 ::2/112
Qtech(config-if)# ipv6 router isis
```

15.4.7 IS-ISv6 Route Summary

- Requirement

Two devices are connected through Ethernet. See Fig 13 for allocation of IP addresses and connection of devices.

Figure 13 IS-ISv6 route summary configuration



Requirement

Two devices run IS-ISv6 route protocol.

Configure Router A, so that Router A only advertises the route of 2000::/96 instead of routes of 2000::1111:0/112 and 2000::2222::0/112.

Detailed configurations

Device A:

```
Configuring IS-IS routing protocol
Qtech(config)# ipv6 unicast-routing
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0001.00
Qtech(config-router)# address-family ipv6 unicast
Qtech (config-router-af)# summary-prefix 2000::/96 level-1-2
Qtech (config-router-af)# exit-address-family
Configuring Ethernet interface
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ipv6 address 5000::1/64
Qtech(config-if)# ipv6 router isis
Qtech(config)# interface FastEthernet 1/0
Qtech(config-if)# ipv6 address 2000::1111:0001/112
Qtech(config-if)# ipv6 router isis
Qtech(config)# interface FastEthernet 1/1
Qtech(config-if)# ipv6 address 2000::2222:0001/112
Qtech(config-if)# ipv6 router isis
```

Device B:

```
Configuring IS-IS routing protocol
Qtech(config)# ipv6 unicast-routing
Qtech(config)# router isis
Qtech(config-router)# net 49.0001.0000.0000.0002.00
Configuring Ethernet interface
Qtech(config)# interface FastEthernet 0/0
Qtech(config-if)# ipv6 address 5000::2/64
Qtech(config-if)# ipv6 router isis
```

Execute "show ipv6 route" on Device B to see only one summary address:

```
Qtech(config)# show ipv6 route
I1   2000::/96 [115/20]     via FE80::C800:1BFF:FEF8:1C, FastEthernet1/0
```

⚠ Caution

If Level is no specified when summary-prefix is used, only Level-2 routes will be summarized by default.

16 CONFIGURING NHRP

16.1 Overview

Next Hop Resolution Protocol (NHRP) is defined by the IETF in RFC 2332. It is used to enable a source node (a host or router) on a non-broadcast multiple access (NBMA) network to obtain the Internet address and NBMA subnet address of "the next hop" destined to the destination node.

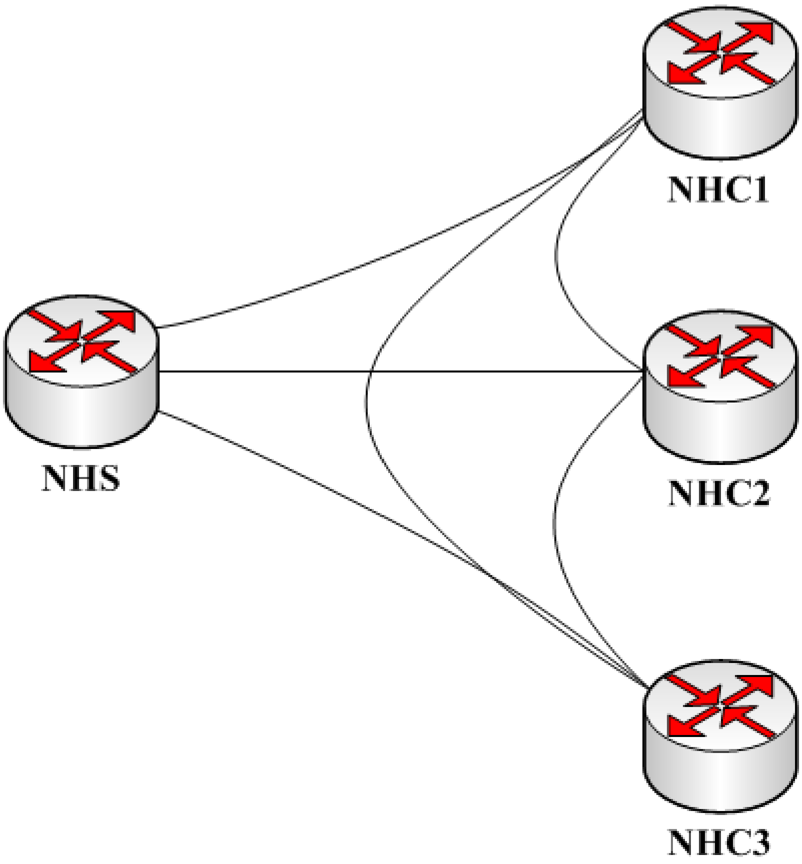
NHRP is a typical client/server protocol and comprises the Next Hop Server (NHS) and Next Hop Clients (NHCs). Each subnet comprises at least one NHS. One NHS can serve multiple subnets and receive request and resolution packets from multiple NHCs. Each end system is an NHC.

16.1.1 Working Principle

NHRP is similar to ARP that is used to complete resolution from IP addresses to MAC addresses. NHRP is used to complete resolution from internal VPN addresses to NBMA addresses (external addresses).

An NHS is used to maintain the NHRP database at the public IP address of each branch NHC. A branch NHC registers the IP address of its external port to the central router NHS through NHRP. The branch NHC can use a dynamic IP address. When the branch NHC wants to establish a direct tunnel with other branch NHCs, the branch NHC queries the NHRP database of the NHS for resolution to determine the real addresses of the other branch NHCs and make redundancy of the addresses to its own NHRP database. In this way, a branch NHC can dynamically join a network, which addresses the problem of establishing VPN tunnels through dynamic IP addresses. In addition, the branch NHC can directly exchange data with other branch NHCs by using the resolved addresses, rather than using the NHS for transmission, finally forming a mesh topology.

Figure 16-1 NHRP network topology



Negotiation packets of NHRP fall into the following types:

- NHRP Resolution Request: NHRP resolution request packet. For packets with uncertain mapping between IP addresses and NBMA addresses, an NHC sends a resolution request packet to the NHS for negotiation and obtain the mapping.
- NHRP Resolution Reply: NHRP resolution reply packet. After receiving the request, the NHS queries its cache table and sends the query result as the reply information to the NHC for address negotiation.
- NHRP Registration Request: NHRP registration request packet. After the NHC joins the network, the NHC needs to register the mapping between its IP address and NBMA address to the NHS in the current network. Therefore, the NHC sends a registration request packet to the NHS and registers itself to the NHS cache table.
- NHRP Registration Reply: NHRP registration reply packet. After receiving the registration request packet, the NHS stores the registration information into its cache table and sends an NHRP registration reply packet based on the registration result.
- NHRP Purge Request: NHRP purge request packet. When either NHRP party wants to purge the cache of a peer, the NHRP party will send an NHRP purge request packet.
- NHRP Purge Reply: NHRP purge reply packet. After receiving the NHRP purge request packet, the peer will send a reply packet to the purge requester based on the cache purge result.
- NHRP Error Indication: NHRP error indication packet. If a receiving error packet is generated during NHRP negotiation, this packet will be send to the original sender.

The two primary work processes of NHRP are the registration process and resolution process.

16.1.1.1 Registration Process

An NHC needs static cache, which is used to cache the mapping between the IP address and NBMA address of the NHS in the current network. The NHC generates and sends a registration request packet to the NHS. After receiving the packet, the NHS will generate dynamic cache, and generate and send a registration reply packet to the NHC for registration.

16.1.1.2 Resolution Process

When no NBMA address is found for the mapped IP address, the NHS and NHC will send a resolution request packet for negotiation about a proper address.

As shown in the figure above, HHC1 wants to ping the IP address of NHC2. The resolution negotiation process is as follows:

- When pinging packets, NHC1 queries the NBMA address mapped to the IP address of NHC2. Since no address is resolved, NHC1 sends a resolution request packet to the NHS.
- After receiving the resolution request packet, the NHS sends the corresponding registration cache entry of NHC2 to NHC1 as a resolution reply packet. In this way, NHC1 has learned the NBMA address mapped to the IP address destined to NHC2.
- After the ping packet reaches NHC2, NHC2 will send an ICMP reply packet. Since the source IP address of the ping packet is unresolved, NHC2 also sends a request packet to the NHS to resolve the correct address of NHC1 by following the method of NHC1.
- Then, data packets from NHC1 to NHC2 will be transmitted through the direct route between them, rather than be transmitted through the NHS, forming a mesh network.

If the NHRP resolution request is triggered by a data packet, NHC1 will select one of the following methods to process the data packet when waiting for an NHRP resolution response:

① Releases a data packet.

② Retains the data packet until the NHRP resolution response arrives and another optimum path is available.

③ Sends a data packet along the route.

Method ③ is the recommended default policy since it can resolve subnet-layer addresses as well as send data flows to destination sites.

16.1.2 Protocols and Standards

RFC 2332.

16.2 Default Configuration

NHRP is disabled by default.

16.3 Configuring NHRP

16.3.1 Enabling NHRP

The NHRP function can take effect on an interface only after it is enabled.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**tunnel mode gre ip**	Configures the tunnel type as GRE. This type of tunnels is used to form a star topology. The tunnel type of the NHS must be MGRE.
	Or:	
	Qtech(config-Tunnel *tunnel-number*)#**tunnel mode gre multipoint**	Configures the tunnel type as MGRE. This type of tunnels is used to form a mesh topology. The tunnel type of the NHS must be MGRE.
Step 3	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp network-id***nhrp-number*	Enables the NHRP function on a tunnel interface. *nhrp-number*: Indicates a 32-bit ID. <1-4294967295>.
	Qtech(config-Tunnel *tunnel-number)*#**no ip nhrp network-id**	Disables the NHRP function.

Caution The NHRP function must be enabled on a tunnel interface and support only GRE and MGRE tunnels.

Example 1: Enable the NHRP function on a tunnel interface.

```
Qtech(config)#interface tunnel1
Qtech(config-Tunnel 1)#ip nhrp network-id1
```

16.3.2 Configuring Static IP-Address and NBMA-Address Mapping Cache

The IP-address and NBMA-address mapping cache of NHRP may be either static or dynamic. Static mapping cache can be manually configured whereas dynamic mapping cache is learned by sending protocol packets.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address nbma-address*	Configures the IP-address and NBMA-address mapping cache on a tunnel interface. The cache is static and will not time out forever. The former address is an IP address and the latter address is an NBMA address.
	Or:	
	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address***mask** *ip-mask nbma-address*	An IP address is a certain address or the IP address of a network segment, which should be configured with a mask.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp map** *ip-address nbma-address*	Deletes the static configuration.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp map** *ip-address* **mask***ip-mask nbma-address*	Deletes the static configuration.

Example 1: Configure static NHRP cache without a mask on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp map 55.1.1.1 11.1.1.1
```

Example 2: Configure static NHRP cache with a mask on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp map 55.1.1.0 mask 255.255.255.0 11.1.1.1
```

16.3.3 Configuring Destination Addresses of Broadcast and Multicast Packets

When a device needs to send broadcast and multicast packets, you need to specify the direction for sending the packets.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map multicast** *nbma-address*	Configures the NBMA addresses for sending broadcast and multicast packets through the tunnel. Multiple NBMA addresses can be configured.
	Or:	
	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map multicast dynamic**	The sending direction can also be dynamically learned. After the dynamic learning command is configured, all NBMA addresses registered on the device can be used as the addresses for sending the broadcast packets.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp map multicast** *nbma-address*	Deletes the static configuration.
	Qtech(config-Tunnel *tunnel-number*)# **no ip nhrp map multicast dynamic**	Disables the configuration.

Caution This command is not supported by point-to-point GRE tunnels, but is supported only by MGRE tunnels.

Example 1: Configure a static address 11.1.1.1 for sending broadcast packets on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp map multicast 11.1.1.1
```

Example 2: Configure a dynamic address learning command for sending broadcast packets on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp map multicast dynamic
```

16.3.4 Configuring NHS

Configure a static address of an NHS to which a registration request packet needs to be sent on an NHC to trigger sending of the registration request.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address nbma-address*	Configures the static IP-address and NBMA-address mapping of the NHS.

	Command	Description
Step 3	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp nhs** *nhs-address*	Configures the address of the NHS with which an NHC needs to register for sending a registration request packet.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp nhs** *nhs-address*	Disables the configuration.

Example 1: Configure the NHRP NHS address 11.1.1.1 on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp nhs11.1.1.1
```

16.3.5 Configuring Only the NHS Mode

Configure only the NHS mode in which no NHRP registration request packet will be sent.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp server-only**	Configures the tunnel interface only in the NHS mode.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp server-only**	Resets the default configuration.

Example 1: Configure only the NHS mode on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp server-only
```

16.3.6 Configuring Hold Time of NHRP Cache

Dynamic cache information obtained by NHRP through negotiation has hold time. The following commands are used to configure the hold time of NHRP cache.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp holdtime** *seconds*	Configures the hold time of dynamic NHRP cache. The value of *seconds* ranges from 1 to 65,535. The default value is 7,200 seconds.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp holdtime**	Resets the default configuration.

Example 1: Configure the hold time of NHRP cache as 300 seconds on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp holdtime 300
```

16.3.7 Configuring NHRP Authentication

The NHRP information transferred in the same network can be authenticated with the following commands. Only NHRP information that can be successfully authenticated can be processed.

Command	Description

Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp authentication***string*	Configures the NHRP authentication *string* of 8 bytes, which should be carried by the negotiation packet.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp authentication**	Disables the configuration.

Example 1: Configure the NHRP authentication string as 123 on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp authentication 123
```

16.3.8 Configuring Conditions for NHRP to Trigger a Resolution Request

NHRP can control the conditions for triggering sending of resolution request packets so that certain packets cannot trigger sending of resolution requests.

	Command	Description
Step 1	Qtech(config)#**access-list***access-list-number*	Configures an ACL rule.
Step 2	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 3	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp interest** *access-list-number*	Uses the ACL for matching. Negotiation can be performed only for packets matching the ACL. *access-list-number*: Indicates a standard or an extended ACL number, ranging from 1 to 199.
	Or:	
	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp interest none**	No ACL can be matched and negotiation cannot be triggered.

Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp use***usage-count*	Configures the packet count. NHRP negotiation can be triggered only after *usage-count* (number of packets) packets are sent. The value range is from 1 to 65,535 and the default value is 1.

	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp interest**	Disables the configuration.
	Qtech(config-Tunnel *tunnel-number*)# **no ip nhrp use**	Resets the default configuration.

Example 1: Configure an NHRP ACL rule on a tunnel interface to match ACL 100.

```
Qtech(config)#access-list 100 permit ip any any
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp interest 100
```

Example 2: Configure the NHRP delayed triggering command on a tunnel interface. Negotiation can be triggered only after 10 packets are sent.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp use 10
```

16.3.9 Configuring NHRP Redirection

When the central node forwards the mutual access traffic of branch nodes, the central node needs to enable the **ip nhrp redirect** function. After that, the central node will send the **ip nhrp redirect** message to the source branch node to enable the source branch to send an NHRP resolution request.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp redirect**	Enables the NHRP redirection function on the NHS.

Example 1: Enable the **nhrp redirect** function on the tunnel 1 interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp redirect
```

16.3.10 Configuring the NHRP Shortcut Function

When forwarding mutual access traffic of branch nodes, the central node identifies whether the inbound and outbound interfaces for forwarding the packets belong to the same DMVPN. If yes, the central node will send a redirection packet to the source branch node. In this case, the source branch node needs to enable the **ip nhrp shortcut** function. The branch node can send a next-hop route resolution request packet only after receiving the redirection packet.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp shortcut**	Enables the NHRP shortcut function on the NHS.

Example 1: Enable the **nhrp shortcut** function on the tunnel 1 interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp shortcut
```

16.3.11 Configuring the NHRP Group to Which an NHC Belongs

An NHC can configure an NHRP group to which it belongs and advertises the NHRP group to the NHS during registration so that the NHS executes the per-spoke flow rate-limit function.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp group** *group-name*	Configures the NHRP group to which an NHC belongs.

Example 1: Configure the name of the NHRP group to which an NHC belongs as group-user-1.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp group group-user-1
```

16.3.12 Configuring Per-spoke Flow Rate-Limit on the NHS

You can limit the rates of flows forwarded by the NHS to each NHC. The same flow rate-limit policy will be applied to all NHCs registering to this *group*.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp map group** *group-name***flow-label** *label-value*	Specifies the flow label corresponding to the NHRP group.

Example 1: Configure the flow label for the group named group-user-1 as 3.

```
Qtech(config)#flow-limit output label 3 300000 3000 3000 conform-action  transmit
exceed-action drop
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp map group group-user-1 flow-label 3
```

16.3.13 Configuring the Transmission Rate of NHRP Negotiation Packets

You can run the following commands to limit the transmission rate of NHRP negotiation packets.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp max-send***pkt-count***every***interval*	Configures the transmission rate of NHRP negotiation packets as *pkt-count* per *interval* seconds. *pkt-count:* Indicates the number of packets sent. The value range is from 1 to 65,535 and the default value is **100**. *interval*: Indicates the interval for sending packets. The value range is from 10 to 65,535 and the default value is **10**.
	Qtech(config-Tunnel *tunnel-number*)#**noip nhrp max-send**	Disables the configuration.

Example 1: Configure the transmission rate of NHRP negotiation packets as 50 per 10 seconds on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp max-send 50every10
```

16.3.14 Configuring the Forward and Reverse Record Function

The forward record and reverse record extension fields are carried in NHRP request and reply packets, which are used to record the information about all NHSs before the request and reply packets reach the final NHS. This command is used to enable or disable the record function. The record function is enabled by default. The following commands are used to disable the function.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp record**	Disables the record function.
	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp record**	Enables the record function again.

Example 1: Disable the record function on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#no ip nhrp record
```

16.3.15 Configuring the Address for the NHRP Responder Field

An NHRP packet has the Responder Address extension field. This extension field is used to add the receiver's address information to this field of the reply packet when a request packet is received. This command is used to specify the address information in the Responder Address field.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.

Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp responderinterface-name***interface-number*	Configures the first IP address of the specified interface as the address to be added to the Responder Address extension field of the NHRP reply packet.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp responder**	Disables the configuration.

Example 1: Add the address of the virtual-ppp1 interface to the Responder Address field on a tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp responder virtual-ppp 1
```

16.3.16 Configuring the Conditions for Sending NHRP Registration Request Packets

An NHC will send a registration request packet to the NHS to register its information. The following commands are used to control sending of the registration request packet.

	Command	Description
Step 1	Qtech(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp registration no-unique**	The **U** field is available in the NHRP registration request packet, indicating that the registration is unique. This command indicates that the registration is not unique. The value **0** of the **U** field indicates multiple registrations.
	Or:	
	Qtech(config-Tunnel *tunnel-number*)#**ip nhrp registration timeout***seconds*	Configures the interval for sending two NHRP registration request packets. *seconds:* Indicates the interval for sending registration request packets, ranging from 1 to 65,535. The default value is 1/3 of the hold time.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp registration no-unique**	Disables the configuration.
	Qtech(config-Tunnel *tunnel-number*)#**no ip nhrp registration timeout**	Resets the default configuration.

Example 1: Configure NHRP registration requests not unique on a tunnel interface.

```
Qtech(config)#interface tunnel1
Qtech(config-Tunnel 1)#ip nhrp registration no-unique
```

Example 2: Configure the interval for sending NHRP registration requests as 10 seconds on the tunnel interface.

```
Qtech(config)#interface tunnel 1
Qtech(config-Tunnel 1)#ip nhrp registration timeout10
```

16.3.17 Enabling the NHRP Debug Command

Qtech Network products provide the debug commands for monitoring and maintaining the NHRP module.

Command	Description
Qtech#**debug nhrp**	Debugs the negotiation process of NHRP.
Qtech#**debug nhrp cache**	Debugs cache management of NHRP.
Qtech#**debug nhrp extension**	Debugs extension fields of NHRP.

Command	Description
Qtech#**debug nhrp packet**	Debugs packet information display of NHRP.
Qtech#**no debug nhrp**	Disables debugging by using the no command.
Qtech#**no debug nhrp cache**	Disables debugging by using the no command.
Qtech#**no debug nhrp extension**	Disables debugging by using the no command.
Qtech#**no debug nhrp packet**	Disables debugging by using the no command.

16.3.18 Clearing NHRP Cache Information

Qtech Network products provide the clear commands for clearing the IP-address and NBMA-address mapping table cache of the NHRP module.

Command	Description
Qtech#**clear ip nhrp**	Clears all non-static cache information of NHRP.
Qtech#**clear ip nhrp***ip-address*	Clears non-static NHRP cache of a specified IP-address without a mask.
Qtech#**clear ip nhrp***ip-addressip-mask*	Clears non-static NHRP cache of a specified IP-address with a mask.

16.3.19 Clearing NHRP Packet Statistics

Qtech Network products provide the clear commands for clearing NHRP packet statistics.

Command	Description
Qtech#**clear ip nhrp counters**	Clears all NHRP packet statistics.
Qtech#**clear ip nhrp counterstunnel***number*	Clears NHRP packet statistics of a specified tunnel interface.

16.3.20 Displaying

Qtech Network products provide the show commands for displaying information of the NHRP configuration.

Command	Description
Qtech#**show ip nhrp**	Displays all NHRP cache information.
Qtech#**show ip nhrp brief**	Displays all NHRP cache information briefly.
Qtech#**show ip nhrp dynamic**	Displays dynamic NHRP cache information.
Qtech#**show ip nhrp incomplete**	Displays NHRP cache information about incomplete negotiation.
Qtech#**show ip nhrp** *ip-address*	Displays NHRP cache information of a specified address.
Qtech#**show ip nhrp multicast**	Displays the address for sending multicast packets.
Qtech#**show ip nhrp nhs**	Displays NHS information.
Qtech#**show ip nhrp static**	Displays static NHRP cache information.
Qtech#**show ip nhrp summary**	Displays statistics about NHRP cache usage.

Command	Description
Qtech#**show ip nhrp traffic**	Displays receiving and sending information of NHRP packets.
Qtech#**show ip nhrpTunnel** *tunnel-number*	Displays NHRP cache information of a specified interface.

Example 1: Display all cache table information of NHRP. Two cache tables are available.

```
Qtech#show ip nhrp
55.1.1.2/32 via 55.1.1.3, Tunnel 2 created 00:04:08, expire: 01:55:52
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.2
 55.1.1.1/32 via 55.1.1.1, Tunnel 2 created 00:05:15, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
```

Example 2: Display the NHRP cache table information in brief.

```
Qtech# show ip nhrp brief
    Target        Via        NBMA     Mode   Intfc   Claimed
  55.1.1.1/32   55.1.1.1   11.1.1.1     static   Tu1      <   >
```

Example 3: Display the address information table for sending NHRP multicast packets.

```
Qtech# show ip nhrp multicast
  I/F      NBMA address
  Tunnel2     1.1.1.1        Flags: static
  Tunnel2     2.2.2.2        Flags: static
```

Example 4: Display NHRP NHS table information.

```
Qtech# show ip nhrp nhs
Tunnel 1  11.1.1.1   NO Responding
Tunnel 1  11.1.1.2   UP
```

Example 5: Display the NHRP cache table statistics.

```
Qtech# show ip nhrp summary
 IP NHRP cache 0 entries, 0 bytes
 0 static  0 dynamic  0 incomplete
```

Example 6: Display the sending and receiving statistics about NHRP packets.

```
Qtech# show ip nhrp traffic
Tunnel 1
  Sent: Total 0
         0 Resolution Request  0 Resolution Reply  0  Registration Request
         0 Registration Reply  0 Purge Request  0  Purge Reply
         0 Error Indication
  Rcvd: Total 0
         0 Resolution Request  0 Resolution Reply  0  Registration Request
         0 Registration Reply  0 Purge Request  0  Purge Reply
         0 Error Indication
```

16.4 Configuration Example

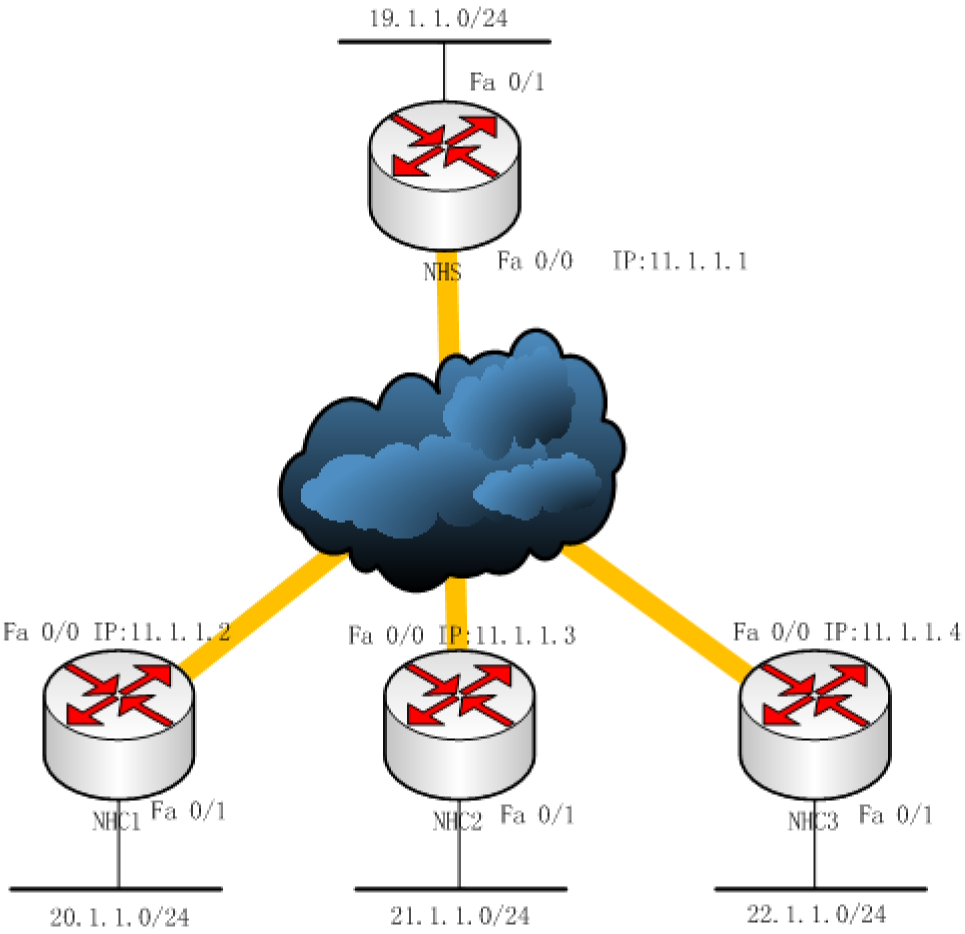
16.4.1 Configuring DMVPN

At present, NHRP is mainly used in a DMVPN (dynamic multi-point VPN) network. The IPSec, GRE and NHRP modules can be used together.

16.4.1.1 Networking Requirements

As shown in the following figure, four routers are needed. One router is used as the NHS and the other routers are used as the NHCs. They form a mesh topology.

16.4.1.2 Network Topology



DMVPN Network Topology

16.4.1.3 Configuration

If a mesh topology is needed, the three NHCs must be configured as MGRE tunnels. If a star topology is needed, the three NHCs must be configured as GRE tunnels.

16.4.1.4 Configuration Steps

Configuring NHS

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
```

```
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.1 255.255.255.0
no ip redirects
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 1000
ip nhrp holdtime 600
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.1 255.255.255.0
 duplex half
!
interface FastEthernet0/1
 ip address 19.1.1.1 255.255.255.0
 duplex half



router ospf 1
network 19.1.1.0 0.0.0.255 area 0
 network 55.1.1.0 0.0.0.255 area 0
```

Configuring NHC1

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.2 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.2 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 20.1.1.0 0.0.0.255 area 0
```

Configuring NHC2

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.3 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.3 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 21.1.1.0 0.0.0.255 area 0
```

Configuring NHC3

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
```

```
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.4 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.4 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 22.1.1.0 0.0.0.255 area 0
```

16.4.1.5 Verification

NHS Verification

```
NHS#show ip nhrp
 55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.2
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.4
```

NHC1 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.4
```

NHC2 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.4
```

NHC3 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:00:30, expire: 01:59:30
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:00:30, expire: 01:59:30
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.2
```