Edge-corE

Powered by Accton

ES4704(10)BD
Routing Management
Guide

www.edge-core.com

Content

Chapter 1 Routing Protocol

1.1 Routing Protocol Overview

To communicate with a remote host over the Internet, a host must choose a proper route via a set of routers or Layer3 switches.

Both routers and layer3 switches calculate the route using CPU, the difference is that layer3 switch adds the calculated route to the switch chip and forward by the chip at wire speed, while the router always store the calculated route in the route table or route buffer, and data forwarding is performed by the CPU. For this reason, although both routers and switches can perform route selection, layer3 switches have great advantage over routers in data forwarding. The following describes basic principle and methods used in layer3 switch route selection.

In route selection, the responsibility of each layer3 switch is to select a proper midway route according to the destination of the package received; and send the package to the next layer3 switch until the last layer3 switch in the route send the package to the destination host. A route is the path selected by each layer3 switch to pass the package to the next layer3 switch. Route can be grouped into direct route, static route and dynamic route.

Direct route refer to the path directly connects to the layer3 switch, and can be obtained with no calculation.

Static route is the manually specified path to a network or a host; static route cannot be changed freely. The advantage of static route is simple and consistent, and it can limit illegal route modification, and is convenient for load balance and route backup. However, as this is set manually, it is not suitable for mid- or large-scale networks for the route in such conditions are too huge and complex.

Dynamic route is the path to a network or a host calculated by the layer3 switch according to the routing protocols enabled. If the next hop layer3 switch in the path is not reachable, layer3 switch will automatically discard the path to that next hop layer3 switch and choose the path through other layer3 switches.

There are two dynamic routing protocols: Interior Gateway Protocol (IGP) and External Gateway protocol (EGP). IGP is the protocol used to calculate the route to a destination inside an autonomous system. IGP supported by ES4700 series include RIP and OSPF, RIP and OSRF can be configured according to the requirement. ES4700 series supports running several IGP dynamic routing protocols at the same time. Or, other dynamic routing protocols and static route can be introduced to a dynamic routing protocol, so that multiple routing protocols can be associated.

EGP is used to exchange routing information among different autonomous systems, such as

BGP protocol. EGP supported by ES4700 series switch include BGP-4, BGP-4+.

1.1.1 Routing Table

As mentioned before, layer3 switch is mainly used to establish the route from the current layer3 switch to a network or a host, and to forward packages according to the route. Each layer3 switch has its own route table containing all routes used by that switch. Each route entry in the route table specifies the physical port should be used for forwarding package to reach a destination host or the next hop layer3 switch to the host.

The route table mainly consists of the following:

Destination address: used to identify the destination address or destination network of an IP package.

Network mask: used together with destination address to identify the destination host or the network the layer3 switch resides. Network mask consists of several consecutive binary 1's, and usually in the format of dotted decimal (an address consists of 1 to 4 255's.) When "AND" the destination address with network mask, we can get the network address for the destination host or the network the layer3 switch resides. For example, the network address of a host or the segment the layer3 switch resides with a destination address of 200.1.1.1 and mask 255.255.255.0 is 200.1.1.0..

Output interface: specify the interface of layer3 switch to forward IP packages.

IP address of the next layer3 switch (next hop): specify the next layer3 switch the IP package will pass.

Route entry priority: There may be several different next hop routes leading to the same destination. Those routes may be discovered by different dynamic routing protocols or static routes manually configured. The entry with the highest priority (smallest value) becomes the current best route. The user can configure several routes of different priority to the same destination; layer3 switch will choose one route for IP package forwarding according to the priority order.

To avoid too large route table, a default route can be set. Once route table look up fails, the default route will be chosen for forwarding packages.

The table below describes the routing protocols supported by ES4700 series and the default route look up priority value.

Routing Protocols or route type	Default priority value
Direct route	0
OSPF	110
Static route	1
RIP	120
OSPF ASE	150
IBGP	200

EBGP	20
Unknown route	255

1.2 IP Routing Policy

1.2.1 Introduction To Routing Policy

Some policies have to be applied when the router publishing and receiving routing messages so to filter routing messages, such as only receiving or publishing routing messages meets the specified conditions. A routing protocol maybe need redistribute other routing messages found by other protocols such as OSPF so to increase its own routing knowledge; when the router redistributing routing messages from other routing protocols there may be only part of the qualified routing messages is needed, and some properties may have to be configured to suit this protocol.

To achieve routing policy, first we have to define the characteristics of the routing messages to be applied with routing policies, namely define a group matching rules. We can configure by different properties in the routing messages such as destination address, the router address publishing the routing messages. The matching rules can be previously configured to be applied in the routing publishing, receiving and distributing policies.

Five filters are provided in ES4700 series: route-map, acl, as-path, community-list and ip-prefix for use. We will introduce each filter in following sections:

1. route-map

For matching certain properties of the specified routing information and setting some routing propertities when the conditions are fulfilled.

Route-map is for controlling and changing the routing messages while also controlling the redistribution among routes. A route-map consists of a series of match and set commands in which the match command specifies the conditions required matching, and the set command specifies the actions to be taken when matches. The route-map is also for controlling route publishing among different route process. It can also used on policy routing which select different routes for the messages other than the shortest route.

A group matches and set clauses make up a node. A route-map may consist of several nodes each of which is a unit for matching test. We match among nodes with by sequence-number. Match clauses define matching rules. The matching objects are some properties of routing messages. Different match clause in the same node is “and” relation logically, which means the matching test of a node, will not be passed until conditions in its entire match clause are matched. Set clause specifies actions, namely configure some properties of routing messages after the matching test is passed.

Different nodes in a route-map is an “or” relation logically. The system checks each node of the route-map in turn and once certain node test is passed the route-map test will be passed without taking the next node test.

2. access control list(acl)

ACL (Access Control Lists) is a data packet filter mechanism in the switch. The switch controls the network access and secure the network service by permitting or denying certain data packet transmtting out from or into the network. Users can establish a group of rules by certain messages in the packet, in which each rule to be applied on certain amount of matching messages: permit or deny. The users can apply these rules to the entrance or exit of specified switch, with which data stream in certain direction on certain port would have to follow the specified ACL rules in-and-out the switch. Please refer to chapter “ACL Configuration”.

3. Ip-prefix list

The ip-prefix list acts similarly to acl while more flexible and more understandable. The match object of ip-prefix is the destination address messages field of routing messages when applied in routing messages filtering.

An ip-prefix is identified by prefix list name. Each prefix list may contain multiple items, each of which specifies a matching range of a network prefix type and identifies with a sequence-number which specifies the matching check order of ip-prefix.

In the process of matching, the switch check each items identified by sequence-number in ascending order and the filter will be passed once certain items is matched(without checking rest items)

4. Autonomic system path information access-list as-path

The autonomic system path information access-list as-path is only used in BGP. In the BGP routing messages packet there is an autonomic system path field (in which autonomic system path the routing messages passes through is recorded). As-path is specially for specifying matching conditions for autonomic system path field.

As for relevant as-path configurations, please refer to the ip as-path command in BGP configuration.

5. community-list

Community-list is only for BGP. There is a community property field in the BGP routing messages packet for identifying a community. The community list is for specifying matching conditions for Community-list field.

As for relevant Community-list configuration, please refer to the ip as-path command in BGP configuration

1.2.2 IP Routing Policy Configuration Task List

1、 Define route-map

2、Define the match clause in route-map

3、Define the set clause in route-map

4、Define address prefix list

1.Define route-map

Command	**Explanation**
Global mode	
route-map <*map_name*> {deny | permit} <*sequence_num*> **no route-map <*map_name*> [{deny | permit} <*sequence_num*>]**	Configure route-map; the **no route-map <*map_name*> [{deny | permit} <*sequence_num*>]** command deletes the route-map

2. Define the match clause in route-map

Command	**Explanation**
Route-map configuration mode	
match as-path <*list-name*> **no match as-path [<*list-name*>]**	Match the autonomous system as path access-list the BGP route passes through; the **no match as-path [<*list-name*>]** command deletes match condition
match community <*community-list-name* | *community-list-num* > [exact-match] **no match community [<*community-list-name* | *community-list-num* > [exact-match]]**	Match a community property access-list. The **no match community [<*community-list-name* | *community-list-num* > [exact-match]]** command deletes match condition
match interface <*interface-name* > **no match interface [<*interface-name* >]**	Match by interface; The **no match interface [<*interface-name* >]** command deletes match condition

match ip <address | next-hop> <*ip-acl-name* | *ip-acl-num* | prefix-list *list-name*> **no match ip <address | next-hop> [<*ip-acl-name* | *ip-acl-num* | prefix-list [*list-name*]>]**	Match the address or next-hop; The **no match ip <address | next-hop> [<*ip-acl-name* | *ip-acl-num* | prefix-list [*list-name*]>]** command deletes match condition
match metric <*metric-val* > **no match metric [<*metric-val* >]**	Match the routing metric value; The **no match metric [<*metric-val* >]** command deletes match condition
match origin <egp | igp | incomplete > **no match origin [<egp | igp | incomplete >]**	Match the route origin; The **no match origin [<egp | igp | incomplete >]** command deletes match condition
match route-type external <type-1 | type-2 > **no match route-type external [<type-1 | type-2 >]**	Match the route type; The **no match route-type external [<type-1 | type-2 >]** command deletes match condition
match tag <*tag-val* > **no match tag [<*tag-val* >]**	Match the route tag; The **no match tag [<*tag-val* >]** command deletes match condition

3. Define the match clause in route-map

Command	**Explanation**
Route-map configuration mode	
set aggregator as <*as-number*> <*ip_addr*> **no set aggregator as [<*as-number*> <*ip_addr*>]**	Distribute an AS No. for BGP aggregator; The **no set aggregator as [<*as-number*> <*ip_addr*>]** command deletes the configuration

set as-path prepend <*as-num*> **no set as-path prepend [<*as-num*>]**	Add a specified AS No. before the BGP routing messages as-path series; The **no set as-path prepend [<*as-num*>]** command deletes the configuration
set atomic-aggregate **no set atomic-aggregate**	Configure the BGP atomic aggregate property; The **no set atomic-aggregate** command deletes the configuration
set comm-list <*community-list-nam*e | *community-list-num* > delete **no set comm-list <*community-list-nam*e | *community-list-num* > delete**	Delete BGP community list value; The **no set comm-list <*community-list-nam*e | *community-list-num* > delete** command deletes the configuration
set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive] **no set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]**	Configure BGP community list value; The **no set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]** command deletes the configuration
set extcommunity <rt | soo> <*AA:NN*> **no set extcommunity <rt | soo> [<*AA:NN*>]**	Configure BGP extended community list property; The **no set extcommunity <rt | soo> [<*AA:NN*>]** command deletes the configuration
set ip next-hop <*ip_addr*> **no set ip next-hop [<*ip_addr*>]**	Set next-hop IP address; The **no set ip next-hop [<*ip_addr*>]** command deletes the configuration

set local-preference <*pre_val*> **no set local-preference [<*pre_val*>]**	Set local preference; The **no set local-preference [<*pre_val*>]** command deletes the configuration
set metric < +/- *metric_val* | *metric_val*> **no set metric [+/- *metric_val* | *metric_val*]**	Set routing metric value; The **no set metric [+/- *metric_val* | *metric_val*]** command deletes the configuration
set metric-type <type-1 | type-2> **no set metric-type [<type-1 | type-2>]**	Set OSPF metric type; The **no set metric-type [<type-1 | type-2>]** command deletes the configuration
set origin <egp | igp | incomplete > **no set origin [<egp | igp | incomplete >]**	Set BGP routing origin; The **no set origin [<egp | igp | incomplete >]** command deletes the configuration
set originator-id <*ip_addr*> **no set originator-id [<*ip_addr*>]**	Set routing originator ID; The **no set originator-id [<*ip_addr*>]** command deletes the configuration
set tag <*tag_val*> **no set tag [<*tag_val*>]**	Set OSPF routing tag value; The **no set tag [<*tag_val*>]** command deletes the configuration
set vpnv4 next-hop <*ip_addr*> **no set vpnv4 next-hop [<*ip_addr*>]**	Set BGP VPNv4 next-hop address; the **no set vpnv4 next-hop [<*ip_addr*>]** command deletes the configuration
set weight < *weight_val*> **no set weight [< *weight_val*>]**	Set BGP routing weight; The **no set weight [< *weight_val*>]** command deletes the configuration

4. Define address prefix list

Command	**Explanation**
Global mode	
ip prefix-list <*list_name*> description <*description*> **no ip prefix-list <*list_name*> description**	Describe the prefix list; The **no ip prefix-list <*list_name*> description** command deletes the configuration
ip prefix-list <*list_name*> [seq <*sequence_number*>] <deny | permit> < any | *ip_addr/mask_length* [ge *min_prefix_len*] [le *max_prefix_len*]> **no ip prefix-list <*list_name*> [seq <*sequence_number*>] [<deny | permit> < any | *ip_addr/mask_length* [ge *min_prefix_len*] [le *max_prefix_len*]>]**	Set the prefix list; The **no ip prefix-list <*list_name*> [seq <*sequence_number*>] [<deny | permit> < any | *ip_addr/mask_length* [ge *min_prefix_len*] [le *max_prefix_len*]>]** command deletes the configuration
ip prefix-list sequence-number **no ip prefix-list sequence-number**	Enable the sequence-number auto-creation function, the "**no ip prefix-list sequence-number**" command close the prefix-list sequence-number.

1.2.3 Command For Routing Policy

1.2.3.1 ip prefix-list description

Command: ip prefix-list <*list_name*> description <*description*>
no ip prefix-list <*list_name*> description

Function: Configure the description of the prefix-list. The "**no ip prefix-list <*list_name*> description**" command deletes the description contents.

Parameter: <*list _name*> is the name of the prefix-list, **<*description* >**is the description contents

Default: None.

Command Mode: Global Mode

Usage Guide: This command can be used for explaining and describing a prefix-list, e.g. the application and attention matters of the prefix-list

Example: Switch#config terminal

Switch(config)#ip prefix-list 3 description This list is used by BGP

1.2.3.2 ip prefix-list seq

Command: ip prefix-list <*list_name*> [seq <*sequence_number*>] <deny | permit> < any | *ip_addr/mask_length* [ge <*min_prefix_len*>] [le <*max_prefix_len*>]>

no ip prefix-list <*list_name*> [seq <*sequence_number*>] [<deny | permit> < any | *ip_addr/mask_length* [ge <*min_prefix_len*>] [le <*max_prefix_len*>]>]

Function: Configure the prefix-list. The "**no ip prefix-list <*list_name*> [seq <*sequence_number*>] [<deny | permit> < any | *ip_addr/mask_length* [ge <*min_prefix_len*>] [le <*max_prefix_len*>]>]**" command deletes the prefix-list.

Parameter: ***<list _name>*** is the name of prefix-list, "seq" shows the following parameters is the sequence number, ***<sequence_number>*** is the sequence number, "deny" means deny this route, "permit" means permit this route, "any" means adaptive to all packets with any prefix as well as any mask length, ***ip_addr/mask_length*** shows the prefix address (dotted decimal notation) and the length of mask, "ge" means greater than or equal to, ***<min_prefix_len>*** is the minimum length of prefix to be matched（ranging between 0～32）, "le" means less than or equal to, ***<max_prefix_len>*** is the maximum length of prefix to be matched（ranging between 0～32）.

Default: None.

Command Mode: Global Mode

Usage Guide: A prefix-list is identified by a prefix-list name. Each prefix-list may include several items each of which independently specifies a matching scope of network prefix-list type which is identified with a *sequence-number. sequence-number* specifies the sequence of matching check in the prefix-list. In the matching process the switch check in turn every items identified by "*sequence-number*" ascending. Once certain item obtains the conditions then the prefix-list filter is passed (without proceeding into the next item check)

Attentions should be paid on that at least one item match mode should be "permit" when more than one prefix-list items is defined. The deny mode items can be previously defined so to remove the unsuitable routing messages fast. However if all items are at deny mode then none of the routes would be able to pass the filter of this prefix-list. We here can define a "permit 0.0.0.0/0 ge 0 le 32" item after several defined "deny mode" items so to grant the passage for all other routing messages.

Example: Switch#config terminal

Switch(config)#ip prefix-list mylist seq 12345 deny 10.0.0.0/8 le 22 ge 14

1.2.3.3 ip prefix-list sequence-number

Command: ip prefix-list sequence-number

no ip prefix-list sequence-number

Function: Enable the sequence-number auto-creation function, the “**no ip prefix-list sequence-number**” command close the prefix-list sequence-number.

Parameter：None.

Default: Sequence-number auto-creation enabled.

Command Mode: Global Mode

Usage Guide: The command can be used to close the prefix-list sequence-number.

Example: Close the prefix-list sequence-number.

Switch(config)#no ip prefix-list sequence-number

1.2.3.4 match as-path

Command: match as-path <*list-name*>

no match as-path [<*list-name*>]

Function: Configure the AS path domain for matching the BGP routing messages. The “**no match as-path [<*list-name*>]**” delete this configuration.

Parameter：**<*list-name* >** is the name of access-list.

Command Mode: route-map mode

Usage Guide: This command matches the AS path domain of the BGP routing message following the rules specified in the as-path list. If the matching succeeded, then the “permit” or “deny” action in the route-map is performed.

Example: Switch#config terminal

Switch(config)#route-map r1 permit 5

Switch(config-route-map)#match as-path 60

1.2.3.5 match community

Command: match community <*community-list-nam*e | *community-list-num* > [exact-match]

no match community [<*community-list-nam*e | *community-list-num* > [exact-match]]

Function: Configure the community attributes of BGP routing messages. The “**no match community [<*community-list-nam*e | *community-list-num* > [exact-match]]**” command deletes this configuration.

Parameter: <*community-list-nam*e > is the name of the community-list, **<*community-list-num* >** is the community-list sequence number, ranging between 1～99（Standard ACL）or 100～199（Extended ACL）, **[exact-match]** means precise matching.

Command Mode: route-map mode

Usage Guide: This command matches the community attributes of the BGP routing message following the rules specified in the community list. If the matching succeeded, then the “permit” or “deny” action in the route-map is performed.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#match community 100 exact-match
```

1.2.3.6 match interface

Command: match interface <*interface-name* >

no match interface [<*interface-name* >]

Function: Configure to match the interfaces. The “**no match interface [<*interface-name* >]**“ deletes this configuration.

Parameter: “ ***<interface-name >***“ is the name of the interface.

Command Mode: route-map mode

Usage Guide: This command matches according to the next-hop messages in the route. If the matching succeeded, then the “permit” or “deny” action in the route-map is performed. This command is only used in RIP and OSPF protocols.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#match interface vlan1
```

1.2.3.7 match ip

Command: match ip <address | next-hop> <*ip-ACL-name* | *ip-ACL-num* | prefix-list *list-name*>

no match ip <address | next-hop> [<*ip-ACL-name* | *ip-ACL-num* | prefix-list *list-name*>]

Function: Configure the routing prefix or next-hop. The “**no match ip <address | next-hop> [<*ip-ACL-name* | *ip-ACL-num* | prefix-list *list-name*>]**” deletes this configuration.

Parameter: <address > means matching the routing prefix, **<next-hop>**means matching the routing next-hop, **<*ip-ACL-name* >** is the name of ip access-list, **<*ip-ACL-num* >** is the ip access-list sequence number, ranging between 1～199 or 1300～2699（extension scope）, **prefix-list** means the matching should follow the prefix-list rules, ***list-name*** is the name of prefix-list.

Command Mode: route-map mode

Usage Guide: This command matches according to the next-hop messages or routing prefix in the route. If the matching succeeded, then the “permit” or “deny” action in the route-map is performed.

Example:

Switch#config terminal

Switch(config)#route-map r1 permit 5

Switch(config-route-map)#match ip address prefix-list mylist

1.2.3.8 match metric

Command: match metric <*metric-val* >

no match metric [<*metric-val* >]

Function: Match the metric value in the routing message. The "**no match metric [<*metric-val* >]**" deletes the configuration.

Parameter: <*metric-val* > is the metric value, ranging between 0～4294967295.

Command Mode: route-map mode

Usage Guide: This command matches according to metric value in the route. If the matching succeeded, then the "permit" or "deny" action in the route-map is performed.

Example:

Switch#config terminal

Switch(config)#route-map r1 permit 5

Switch(config-route-map)#match metric 60

1.2.3.9 match origin

Command: match origin <egp | igp | incomplete >

no match origin <egp | igp | incomplete >

Function: Configure to matching with the origin of the BGP routing message. The "**no match origin <egp | igp | incomplete >**" deletes the configuration.

Parameter: egp means the route is learnt from the external gateway protocols, **igp** means the route is learnt from the internal gateway protocols, **incomplete** means the route origin is uncertain.

Command Mode: route-map mode

Usage Guide: This command matches according to origin message in the BGP route. If the matching succeeded, then the "permit" or "deny" action in the route-map is performed.

Example:

Switch#config terminal

Switch(config)#route-map r1 permit 5

Switch(config-route-map)#match origin egp

1.2.3.10 match route-type

Command: match route-type external <type-1 | type-2 >

no match route-type external [<type-1 | type-2 >]

Function: Configure to matching with the route type of OSPF routing message. The "**no match route-type external [<type-1 | type-2 >]**" deletes the configuration.

Parameter: type-1 means match with the OSPF type 1 external route, **type-2** means match with the OSPF type 2 external route.

Command Mode: route-map mode

Usage Guide: This command matches according to the type of OSPF routes （OSPF AS-external LSA type is either type 1 or type 2）. If the matching succeeded, then the "permit" or "deny" action in the route-map is performed.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#match route-type external type-1
```

1.2.3.11 match tag

Command: match tag <*tag-val* >

no match tag [<*tag-val* >]

Function: Configure to matching with the tag domain of the OSPF routing message. The "**no match tag [<*tag-val* >]**" deletes this configuration.

Parameter: <*tag-val* > is the tag value, ranging between 0～4294967295.

Command Mode: route-map mode

Usage Guide: This command matches according to the tag value in the OSPF route. If the matching succeeded, then the "permit" or "deny" action in the route-map is performed.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#match tag 60
```

1.2.3.12 route-map

Command: route-map <*map_name*> {deny | permit} <*sequence_num*>

no route-map <*map_name*> [{deny | permit} <*sequence_num*>]

Function: Configure the route-map and entering the route-map mode. The "**no route-map <*map_name*> [{deny | permit} <*sequence_num*>]**" command deletes route-map.

Parameter: <*map_name*> is the name of route-map, **permit** sets route-map matching mode to permit mode, **deny** sets route-map matching mode to permit mode（**set** sub will not be executed under this mode）, **<*sequence_num*>** is the route-map sequence number, ranging between 1～65535.

Default: None

Command Mode: Global Mode

Usage Guide: A route-map may consist of several nodes each of which is a check unit. The check sequence among nodes is identified by *sequence-number*. "permit" means the node filter will be passed if all match subs are obtained by current route and then further all the set sub of this node will be executed without entering the check in the next node; if the match subs can not be met, the proceed to the check in next node. Relation among different node should be "or", namely one node check passed then the route filter is passed when the switch checks each node in turn in the route-map.

Attentions should be paid on that at least one node match mode should be "permit" when more than one node is defined. When a route-map is used for filtering routing messages, if certain routing message can not pass any node check, then it is considered denied by the route-map. If all nodes in the route-map are set to deny mode, then all routing message should not be able to pass that route-map.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#match as-path 60
Switch(config-route-map)#set weight 30
```

1.2.3.13 set aggregator

Command: set aggregator as <*as-number*> <*ip_addr*>

no set aggregator as [<*as-number*> <*ip_addr*>]

Function: Assign an AS number for BGP aggregator. The "**no set aggregator as [<*as-number*> <*ip_addr*>]**" deletes this configuration.

Parameter: <*as-number* > is the AS number, **<*ip_addr*>** is the ip address of the aggregator shown in decimal notation.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set aggregator as 200 10.1.1.1
```

1.2.3.14 set as-path

Command: set as-path prepend <*as-num*>

no set as-path prepend [<*as-num*>]

Function: Add AS numbers in the AS path domain of the BGP routing message. The "**no set as-path prepend [<*as-num*>]**" command deletes this configuration.

Parameter: ***<as-num >*** is the AS number, circulating inputting several numbers is available.

Command Mode: route-map mode

Usage Guide: To add AS number in the As domain of the BGP, the AS path length should be lengthened so to affect the best neighbor path option. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set as-path prepend 200
```

1.2.3.15 set atomic-aggregate

Command: set atomic-aggregate

no set atomic-aggregate

Function: Configure the atomic aggregate attributes. The "**no set atomic-aggregate**" command deletes this configuration.

Parameter: None

Command Mode: route-map mode

Usage Guide: The BGP informs other BGP speaker by the atomic aggregate attributes. Local system selects a sub-specified route other than the more specified routes included in it. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set atomic-aggregate
```

1.2.3.16 set comm-list

Command: set comm-list <*community-list-name*e | *community-list-num* > delete

no set comm-list <*community-list-name*e | *community-list-num* > delete

Function: Configure to delete the community attributes from the inbound or outbound routing messages. The "**no set comm-list <*community-list-name*e | *community-list-num* > delete**" command deletes the configuration.

Parameter: **<*community-list-name*e >** is the name of community list, **<*community-list-num* >** is the sequence number of community list, ranging between 1～99（standard community list） or 100～199（extended community list）.

Command Mode: route-map mode

Usage Guide: None

Example:

```
Switch#config terminal
```

```
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set comm-list 100 delete
```

1.2.3.17 set community

Command: set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]

no set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]

Function: Configure the community attributes of the BGP routing message. The "**no set community [*AA:NN*] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]**" command deletes this configuration.

Parameter: [*AA:NN*] is the community attribute value, **[internet]** is the internet scope, **[local-AS]** means this route do not announce outside the local AS (but can announce among the sub AS within the confederation), **[no-advertise]** means this route do not send to any neighbor, **[no-export]** means this route do not send to EBGP neighbors, **[none]** means delete the community attributes from the prefix of this route, **[additive]** means add following existing community attributes.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set community local-as additive
```

1.2.3.18 set extcommunity

Command: set extcommunity <rt | soo> <*AA:NN*>

no set extcommunity <rt | soo> [<*AA:NN*>]

Function: Configure the extended community attributes of the BGP routing message. The "**no set extcommunity <rt | soo> [<*AA:NN*>]**" command deletes this configuration.

Parameter: <*rt*> is the route target, **<*soo*>** is the site of origin, **<*AA:NN*>** is the value of community attributes, amongst AA is AS number, NN is a random two byte number.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set extcommunity rt 100:10
```

1.2.3.19 set ip next-hop

Command: set ip next-hop <*ip_addr*>

no set ip next-hop [<*ip_addr*>]

Function: Configure the next-hop of the route. The “**no set ip next-hop [<*ip_addr*>]**” command deletes the configuration.

Parameter: <*ip_addr* > is the ip address of next-hop shown with dotted decimal notation.

Command Mode: route-map mode

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set ip next-hop 10.2.2.2
```

1.2.3.20 set local-preference

Command: set local-preference <*pre_val*>

no set local-preference [<*pre_val*>]

Function: Configure the local priority of BGP route. The “**no set local-preference [<*pre_val*>]**” command deletes this configuration.

Parameter: <*pre_val* > is the value of local priority, ranging between 0～4294967295.

Command Mode: route-map mode

Usage Guide: The local priority attribute is the priority level of a route. A route with a higher local priority level when compared with other route of the same destination, will be more preferred than other route. The local priority validates only within this AS and will not be transported to EBGP neighbors. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set local-preference 60
```

1.2.3.21 set metric

Command: set metric <*metric_val*>

no set metric [<*metric_val*>]

Function: Configure the metric value of the route. The “**no set metric [<*metric_val*>]**” command deletes the configuration.

Parameter: <*metric_val* > is the metric value, ranging between 1～4294967295.

Command Mode: route-map mode

Usage Guide: The metric value only affects the path option from external neighbors to local AS. The less the metric value is the higher is the priority. Under normal circumstances only the path metric value of the neighbors of the same AS will be compared. To extend the comparison to the metric values of different neighbor path, the bgp always-compare-med command should be configured. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set metric 60
```

1.2.3.22 set metric-type

Command: set metric-type <type-1 | type-2>

no set metric-type [<type-1 | type-2>]

Function: Configure the metric type of the OSPF routing message. The "**no set metric-type [<type-1 | type-2>]**" command deletes this configuration.

Parameter: type-1 means matches the OSPF type 1 external route, **type-2** means matches the OSPF type 2 external route.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set metric-type type-1
```

1.2.3.23 set origin

Command: set origin <egp | igp | incomplete >

no set origin [<egp | igp | incomplete >]

Function: Configure the origin code of the BGP routing message. The "**no set origin [<egp | igp | incomplete >]**" command deletes this configuration.

Parameter: egp means the route is learnt from the external gateway protocols, **igp** means the route is learnt from the internal gateway protocols, **incomplete** means the route origin is uncertain.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set origin egp
```

1.2.3.24 set originator-id

Command: set originator-id <*ip_addr*>

no set originator-id [<*ip_addr*>]

Function: Configure the origin ip address of the BGP routing message. The "**no set originator-id [<*ip_addr*>]**" command deletes the configuration.

Parameter: <*ip_addr*> is the ip address of the route source shown by dotted decimal notation.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set originator-id 10.1.1.1
```

1.2.3.25 set tag

Command: set tag <*tag_val*>

no set tag [<*tag_val*>]

Function: Configure the tag domain of OSPF routing messages. The "**no set tag [<*tag_val*>]**" command deletes this configuration.

Parameter: <*tag-val* > is the tag value, ranging between 0～4294967295.

Command Mode: route-map mode

Usage Guide: There is a route-tag domain at the AS-external-LSA type LSA. The domain is normally identified by other routing protocols. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set tag 60
```

1.2.3.26 set vpnv4 next-hop

Command: set vpnv4 next-hop <*ip_addr*>

no set vpnv4 next-hop [<*ip_addr*>]

Function: Configure the next-hop of BGP VPNv4 routing message. The "**no set vpnv4 next-hop [<*ip_addr*>]**" command deletes the configuration.

Parameter: <*ip_addr*> is the next-hop ip address of VPNv4 route shown by dotted decimal notation.

Command Mode: route-map mode

Usage Guide: To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
```

```
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set vpnv4 next-hop 10.1.1.1
```

1.2.3.27 set weight

Command: set weight <*weight_val*>

no set weight [<*weight_val*>]

Function: Configure the weight value of BGP routing message. The "**no set weight [<*weight_val*>]**" command deletes this configuration.

Parameter: <*weight_val*> is weight value, ranging between 0～4294967295

Command Mode: route-map mode

Usage Guide: Weight value is adopted to facilitate the best path option and validates only within the local switch. While there are several route to the same destination the one with higher priority is more preferred. To use this command, one match clause should at first be defined.

Example:

```
Switch#config terminal
Switch(config)#route-map r1 permit 5
Switch(config-route-map)#set weight 60
```

1.2.4 Configuration Examples

The figure below shows a network consisting of four Layer 3 switches. This example demonstrates how to set the BGP as-path properties through route-map. BGP protocol is applied among the Layer 3 switches. As for switchC, the network 192.68.11.0/24 can be reached through two paths in which one is AS-PATH 1 by IBGP (going through SwitchD), the other one is AS-PATH 2 by EBGP (going through SwitchB). BGP selects the shortest path, so AS-PATH 1 is the preferred path. If the path 2 is wished, which is through EBGP path, we can add two extra AS path numbers into the AS-PATH messages from SwitchA to SwitchD so as to change the determination SwitchC take to 192.68.11.0/24.

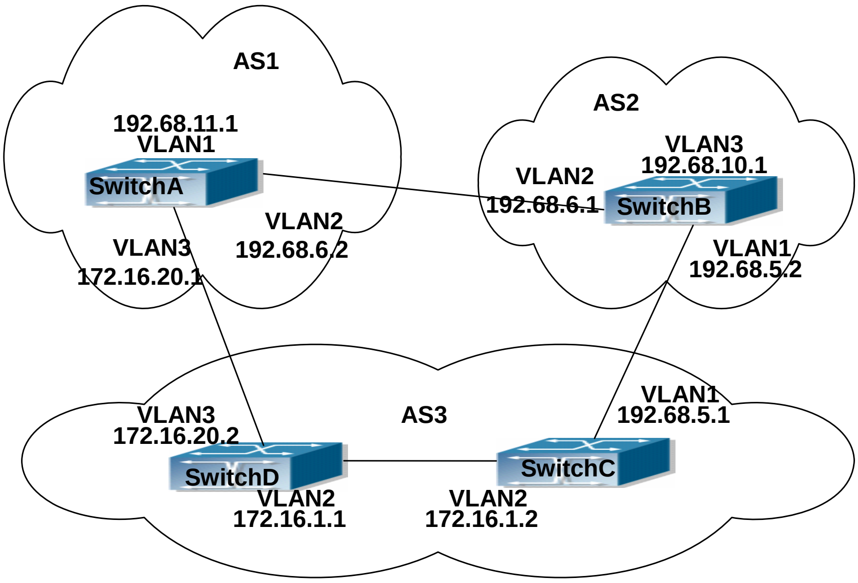

Fig 1-1 Policy routing Configuration

configuration procedure: (only SwitchA is listed,configurations for other switches are omitted.)

The configuration of Layer 3 switchA:

```
SwitchA#config
SwitchA (config) #router bgp 1
SwitchA (config-router)#network 192.68.11.0 mask 255.255.255.0
SwitchA(config-router)#neighbor 172.16.20.2 remote-as 3
SwitchA(config-router)#neighbor 172.16.20.2 route-map AddAsNumbers out
SwitchA(config-router)#neighbor 192.68.6.1 remote-as 2
SwitchA(config-router)#exit
SwitchA(config)#route-map AddAsNumbers permit 10
SwitchA(config-route-map)#set as-path prepend 1 1
```

1.2.5 Troubleshooting Help

Faq: **The routing protocol could not achieve the routing messages study under normal protocol running state**

Troubleshooting: check following errors:

Each node of route-map should at least has one node is permit match mode. When the route map is used in routing messages filtering, the routing messages will be considered not pass the routing messages filtering if certain routing messages does not pass the filtering of any nodes. When all nodes are set to deny mode, all routing messages will not pass the filtering in this route-map.

Items in address prefix list should at least have one item set to permit mode. The deny mode items can be defined first to fast remove the unmatched routing messages, however if all the items are set to deny mode, any route will not be able to pass the filtering of this address prefix list. We can define a permit 0.0.0.0/0 le 32 item after several deny mode items are defined so to permit all other routing messages pass through. Only default route will be matched in less-equal 32 is not specified.

1.2.5.1 Monitor And Debug Command

1.2.5.1.1 show ip prefix-list <list-name>

Command: show ip prefix-list [<*list-name*> [<*ip_addr/len*> [first-match | longer] | seq <*sequence-number*>]]

Function: Show by prefix-list names.

Parameter: <*list-name*> is the name of prefix-list, **<*ip_addr/len*>** is the prefix ip address and the length of mask, **first-match** stands for the first route table matched with specified ip address, **longer** means longer prefix is required, **seq** means show by sequence number,**<*sequence-number*>** is the sequence number, ranging between 0～4294967295.

Default: None

Command Mode: all modes

Usage Guide: All prefix-list will be listed when no prefix-list name is specified.

Example:

```
Switch#show ip prefix-list
ip prefix-list 1: 1 entries
    deny any
ip prefix-list mylist: 1 entries
    deny 1.1.1.1/8
Switch#show ip prefix-list mylist 1.1.1.1/8
    seq 5 deny 1.1.1.1/8 (hit count: 0, recount: 0)
```

Displayed information	Explanation
ip prefix-list mylist: 1 entries	Show a prefix-list named mylist which includes 1 instance.
seq 5 deny 1.1.1.1/8 (hit count: 0, recount: 0)	Show the prefix-list contents sequence numbered 5. hit count: 0 means being hit 0 time, recount: 0 means referred 0 time.

1.2.5.1.2 show ip prefix-list <detail | summary>

Command: show ip prefix-list [<detail | summary> [<*list-name*>]]

Function: Show the prefix-list contents.

Parameter: detail means show detailed messages, **summary** means show summary messages, ***<list-name>*** is the name of prefix-list.

Default: None

Command Mode: all modes

Usage Guide: All prefix-lists will be shown if no prefix-list name is specified.

Example:

```
Switch#show ip prefix-list detail mylist
ip prefix-list mylist:
   count: 2, range entries: 0, sequences: 5 - 10
   seq 5 deny 1.1.1.1/8 (hit count: 0, recount: 0)
   seq 10 permit 2.2.2.2/8 (hit count: 0, recount: 0)
Switch#show ip prefix-list summary mylist
ip prefix-list mylist:
   count: 2, range entries: 0, sequences: 5 - 10
```

Displayed information	Explanation
ip prefix-list mylist:	Show the prefix-list named mylist
count: 2, range entries: 0, sequences: 5 -10	count: 2 means two prefix-list entries, sequences: 5-10 shows the sequence number, 5 is the starting sequence number, 10 is the last sequence number.
deny 1.1.1.1/8 (hit count: 0, recount: 0)	deny 1.1.1.1/8 is the detailed contents in the prefix-list entries, hit count: 0 means being hit 0 times, recount: 0 means being referred 0 times.

1.2.5.1.3 show route-map

Command: show route-map

Function: Show the content of route-map

Parameter: None

Default: None

Command Mode: all modes

Usage Guide: None

Example:

```
Switch# show route-map
route-map a, deny, sequence 10
  Match clauses:
    as-path 60
  Set clauses:
    metric 10
```

Displayed information	Explanation
route-map a, deny, sequence 10	route-map a means the name of route map is a, deny means the deny mode, sequence 10 means the sequence number is 10
Match clauses:	Match sub
as-path 60	Detailed contents in the Match sub
Set clauses:	Set sub
metric 10	Detailed content in the Set clause

1.2.5.1.4 show router-id

Command: show router-id

Function: Show the content of router-id.

Parameter: None

Default: None

Command Mode: Admin and Configuration Mode

Usage Guide: None

Example:

1:

```
Switch#show router-id
Router ID: 20.1.1.1 (automatic)
```

2:

```
Switch#show router-id
Router ID: 20.1.1.2 (config)
```

1.3 Static Route

1.3.1 Introduction to Static Route

As mentioned earlier, the static route is the manually specified path to a network or a host. Static route is simple and consistent, and can prevent illegal route modification, and is convenient for load balance and route backup. However, it also has its own defects. Static route, as its name indicates, is static, it won’t modify the route automatically on network failure, and manual configuration is required on such occasions, therefore it is not suitable for mid and large-scale networks.

Static route is mainly used in the following two conditions: 1) in stable networks to reduce load of route selection and routing data streams. For example, static route can be used in route to STUB network. 2) For route backup, configure static route in the backup line, with a lower priority than the main line.

Static route and dynamic route can coexist; layer3 switch will choose the route with the highest priority according to the priority of routing protocols. At the same time, static route can be introduced (redistribute) in dynamic route, and change the priority of the static route introduced as required.

1.3.2 Introduction to Default Route

Default route is a kind of static route, which is used only when no matching route is found. In the route table, default route is indicated by a destination address of 0.0.0.0 and a network mask of 0.0.0.0, too. If the route table does not have the destination of a package and has no default route configured, the package will be discarded, and an ICMP packet will be sent to the source address indicate the destination address or network is unreachable.

1.3.3 Static Route Configuration Task List

1．Static route configuration
2．Default route configuration

1. static route configuration

1. static route configuration **Command**	**Explanation**
Global mode	
ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} {<*gateway-address*> | <*gateway-interface*>} [<*distance*>] **no ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} [<*gateway-address*> | <*gateway-interface*>] [<*distance*>]**	Set static routing; the **no ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} [<*gateway-address*> | <*gateway-interface*>] [<*distance*>]** command deletes a static route entry

2. VPN configuration

Command	**Explanation**

Global mode	
ip route vrf <*name*> {<*ip-prefix*> <*mask*>|<*ip-prefix/*<*prefix-length*>} {<*gateway-address*>|<*gateway-interface*>} [<*distance*>] **no ip route vrf <*name*> {<*ip-prefix*> <*mask*>|<*ip-prefix/*<*prefix-length*>} [<*gateway-address*>|<*gateway-interface*>] [<*distance*>]**	Configures static routing; the **no ip route vrf <*name*> {<*ip-prefix*> <*mask*>|<*ip-prefix/*<*prefix-length*>} [<*gateway-address*>|<*gateway-interface*>] [<*distance*>]** command deletes a static route entry

1.3.4 Command For Static Route

1.3.4.1 ip route

Command:

ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} {<*gateway-address*> | <*gateway-interface*>} [<*distance*>]

no ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} [<*gateway-address*> | <*gateway-interface*>] [<*distance*>]

Function: Configure the static route. The "**no ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*>/<*prefix-length*>} [<*gateway-address*> | <*gateway-interface*>] [<*distance*>]**" command deletes the static route.

Parameter: The **<*ip-prefix*>** and **<*mask*>** are respectively destination IP address and subnet mask, shown in dotted decimal notation; **<*ip-prefix*>** and **<*prefix-length*>** are respectively the destination IP address and the length of prefix; **<*gateway-address*>** is the next-hop IP address shown in dotted decimal notation; **<*gateway-interface*>** is the next-hop interface, **< *distance* >** is the manage distance of route management, ranging between 1～255.

Default: The management distance of static routing is defaulted at 1.

Command Mode: Global Mode.

Usage Guide: When configuring the next-hop of static routing, both by specifying the next-hop IP address of the route data packet and the exit interface are available.

The default distance values of each route type in the layer 3 switch of our company are listed below:

Route Type	Distance Value
Direct Route	0
Static Route	1
OSPF	110

RIP	120
IBGP	200
EBGP	20

The direct route has the highest priority when each route management distance value remain unchanged and followed by static route, EBGP、OSPF、RIP、IBGP.

Example:

Example 1．Add a static route

```
Switch(config)#ip route 1.1.1.0 255.255.255.0 2.1.1.1
```

Example 2．Add default route

```
Switch(config)#ip route 0.0.0.0 0.0.0.0 2.2.2.1
```

1.3.4.2 show ip route

Command: show ip route [<*destination*>|<*destination* >|<*length*>|connected | static | rip| ospf | bgp | isis| kernel| statistics| database [connected | static | rip| ospf | bgp | isis| kernel] |fib[statistics]]

Function: Show the route table

Parameter: **<*destination*>** is the destination network address; **<*destination* >/<*length*>** is the destination network address plus the length of prefix**;** **connected** is direct route; **static** is static route; **rip** is RIP route; **ospf** is OSPF route; **bgp** is BGP route; **isis** is ISIS route; **kernel** is kernel route; **statistics** shows the number of routes; **database** is route database; **fib** is kernel route table.

Command Mode: all modes

Usage Guide: Show all the contents in the route table including: route type, destination network, mask, next-hop address, interface, etc

Example:

```
Switch#show ip route fib
Codes: C - connected, S - static, R - RIP derived, O - OSPF derived
         A - OSPF ASE, B - BGP derived , D - DVMRP derived

    Destination      Mask             Nexthop          Interface        Preference
C   2.2.2.0          255.255.255.0     0.0.0.0           vlan2             0
C   4.4.4.0          255.255.255.0     0.0.0.0           vlan4             0
S   6.6.6.0          255.255.255.0     9.9.9.9           vlan9             1
R   7.7.7.0          255.255.255.0     8.8.8.8           vlan8            120
```

Displayed information	Explanation
C –connected	Direct route, namely the segment directly connected with the layer 3 switch
S –static	Static route, the route manually configured

	by users
R - RIP derived	RIP route, acquired by layer 3 switch through the RIP protocol.
O - OSPF derived	OSPF route, acquired by layer 3 switch through the OSPF protocol
A- OSPF ASE	Route introduced by OSPF
B- BGP derived	BGP route, acquired by the BGP protocol.
Destination	Target network
Mask	Target network mask
Nexthop	Next-hop IP address
Interface	Next-hop pass-by layer 3 swtich interfaces
Preference	Route priority. If other types of route to the target network exists, the kernel route will only shows those with high priority.

1.3.4.3 show ip route vrf

Command:show ip route vrf <*name*> [connected | static | rip| ospf | bgp | isis| kernel|statistics| database[connected | static | rip| ospf | bgp | isis|kernel]]

show ip route fib vrf <*name*> [default|main|local]

Function: Show the routing table

Parameter: ***<name>*** is the name of VPN route forwarding instances; ***<destination>*** is the destination network address; ***<destination >/<length>***为 is the prefix-length plus destination network address**;** **connected** is direct routed; **static** is static route; **rip** is RIP route; **ospf** is OSPF route; **bgp** is BGP route; **isis** is ISIS route; **kernel** is kernel ; **statistics** shows the number of routes; **database** is the route database; **fib** is the kernel route table.

Command Mode: all modes

Usage Guide: Show VPN route table contents includingL route type, destination network, mask, next-hop, interface, etc.

1.3.4.4 ip route vrf

Command:

ip route vrf <*name*> {<*ip-prefix*> <*mask*>|<*ip-prefix*/<*prefix-length*>} {<*gateway-address*>|<*gateway-interface*>} [<*distance*>]

no ip route vrf <*name*> {<*ip-prefix*> <*mask*>|<*ip-prefix*/<*prefix-length*>} [<*gateway-address*>|<*gateway-interface*>] [<*distance*>]

Function: <*name*> is the name of VPN route forwarding instances; ***<ip-prefix>*** and ***<mask>*** are respectively destination IP address and sub network mask shown in dotted decimal notation;

<ip-prefix> and ***<prefix-length>*** are respectively destination IP address and prefix length; ***<gateway-address>*** is the next-hop IP address show in dotted decimal notation; ***<gateway-interface>*** is the next-hop interface, **<** ***distance*** **>** is the route managing distance value ranging between 1～255.

Default: Default static route managing value is 1.

Command Mode: Global mode

Usage Guide: VPN route forwarding instances have to be successfully configured before using this command.

1.3.5 Configuration Examples

The figure shown below is a simple network consisting of three layer3 switches, the network mask for all switches and PC is 255.255.255.0. PC-A and PC-C are connected via the static route set in SwtichA and SwitchC; PC-C and PC-B are connected via the static route set in SwtichC to SwitchB; PC-B and PC-C is connected via the default route set in SwitchB.

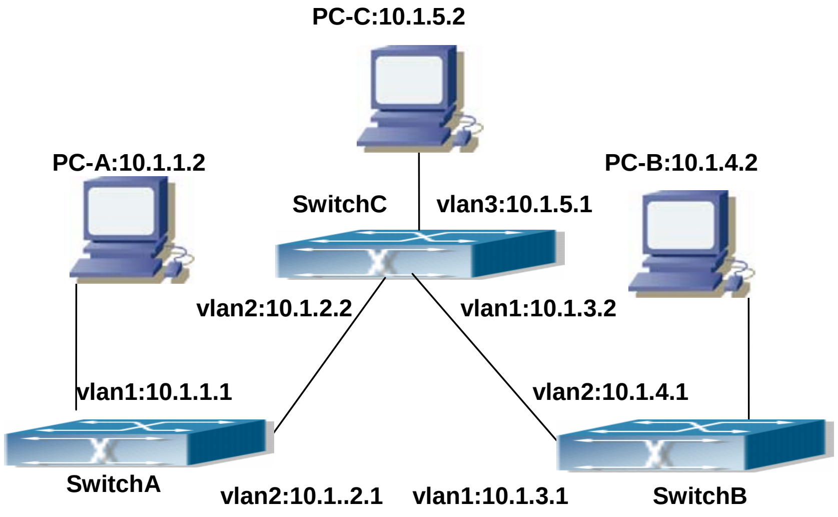

Fig　1-2 Static Route Configurations

Configuration steps:

Configuration of layer3 SwitchA

```
Switch#config
Switch (config) #ip route 10.1.5.0 255.255.255.0 10.1.2.2
```

Configuration of layer3 SwitchC

```
Switch#config
```

Next hop use the partner IP address

```
Switch(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1
```

Next hop use the partner IP address

```
Switch(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1
```

Configuration of layer3 SwitchB

```
Switch#config
Switch(config)#ip route 0.0.0.0 0.0.0.0 10.1.3.2
```

In this way, ping connectivity can be established between PC-A and PC-C, and PC-B and PC-C

1.4 RIP

1.4.1 Introduction to RIP

RIP is first introduced in ARPANET, this is a protocol dedicated to small, simple networks. RIP is a distance vector routing protocol based on the Bellman-Ford algorithm. Network devices running vector routing protocol send 2 kind of information to the neighboring devices regularly:

- Number of hops to reach the destination network, or metrics to use or number of networks to pass.
- What is the next hop, or the director (vector) to use to reach the destination network.

The distance vector Layer 3 switch send all their route selecting tables to the neighbor layer3 switches at regular interval. A layer3 switch will build their own route selecting information table based on the information received from the neighbor layer3 switches. Then, it will send this information to its own neighbor layer3 switches. As a result, the route selection table is built on second hand information, route beyond 15 hops will be deemed as unreachable.

RIP protocol is an optional routing protocol based on UDP. Hosts using RIP send and receive packets on UDP port 520. All layer3 switches running RIP send their route table to all neighbor layer3 switches every 30 seconds for update. If no information from the partner is received in 180 seconds, then the device is deemed to have failed and the network connected to that device is considered to be unreachable. However, the route of that layer3 switch will be kept in the route table for another 120 seconds before deletion.

As layer3 switches running RIP built route table with second hand information, infinite count may occur. For a network running RIP routing protocol, when an RIP route becomes unreachable, the neighboring RIP layer3 switch will not send route update packets at once, instead, it waits until the update interval timeout (every 30 seconds) and sends the update packets containing that route. If before it receives the updated packet, its neighbors send packets containing the information about the failed neighbor, “infinite count” will be resulted. In other words, the route of unreachable layer3 switch will be selected with the metrics increasing progressively. This greatly affects the route selection and route aggregation time.

To avoid “infinite count”, RIP provides mechanism such as “split horizon” and “triggered update” to solve route loop. “Split horizon” is done by avoiding sending to a gateway routes leaned from that gateway. There are two split horizon methods: “simple split horizon” and “poison reverse split horizon”. Simple split horizon deletes from the route to be sent to the neighbor gateways the routes learnt from the neighbor gateways; poison reverse split horizon not only deletes the abovementioned routes, but set the costs of those routes to infinite. “Triggering update” mechanism defines whenever route metric changed by the gateway, the gateway advertise the update packets immediately, regardless of the 30 second update timer status.

There two versions of RIP, version 1 and version 2. RFC1058 introduces RIP-I protocol, RFC2453 introduces RIP-II, which is compatible with RFC1723 and RFC1388. RIP-I updates packets by packets broadcast, subnet mask and authentication is not supported. Some fields in the RIP-I packets are not used and are required to be all 0’s; for this reason, such all 0's fields should be checked when using RIP-I, the RIP-I packets should be discarded if such fields are non-zero. RIP-II is a more improved version than RIP-I. RIP-II sends route update packets by multicast packets (multicast address is 224.0.0.9). Subnet mask field and RIP authentication filed (simple plaintext password and MD5 password authentication are supported), and support variable length subnet mask. RIP-II used some of the zero field of RIP-I and require no zero field verification. ES4700 series layer3 switches send RIP-II packets in multicast by default, both RIP-I and RIP-II packets will be accepted.

Each layer3 switch running RIP has a route database, which contains all route entries for reachable destination, and route table is built based on this database. When a RIP layer3 switch sent route update packets to its neighbor devices, the complete route table is included in the packets. Therefore, in a large network, routing data to be transferred and processed for each layer3 switch is quite large, causing degraded network performance.

Besides the above mentioned, RIP protocol allows route information discovered by the other routing protocols to be introduced to the route table. It can also be as the protocol exchanging route messages with CE on PE routers, and supports the VPN route/transmitting examples.

The operation of RIP protocol is shown below:

Enable RIP. The switch sends request packets to the neighbor layer3 switches by broadcasting; on receiving the request, the neighbor devices reply with the packets containing their local routing information.

The Layer3 switch modifies its local route table on receiving the reply packets and sends triggered update packets to the neighbor devices to advertise route update information. On receiving the triggered update package, the neighbor lay3 switches send triggered update packages to their neighbor lay3 switches. After a sequence of triggered update package broadcast, all layer3 switches get and maintain the latest route information.

In addition, RIP layer3 switches will advertise its local route table to their neighbor devices every 30 seconds. On receiving the packets, neighbor devices maintain their local route table, select the best route and advertise the updated information to their own neighbor devices, so that the updated routes are globally valid. Moreover, RIP uses a timeout mechanism for outdated route, that is, if a switch does not receive regular update packets from a neighbor within a certain interval (invalid timer interval), it considers the route from that neighbor invalid, after holding the route fro a certain interval (holddown timer interval), it will delete that route.

1.4.2 RIP Configuration Task List

1. Enable RIP (required)

   (1) Enable/disable RIP module.

   (2) Enable interface to send/receive RIP packets

2. Configure RIP protocol parameters (optional)

   （1）Configure RIP sending mechanism

   1）Configure specified RIP packets transmission address

   2）Configure RIP interface broadcast

   （2）Configure the RIP routing parameters

   1）Configure route introduction (default route metric, configure routes of the other protocols to be introduced in RIP)

   2）Configure interface authentication mode and password

   3）Configure the route deviation

   4）Configure and apply route filter

   5）Configure Split Horizon

   （3）Configure other RIP protocol parameters

   1）Configure the administratively distance of RIP route

   2）Configure the RIP route capacity limit in route table

   3）Configure the RIP update, timeout, holddown and other timer.

   4）Configure the receiving buffer size of RIP UDP

3. Configure RIP-I/RIP-II switch

   （1）Configure the RIP version to be used in all interfaces

   （2）Configure the RIP version to send/receive in all interfaces

   （3）Configure whether to enable RIP packets sending/receiving for interfaces

4. Delete the specified route in RIP route table
5. Configure the RIP VPN command
6. RIP route aggregation configuration

   （1）To configure aggregation route of ipv4 route mode

   （2）To configure aggregation of ipv4 interface configuration mode

   （3）To display ipv4 aggregation route information

7. Redistribution of OSPF Routing to RIP

   （1）Enable Redistribution of OSPF routing to RIP

（2）Display the information about configuration of redistribution of OSPF routing to RIP

1. Enable RIP protocol

Applying RIP route protocol with basic configuration in ES4700 series is simple. Normally you only have to open the RIP switch and configure the segments running RIP, namely send and receive the RIP data packet by default RIP configuration. The version of data packet sending and receiving is variable when needed, allow/deny sending, receiving RIP data packet. Refer to 3.

Command	Explanation
Global mode	
router rip **no router rip**	Enables RIP; the "**no router rip**" command disables RIP
Router and address family configuration mode	
network <*A.B.C.D/M* | *ifname*> **no network <*A.B.C.D/M* | *ifname*>**	Enables the segment running RIP protocol; the **no network <*A.B.C.D/M* | *ifname*>** command deletes the segment.

2. Configure RIP protocol parameters

（**1**）Configure RIP packet transmitting mechanism

1）Configure the RIP data packet point-transmitting

2）Configure the Rip broadcast

Command	**Explanation**
Router configuration mode	
neighbor <*A.B.C.D*> **no neighbor <*A.B.C.D*>**	Specify the IP address of the neighbor router needs point-transmitting; the **no neighbor <*A.B.C.D*>** command cancels the appointed router.
passive-interface<*ifname*> **no passive-interface<*ifname*>**	Block the RIP broadcast on specified pot and the RIP data packet is only transmittable among Layer 3 switch configured with neighbor. the **no passive-interface<*ifname*>** command cancels the function

（**2**）**Configure RIP route parameters**

1）configure route introduction (default route metric, configure routes of the other protocols to be introduced in RIP)

Command	**Explanation**
Router configuration mode	

default-metric <*value*> **no default-metric**	Sets the default route metric for route to be introduced; the "**no default-metric**" command restores the default setting.
redistribute {kernel |connected| static| ospf| isis| bgp} [metric<value>] [route-map<word>] **no redistribute {kernel |connected| static| ospf| isis| bgp} [metric<value>] [route-map<word>]**	Redistribute the routes distributed in other routing protocols into the RIP data packet; the **no redistribute {kernel |connected| static| ospf| isis| bgp} [metric<value>] [route-map<word>]** command cancels the distributed route of corresponding protocols
default-information originate **no default-information originate**	Generate a default route to the RIP protocol; the **no default-information originate** command cancels the feature.

2）Configure interface authentication mode and password

Command	**Explanation**
Interface configuration mode	
ip rip authentication mode { text| md5} **no ip rip authentication mode [text| md5]**	Sets the authentication method; the **no ip rip authentication mode [text| md5]** command cancels the authentication action
ip rip authentication string <*text*> **no ip rip authentication string**	Sets the authentication key; the **no ip rip authentication string** command means no key is needed.
ip rip authentication key <*name-of-chain*> **no ip rip authentication key** [**<*name-of-chain*>**]	Sets the key chain used in authentication, the **no ip rip authentication key** [**<*name-of-chain*>**] command means the key chain is not used
Global mode	
key chain <*name-of-chain*> **no key chain < *name-of-chain* >**	Enter keychain mode, and configure a key chain, the **no key chain < *name-of-chain* >** command deletes the key chain
Keychain mode	
key <*keyid*> **no key <*keyid*>**	Enter the keychain-key mode and configure a key of the keychain; the **no key <*keyid*>** command deletes one key.
Keychain-key mode	

key-string <*text*> **no key-string <*text*>**	Configure the password used by the key, the **no key-string <*text*>** command deletes the password
accept-lifetime <*start-time*> {<*end-time*>| duration<*seconds*>| infinite} **no accept-lifetime**	Configure a key on the key chain and accept it as an authorized time; the **no accept-lifetime** command delete it
send-lifetime <*start-time*> {<*end-time*>| duration<*seconds*>| infinite} **no send-lifetime**	Configure the transmitting period of a key on the key chain; the **no send-lifetime** command delete the send-lifetime

3）Configure the route deviation

Command	Explanation
Router configuration mode	
offset-list <access-list-number |access-list-name> {in|out }<number >[<ifname>] **no offset-list <access-list-number |access-list-name> {in|out }<number >[<ifname>]**	Configure that provide a deviation value to the route metric value when the port sends or receives RIP data packet; the **no offset-list <access-list-number |access-list-name> {in|out }<number >[<ifname>]** command removes the deviation table

4）configure and apply the route filtering

Command	Explanation
Router configuration mode	
distribute-list {< *access-list-number* |*access-list-name* >|prefix<*prefix-list-name*>}{in|out} [<*ifname*>] **no distribute-list {< *access-list-number* |*access-list-name* >|prefix<*prefix-list-name*>}{in|out} [<*ifname*>]**	Configure and apply the access table and prefix table to filter the routes. the **no distribute-list {< *access-list-number* |*access-list-name* >|prefix<*prefix-list-name*>}{in|out} [<*ifname*>]**command means do not use the access table and prefix table

5）configure the split horizon

Command	**Explanation**
Interface configuration mode	

ip rip split-horizon [poisoned] **no ip rip split-horizon**	Configure that take the split horizon when the port sends data packets; poisoned for poison reverse the **no ip rip split-horizon** command cancels the split horizon

（**3**）Configure other RIP protocol parameters

1）Configure RIP routing priority

2）Configure the RIP route capacity limit in route table

3）Configure timer for RIP update, timeout and hold-down

4）Configure RIP UDP receiving buffer size

Command	**Explanation**
Router configuration mode	
distance <*number*> [<*A.B.C.D/M*>] [<*access-list-name*|*access-list-number* >] **no distance [<*A.B.C.D/M*>]**	Specify the route administratively distance of RIP protocol; the **no distance [<*A.B.C.D/M*>]** command restore the default value 120
maximum-prefix <*maximum-prefix*>[**<*threshold*>**] **no maximum-prefix <*maximum-prefix* >** **no maximum-prefix**	Configure the maximum of RIP route; the **no maximum-prefix <*maximum-prefix* >** **no maximum-prefix** command cancels the limit
timers basic <*update*> <*invalid*> <*garbage*> **no timers basic**	Adjust the update, timeout and garbage collection time, the **no timers basic** command restore the default configuration
recv-buffer-size <*size*> **no recv-buffer-size**	The command configures the UDP receiving buffer size of the RIP; the **no recv-buffer-size** command restore the system default values

3. Configure RIP-I/RIP-II toggling

（**1**）Configure the RIP version to be used in all ports

Command	**Explanation**
RIP configuration mode	
version { 1 | 2 } **no version**	Configure the versions of all the RIP data packets transmitted/received by the Layer 3 switch port sending/receiving the **no version** command restores the default configuration, version 2.

（**2**）Configure the RIP version to send/receive in all ports**.**

（**3**）Configure whether to enable RIP packets sending/receiving for ports

Command	**Explanation**
Interface configuration mode	
ip rip send version { 1 | 1-compatible | 2 } **no ip rip send version**	Sets the version of RIP packets to send on all interfaces; the **no ip rip send version** command set the version to the one configured by the version command
ip rip receive version {1 | 2 | } **no ip rip receive version**	Sets the version of RIP packets to receive on all interfaces; the no action of this command set the version to the one configured by the version command
ip rip receive-packet **no ip rip receive-packet**	Enables receiving RIP packets on the interface; the **no ip rip receive-packet** command close data receiving on this interface
ip rip send-packet **no ip rip send-packet**	Enables sending RIP packets on the interface; the “**no ip rip send-packet**” command disables sending RIP packets on the interface

4. Delete the specified route in RIP route table

Command	**Explanation**
Admin mode	
clear ip rip route {<*A.B.C.D/M*>|kernel|static|connected |rip|ospf|isis|bgp|all}	The command deletes a specified route from the RIP route table

5. Configure the RIP VPN command

Command	Explanation
Router configuration mode	
address-family ipv4 vrf <*vrf-name*> **no address-family ipv4 vrf <*vrf-name*>**	The command configures a RIP address family on the VRF of the PE router. the **no address-family ipv4 vrf <*vrf-name*>** command deletes the configured address family
Address family configuration mode	
exit-address-family	This command exits the address family mode

6. RIP route aggregation configuration

(1) To configure ipv4 aggregation route globally

Command	Explanation
Router Configuration Mode	
ip rip aggregate-address A.B.C.D/M **no ip rip aggregate-address A.B.C.D/M**	To configure or delete ipv4 aggregation route globally.

(2) To configure ipv4 aggregation route on interface

Command	Explanation
Interface Configuration Mode	
ip rip aggregate-address A.B.C.D/M **no ip rip aggregate-address A.B.C.D/M**	To configure or delete ipv4 aggregation route on interface.

(3) To display ipv4 aggregation route information

Command	Explanation
Admin Mode and Configuration Mode	
show ip rip aggregate	To display aggregation route information.

7. Redistribution of OSPF Routing to RIP

(1) To enable Introduction of OSPF Routing for RIP

Command	Notes
Router rip configuration mode	
redistribute ospf [<process-id>] [metric<*value*>] [route-map<*word*>] **no redistribute ospf [<process-id>]**	To enable or disable the redistribution of OSPFv2 routing to RIP.

（**2**）To display the redistribution configuration information

Command	Notes
Admin mode and configuration mode	
show ip rip redistribute	To display RIP routing introduced from other routing protocols.

（**3**）Debugging

Command	Notes
Admin mode	
debug rip redistribute message send **no debug rip redistribute message send**	To enable or disable debugging messages sent by RIP for redistribution of OSPFv2 routing.
debug rip redistribute route receive **no debug rip redistribute route receive**	To enable or disable debugging messages received from NSM.

1.4.3 Command For RIP

1.4.3.1 accept-lifetime

Command: accept-lifetime <*start-time*> {<*end-time*>| duration<*seconds*>| infinite}

no accept-lifetime

Function: Use this command to specify a key accept on the key chain as a valid time period. The "**no accept-lifetime**" command deletes this configuration.

Parameter: <*start-time*> parameter specifies the start time of the time period, of which the form should be:

<start-time>={<hh:mm:ss> <month> <day> <year>|<hh:mm:ss> <day> <month> <year>}

<hh:mm:ss> specify the concrete valid time of **accept-lifetime** in hours, minutes and second

<day> specifies the date of valid, ranging between 1 -31

<month> specifies the month of valid shown with the first three letters of the month, such as Jan

<year> specifies the year of valid start, ranging between 1993 - 2035

<end-time> specifies the due of the time period, of which the form should be:

<end-time>={<hh:mm:ss> <month> <day> <year>|<hh:mm:ss> <day> <month> <year>}

<hh:mm:ss> specify the concrete valid time of **accept-lifetime** in hours, minutes and second

<day> specifies the date of valid, ranging between 1 -31

<month> specifies the month of valid shown with the first three letters of the month, such as Jan

<year> specifies the year of valid start, ranging between 1993 - 2035

<seconds> the valid period of the key in seconds, ranging between 1-2147483646

Infinite means the key will never be out of date.

Default: No default configuration

Command Mode: keychain-key mode

Usage Guide: Refer to the 3.13 RIP authentication Introduction

Example: The example below shows the accept-lifetime configuration of key 1 on the keychain named mychain

```
Switch# config terminal
Switch(config)# key chain mychain
Switch(config-keychain)# key 1
Switch(config-keychain-key)# accept-lifetime 03:03:01 Dec 3 2004 04:04:02 Oct 6 2006
```

1.4.3.2 address-family ipv4

Command: address-family ipv4 vrf <*vrf-name*>

no address-family ipv4 vrf <*vrf-name*>

Function: Configure this command to enable the routing message switching among VRF and enter the address-family mode. The "**no address-family ipv4 vrf <*vrf-name*>**" command deletes the RIP instances related to this VPN routing/forwarding instances

Parameter: <*vrf-name*> specifies the name of VPN routing/forwarding instances

Command Mode: router mode

Usage Guide: This command is only used on PE router. A VPN routing/forwarding instances must be generated with command ip vrf prior to using this command by which the VPN routing/forwarding instances can be related to RIP instances.

Example: Switch# config terminal

```
Switch(config)# router rip
Switch(config-router)# address-family ipv4 vrf VRF1
Switch(config-router-af)#
```

1.4.3.3 clear ip rip route

Command: clear ip rip route {<*A.B.C.D/M*>|kernel|static|connected|rip|ospf|isis|bgp|all}

Function: Clear specific route in the RIP route table

Parameter: Clear the routes which match the destination address from the RIP route table. **<*A.B.C.D/M*>** specifies the IP address prefix and its length of the destination address

kernel delete kernel routes from the RIP route table

static delete static routes from the RIP route table

connected delete direct routes from the RIP route table

rip only delete RIP routes from the RIP route table

ospf only delete OSPF routes from the RIP route table

isis only delete ISIS routes from the RIP route table

bgp only delete BGP routes from the RIP route table

all delete all routes from the RIP route table

Default: No default configurations

Command Mode: Privilleged mode

Usage Guide: Use this command with the all parameter will delete all learnt route in the RIP route which will be immediately recovered except for rip route. The dynamic learnt RIP route can only be recovered by studying one more time.

Example: Switch# clear ip rip route 10.0.0.0/8

Switch# clear ip rip route ospf

1.4.3.4 debug rip

Command: [no] debug rip [events| nsm| packet[recv|send][detail]| all]

Function: Open various RIP adjustment switches and show various adjustment debugging messages. The "**[no] debug rip [events| nsm| packet[recv|send][detail]| all]**" command close corresponding debugging switch.

Parameter : **events** shows the debugging messages of RIP events

nsm shows the communication messages between RIP and NSM.

packet shows the debugging messages of RIP data packets.

recv shows the messages of the received data packets

send shows the messages of the sent data packets

detail shows the messages of received or sent data packets.

Default: Debug switch closed.

Command Mode: Privilleged mode

Example: Switch# debug rip packet

```
Switch#1970/01/01 01:01:43 IMI: SEND[Vlan1]: Send to 224.0.0.9:520
1970/01/01 01:01:43 IMI: SEND[Vlan1]: Send to 224.0.0.9:520
1970/01/01 01:01:47 IMI: RECV[Vlan1]: Receive from 20.1.1.2:520
```

1.4.3.5 default-information originate

Command: default-information originate

no default-information originate

Function: Allow the network 0.0.0.0 to be redistributed into the RIP. The "**no default-information originate**" disable this function.

Parameter: None

Default: Disabled

Command Mode: router mode

Example: Switch# config terminal

```
Switch(config)# router rip
Switch(config-router)# default-information originate
```

1.4.3.6 default-metric

Command: default-metric <*value*>

no default-metric

Function: Set the default metric value of the introduced route. The “**no default-metric**” command restores the default value to 1.

Parameter: <*value*> is the metric value to be set, ranging between 1~16.

Default: Default route metric value is 1

Command Mode: Router mode and address-family mode

Usage Guide: default-metric command is used for setting the default route metric value of the routes from other routing protocols when distributed into the RIP routes. When using the **redistribute** command for introducing routes from other protocols, the default route metric value specified by **default-metric** will be adopted if no specific route metric value is set.

Example: Set the default route metric value to 3 for introducing routes from other routing protocols into the RIP routes.

```
Switch(config-router)#default-metric 3
```

1.4.3.7 distance

Command: distance <*number*> [<*A.B.C.D/M*>] [<*access-list-name*|*access-list-number* >]

no distance [<*A.B.C.D/M*>]

Function: Set the managing distance with this command. The “**no distance [<*A.B.C.D/M*>]**” command restores the default value to 120

Parameter: <*number*> specifies the distance value, ranging between 1-255.**<*A.B.C.D/M*>** specifies the network prefix and its length. **<*access-list-name*|*access-list-number* >** specifies the access-list number or name applied

Default: The default managing distance of RIP is 120

Command Mode: Router mode and address-family mode

Usage Guide: In case there are routes from two different routing protocols to the same destination, the managing distance is then used for selecting routes. The less the managing distance of the route protocol is , the more reliable will be the route acquired from the protocol.

Example: Switch# config terminal

```
Switch(config)# router rip
Switch(config-router)# distance 8 10.0.0.0/8 mylist
```

1.4.3.8 distribute-list

Command: distribute-list{<*access-list-number*|

***access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>]**

no distribute-list{<*access-list-number*|

***access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>]**

Function: This command uses access-list or prefix-list to filter the route update packets sent and received. The "**no distribute-list{<*access-list-number*| *access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>]**" command cancels this route filter function.

Parameter: ***<access-list-number* |*access-list-name*>** is the name or access-list number to be applied. ***<prefix-list-name>*** is the name of the prefix-list to be applied. ***<ifname>*** specifies the name of interface to be applied with route filtering.

Default: The function in default situation is disabled.

Command Mode: Router mode and address-family mode

Usage Guide: The filter will be applied to all the interfaces in case no specific interface is set.

Example: Switch# config terminal

```
Switch(config)# router rip
Switch(config-router)# distribute-list prefix myfilter in vlan 1
```

1.4.3.9 exit-address-family

Command: exit-address-family

Function: Exit address-family mode

Command Mode: address-family mode

Example:

```
Switch(config)# router rip
Switch(config-router)# address-family ipv4 vrf IPI
Switch(config-router-af)# exit-address-family
Switch(config-router)#
```

1.4.3.10 ip rip aggregate-address

Command: ip rip aggregate-address A.B.C.D/M

no ip rip aggregate-address A.B.C.D/M

Function: To configure RIP aggregation route. The no form of this command will delete this configuration.

Parameter: A.B.C.D/M: ipv4 address and mask length.

Command Mode: Router Mode or Interface Configuration Mode.

Default: Disabled.

Usage Guide: If to configure aggregation route under router mode, RIP protocol must be enabled. If configured under interface configuration mode, RIP protocol may not be enabled, but the aggregation router can operation after the RIP protocol be enabled on interface.

Example: To configure aggregation route as 192.168.20.0/22 globally.

```
Switch(config)#router rip
Switch(config-router)#ip rip agg 192.168.20.0/22
```

1.4.3.11 ip rip authentication key

Command: ip rip authentication key <*name-of-chain*>
no ip rip authentication key

Function: Use this command to enable RIPV2 authentication on an interface and further configures the adopted key chain. The "**no ip rip authentication key**" command cancels the authentication.

Parameter: <*name-of-chain*> is the name of the adopted key chain. There may be spaces in the string. The input ends with an enter and the string should not be longer than 256 bytes

Default: Not configured

Command Mode: Interface Mode

Usage Guide: If the authentication is only configured without configuring the key chain or password used by the interface, the authentication do no effect. If mode has not been configured prior to configuring this command, the mode will be set to plaintext authentication. The "no ip rip authentication key" command will cancel the authentication which only cancels the authentication process when sending or receiving data packet other than set non authentication mode.

Example: Switch# config terminal

```
Switch(config)# interface vlan 1
Switch(Config-if-Vlan1)# ip rip authentication key my key
```

1.4.3.12 ip rip authentication mode

Command: ip rip authentication mode {text|md5}
no ip rip authentication mode {ext|md5}

Function: Configure the authentication mode; the "**no ip rip authentication mode {ext|md5}**" command restores to the default authentication mode namely text authentication mode.

Parameter: text means text authentication; **md5** means MD5 authentication.

Default: Not configured authentication

Command Mode: Interface Mode

Usage Guide: RIP-I do not support authentication which the RIP-II supports two authentication modes: text authentication (i.e. Simple authentication) and data packet authentication (i.e. MD5 authentication). This command should be used associating the ip rip authentication key or ip rip authentication string. Independently configuration will not lead to authentication process.

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip authentication mode md5

1.4.3.13 ip rip authentication string

Command: ip rip authentication string <*text*>

no ip rip authentication string

Function: Set the password used in RIP authentication. The "**no ip rip authentication string**" cancels the authentication

Parameter: <*text*> is the password used in authentication of which the length should be 1-16 characters with space available. The password should end with enter

Command Mode: Interface mode

Usage Guide: The ip rip authentication key will not be able to be configured when this command is configured, key id value is required in MD5 authentication which is 1 when use this command. The mode will be set to plaintext authentication in case no mode configuration is available. The "**no ip rip authentication string**" command will cancel the authentication which only cancels the authentication process when sending or receiving data packet other than set non authentication mode. Input ip rip authentication string aaa aaa to set the password as aaa aaa which is 7 characters.

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip authentication string guest

1.4.3.14 ip rip authentication cisco-compatible

Command: ip rip authentication cisco-compatible

no ip rip authentication cisco-compatible

Function: After configured this command, the cisco RIP packets will be receivable by configuring the plaintext authentication or MD5 authentication.

Parameter: None

Default: Not configured

Command Mode: Interface mode

Usage Guide: After authentication is configured on the cisco router, the RIP packets will exceeds the length of the defined standard length of the protocol once the number of route items is greater than 25. By configuring this command the over-lengthen RIP packets will be receivable other than denied.

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip authentication cisco-compatible

1.4.3.15 ip rip receive-packet

Command: ip rip receive-packet

no ip rip receive-packet

Function: Set the interface to be able to receivable RIP packets; the “**no ip rip receive-packet**” command set the interface to be unable to receivable RIP packets

Default: Interface receives RIP packets

Command Mode: Interface Mode

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip receive-packet

1.4.3.16 ip rip receive version

Command: ip rip receive version { 1 | 2|1 2 }

no ip rip receive version

Function: Set the version information of the RIP packets the interface receives. The default version is 2; the “**no ip rip receive version**” command restores the value set by using the version command.

Parameter: 1 and 2 respectively stands for RIP version 1 and RIP version 2, 1 2 stands for the RIP versions 1, 2.

Default: Version 2

Command Mode: Interface Mode

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip receive version 1 2

1.4.3.17 ip rip send-packet

Command: ip rip send-packet

no ip rip send-packet

Function: Set the Interface to be able to receive the RIP packets; the “**no ip rip send-packet**” set the interface to be unable to receive the RIP packets.

Default: Interface sends RIP packets

Command Mode: Interface Mode

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip send-packet

1.4.3.18 ip rip send version

Command: ip rip send version { 1 | 2 | 1-compatible | 1 2}

no ip rip send version

Function: Set the version information of the RIP packets the interface receives. The default version is 2; the “**no ip rip send version**” command restores the value set by using the version command.

Parameter: 1 and 2 respectively stands for RIP version 1 and RIP version 2, 1 2 stands for the RIP versions 1, 2.

Default: Version 2

Command Mode: Interface Mode

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip send version 1

1.4.3.19 ip rip split-horizon

Command: ip rip split-horizon [poisoned]

no ip rip split-horizon

Function: Enable split horizon. The “**no ip rip split-horizon**” disables the split horizon.

Parameter: [poisoned] means configure the split horizon with poison reverse.

Default: Split Horizon with poison reverse by default

Command Mode: Interface Mode

Usage Guide: The split horizon is for preventing the Routing Loops, namely preventing the layer 3 switches from broadcasting the routes which is learnt from the same interface on which the route to be broadcasted

Example: Switch# config terminal

Switch(config)# interface vlan 1

Switch(Config-if-Vlan1)# ip rip split-horizon poisoned

1.4.3.20 key

Command: key <*keyid*>

no key <*keyid*>

Function: This command is for managing and adding keys in the key chain. The “**no key <*keyid*>**“ command deletes one key.

Parameter: <*keyid*> is key ID, ranging between 0-2147483647.

Command Mode: Keychain mode

Usage Guide: The command permits entering the keychain-key mode and set the passwords corresponding to the keys.

Example: Switch#config terminal

Switch(config)#key chain mychain

Switch(config-keychain)#key 1

Switch(config-keychain-key)#

1.4.3.21 key chain

Command: key chain <*name-of-chain*>

no key chain < *name-of-chain* >

Function: This command is for entering a keychain manage mode and configure a keychain. The “**no key chain < *name-of-chain* >**“ delete one keychain.

Parameter: <*name-of-chain*> is the name string of the keychain the length of which is not specifically limited.

Command Mode: Global Mode

Example: Switch#config terminal

Switch(config)#key chain mychain

Switch(config-keychain)#

1.4.3.22 key-string

Command: key-string <*text*>

no key-string <*text*>

Function: Configure a password corresponding to a key. The “**no key-string <*text*>**“ command delete the corresponding password.

Parameter: <*text*> is a character string without length limit. However when referred by RIP authentication only the first 16 characters will be used.

Command Mode: Keychain-key mode

Usage Guide: This command is for configure different passwords for keys with different ID.

Example: Switch# config terminal

Switch(config)# key chain mychain

Switch(config-keychain)# key 1

Switch(config-keychain-key)# key-string prime

1.4.3.23 maximum-prefix

Command: maximum-prefix <*maximum-prefix*>[**<*threshold*>**]

no maximum-prefix

Function: Configure the maximum number of RIP routes in the route table. The “**no maximum-prefix**” command cancels the limit.

Parameter: <*maximum-prefix*> the maximum number of RIP route, ranging between 1-65535;

a warning is given when the number rate of current route exceeds ***<threshold>*** ranging between 1-100, default at 75

Command Mode: router mode

Usage Guide: The maximum RIP routes only limits the number of routes learnt through RIP but not includes direct route or the RIP static route configured by the route command. The base on which the comparison is performed is the number of route marked R in the show ip route database, and also the number of RIP routes displayed in the show ip route statistics command.

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# maximum-prefix 150

1.4.3.24 neighbor

Command: neighbor <*A.B.C.D*>

no neighbor <*A.B.C.D*>

Function: Specify the destination address requires targeted-peer sending. The “**no neighbor <*A.B.C.D*>**“ command cancels the specified address and restores all gateways to trustable.

Parameter: <*A.B.C.D*> is the specified destination address for the sending, shown in dotted decimal notation.

Default: Not sending to any targeted-peer destination address.

Command Mode: Router mode

Usage Guide: When used accompany with passive-interface command it can be configured to only sending routing messages to specific neighbor.

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# neighbor 1.1.1.1

1.4.3.25 network

Command: [no] network <*A.B.C.C/M*|*ifname*>

Function: Configure the RIP protocol network

Parameter: <*A.B.C.C/M*|> is the IP address prefix and its length in the network

<*ifname*> is the name of a interface.

Default: Not running RIP protocol

Command Mode: Router mode and address-family mode.

Usage Guide: Use this command to configure the network for sending or receiving RIP update packets. If the network is not configured, all interfaces of the network will not be able to send or receive data packets.

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# network 10.0.0.0/8

Switch(config-router)# network vlan 1

1.4.3.26 offset-list

Command: offset-list <*access-list-number*

|*access-list-name*> {in|out }<*number* >[<*ifname*>]

no offset-list <*access-list-number*

|*access-list-name*> {in|out }<*number* >[<*ifname*>]

Function: Add an offset value to the metric value of the routes learnt by RIP. The "**no offset-list <*access-list-number***

|*access-list-name*> {in|out }<*number* >[<*ifname*>]" command disables this function

Parameter: < *access-list-number* |*access-list-name*> is the access-list or name to be applied. ***<number >*** is the added offset value, ranging between 0-16; ***<ifname>*** is the specific interface name

Default: Default offset value is the metric value defined by the system

Command Mode: Router mode and address-family mode.

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# offset-list 1 in 5 vlan 1

1.4.3.27 passive-interface

Command: passive-interface <*ifname*>

no passive-interface <*ifname*>

Function: Set the RIP layer 3 switch blocks RIP broadcast on specified interface, on which the RIP data packets will only be sent to layer 3 switches configured with neighbor.

Parameter: <***ifname***> is the name of specific interface.

Default: Not configured

Command Mode: Router mode

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# passive-interface vlan 1

1.4.3.28 recv-buffer-size

Command: recv-buffer-size<*size*>

no recv-buffer-size

Function: This command configures the size of UDP receiving buffer zone of RIP; the "**no recv-buffer-size**" command restores the system default.

Parameter: ***<size>*** is the buffer zone size in bytes, ranging between 8192-2147483647

Default: 8192 bytes

Command Mode: Router mode

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# recv-buffer-size 23456789

1.4.3.29 redistribute

Command:redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]

no redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]

Function: Introduce the routes learnt from other routing protocols into RIP

Parameter: kernel introduce from kernel routes

connected introduce from direct routes

static introduce from static routes

ospf introduce from OSPF routes

isis introduce from ISIS routes

bgp introduce from BGP routes

<value> is the metric value assigned to the introduced route, ranging between 0-16

<word> is the probe pointing to the route map for introducing routes.

Command Mode: **Router mode and address-family mode.**

Usage Guide: Under the address-family mode, the parameter kernel and isis is unavailable

Example: Switch# config terminal

Switch(config)# router rip

Switch(config-router)# redistribute kernel route-map ipi

1.4.3.30 route

Command: route <*A.B.C.D/M*>

no route <*A.B.C.D/M*>

Function: This command configures a static RIP route. The "**no route <*A.B.C.D/M*>**" command deletes this route.

Parameter: Specifies this destination IP address prefix and its length.

Command Mode: Router mode

Usage Guide: The command add a static RIP route, and is mainly used for debugging. Routes configured by this command will not appear in kernel route table but in the RIP route database.

Example: Switch# config terminal

```
Switch(config)# router rip
Switch(config-router)# route 1.0.0.0/8
```

1.4.3.31 router rip

Command: router rip

no router rip

Function: Enable the RIP routing process and enter the RIP mode; the "**no router rip**" command closes the RIP routing protocol

Default: Not running RIP route

Command Mode: Global mode

Usage Guide: This command is the switch for starting the RIP routing protocol which is required to be open before configuring other RIP protocol commands.

Example: Enable the RIP protocol mode

```
Switch(config)#router rip
Switch(config-router)#
```

1.4.3.32 redistribute ospf

Command: redistribute ospf [<*process-id*>] [metric<*value*>] [route-map<*word*>]

no redistribute ospf [<*process-id*>]

Function: To redistribute of OSPF routing learnt from external processes to RIP. The no form command deletes the redistribution of OSPF routing learned from specified process to rip.

Parameter: process-id is ospfv2 process id, if there is no parameter, that means the process by default, range between 1 to 65535.

metric <*value*> is the metric for redistributed routing, range between 0 to 16.

route-map <*word*> is the pointer to the introduced routing map.

Default: Not redistributed by default.

Command Mode: RIP protocol configuration mode.

Usage Guide: None.

Example: To redistribute OSPFv2 routing to rip.

```
Switch(config)#router rip
Switch(config-router)#redistribute ospf 2
```

1.4.3.33 redistribute ospf (vrf)

Command: redistribute ospf [<*process-id*>] [metric<*value*>] [route-map<*word*>]

no redistribute ospf [<*process-id*>]

Function: To introduce the routing information from OSPF to RIP for local vrf. The no form of this command will remove the introduced routing information.

Parameters: process-id is the process id for OSPF routing process. The value is limited between 1 and 65535. If no process-id is appended, the default routing process will be used.

Default: Not redistributed by default.

Command Mode: RIP vrf configuration mode.

Usage Guide:

Example: To redistribute OSPFv2 routing information to RIP in vrf aaa.

Switch(config)#router rip

Switch(config-router)#address-family ipv4 vrf aaa

Switch(config-router-af)#redistribute ospf 2

1.4.3.34 send-lifetime

Command: send-lifetime <*start-time*> {<*end-time*>| duration<*seconds*>| infinite}

no send-lifetime

Function: Use this command to specify a key on the keychain as the time period of sending keys. The "**no send-lifetime**" cancels this configuration.

Parameter: **<*start-time*>** parameter specifies the starting time of the time period, which is :

<start-time>={<hh:mm:ss> <month> <day> <year>|<hh:mm:ss> <day> <month> <year>}

<hh:mm:ss> Specify the concrete valid time of **accept-lifetime** in hours, minutes and second

<day> Specifies the date of valid, ranging between 1 -31

month> Specifies the month of valid shown with the first three letters of the month, such as Jan

<year> Specifies the year of valid start, ranging between 1993 - 2035

<end-time>={<hh:mm:ss> <month> <day> <year>|<hh:mm:ss> <day> <month> <year>}

<end-time> Specifies the due of the time period, of which the form should be:

<end-time>={<hh:mm:ss> <month> <day> <year>|<hh:mm:ss> <day> <month> <year>}

<hh:mm:ss> Specify the concrete valid time of **accept-lifetime** in hours, minutes and second

<day> Specifies the date of valid, ranging between 1 -31

<month> Specifies the month of valid shown with the first three letters of the month, such as Jan

<year> Specifies the year of valid start, ranging between 1993 -2035

<seconds> is the valid period of the key in seconding and ranging between 1-2147483646

Default: No default configuration

Command Mode: Keychain-key mode

Usage Guide: Refer to the 3.13 RIP authentication section.

Example: The example below shows the send-lifetime configuration on the keychain named mychain for key 1.

```
Switch# config terminal
Switch(config)# key chain mychain
Switch(config-keychain)# key 1
Switch(config-keychain-key)# send-lifetime 03:03:01 Dec 3 2004 04:04:02 Oct 6 2006
```

1.4.3.35 timers basic

Command: timers basic *<update> <invalid> <garbage>*

no timers basic

Function: Adjust the RIP timer update, timeout, and garbage collecting time. The "**no timers basic**" command restores each parameters to their default values.

Parameter: ***<update>*** time interval of sending update packet, shown in seconds and ranging between 5-2147483647; ***<invalid>*** time period after which the RIP route is advertised dead, shown in seconds and ranging between 5-2147483647; ***<garbage>*** is the hold time in which the a route remains in the routing table after advertised dead, shown in seconds and ranging between 5-2147483647.

Default: ***<update>*** defaulted at 30; ***<invalid>*** defaulted at 180; ***<garbage>*** defaulted at120

Command Mode: Router mode

Usage Guide: The system is defaulted broadcasting RIPng update packets every 30 seconds; and the route is considered invalid after 180 seconds but still exists for another 120 seconds before it is deleted from the routing table.

Example: Set the RIP update time to 20 seconds and the timeout period to 80 second, the garbage collecting time to 60 seconds.

```
Switch(Config-Router)#timers basic 20 80 60
```

1.4.3.36 version

Command:version {1| 2}

no version

Function: Configure the version of all RIP data packets sent/received by router interfaces: the "**no version**" restores the default configuration

Parameter: 1 is version 1 rip; **2** is version 2 rip

Default: Sent and received data packet is version 2 by default

Command Mode: Router mode and address-family mode

Usage Guide: 1 refers to that each interface of the layer 3 switch only sends/receives the RIP-I data packets. 2 refers to that each interface of the layer 3 switch only sends/receives the RIP-II data packets. The RIP-II data packet is the default version.

Example: Configure the version of all RIP data packets sent/received by router interfaces to version 2.

Switch(config-router)#version 2

1.4.4 RIP Examples

1.4.4.1 Typical RIP Examples

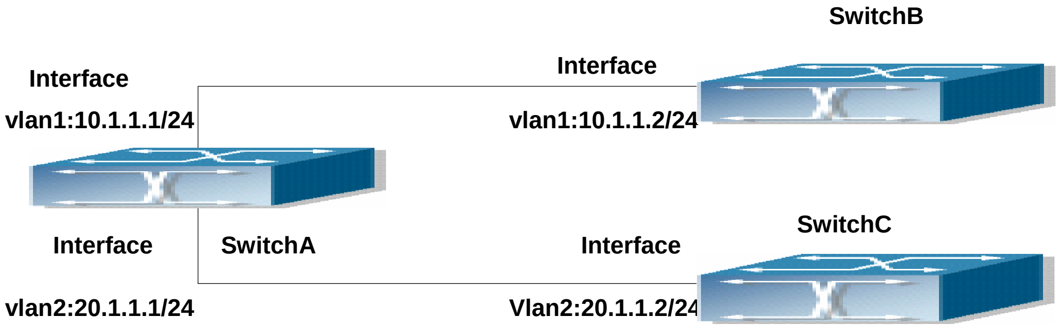

Fig 1-3 RIP example

In the figure shown above, a network consists of three Layer 3 switches, in which SwitchA connected with SwitchB and SwitchC, and RIP routing protocol is running in all of the three switches. SwitchA（interface vlan1:10.1.1.1,interface vlan2:20.1.1.1）exchanges Layer 3 switch update messages only with SwitchB（interface vlan1:10.1.1.2）, but not with SwitchC（interface vlan 2: 20.1.1.2）.

SwitchA, SwitchB, SwitchC configurations are as follows:

a) Layer 3 switch SwitchA:

Configure the IP address of interface vlan 1

```
SwitchA#config
SwitchA(config)# interface vlan 1
SwitchA(Config-if-Vlan1)# ip address 10.1.1.1 255.255.255.0
SwitchA (config-if-Vlan1)#
```

Configure the IP address of interface vlan 2

```
SwitchA (config)# vlan 2
SwitchA (Config-Vlan2)# switchport interface ethernet 1/2
Set the port Ethernet1/2 access vlan 2 successfully
SwitchA (Config-Vlan2)# exit
SwitchA (config)# interface vlan 2
SwitchA (Config-if-Vlan2)# ip address 20.1.1.1 255.255.255.0
```

Initiate RIP protocol and configure the RIP segments

```
SwitchA(config)#router rip
SwitchA(config-router)#network vlan 1
SwitchA(config-router)#network vlan 2
SwitchA(config-router)#exit
```

Configure that the interface vlan 2 do not transmit RIP messages to SwitchC

```
SwitchA(config)#router rip
SwitchA(config-router)#passive-interface vlan 2
SwitchA(config-router)#exit
SwitchA (config) #
```

b) Layer 3 switch SwitchB

Configure the IP address of interface vlan 1

```
SwitchB#config
SwitchB(config)# interface vlan 1
SwitchB(Config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (Config-if-Vlan1)exit
```

Initiate RIP protocol and configure the RIP segments

```
SwitchB(config)#router rip
SwitchB(config-router)#network vlan 1
SwitchB(config-router)#exit
```

c) Layer 3 switch SwitchC

```
SwitchC#config
SwitchC(config)# interface vlan 1
```

Configure the IP address of interface vlan 1

```
SwitchC(Config-if-Vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC (Config-if-Vlan1)#exit
```

Initiate RIP protocol and configure the RIP segments

```
SwitchC(config)#router rip
SwitchC(config-router)#network vlan 1
SwitchC(config-router)#exit
```

1.4.4.2 Configuration Examples of RIP VPN

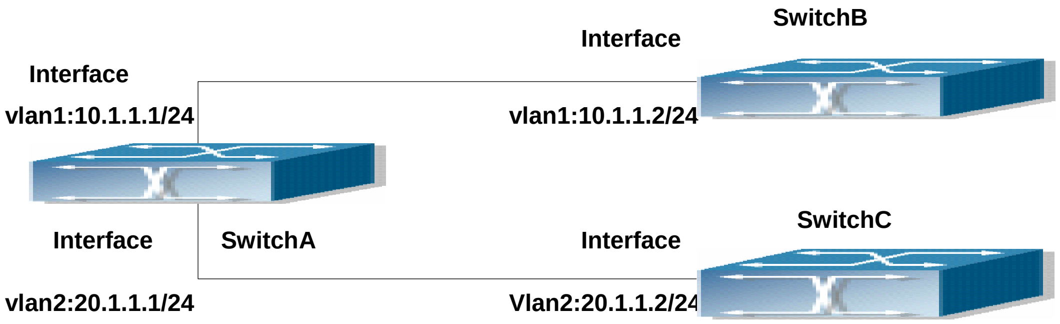

Fig 1-4 RIP VPN example

In the figure shown above, a network consists of three Layer 3 switches, in which the SwitchA as PE, SwitchB and SwitchC as CE1 and CE2. The PE is connected to CE1 and CE2 through vlan 1 and vlan 2. The routing messages are exchanged between PE and CE through RIP protocol.

a) Switch, the Layer 3 switch of PE

Configures the VPN route/transmit example vpnb and vpnc

```
SwitchA#config
SwitchA(config)#ip vrf vpnb
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit
SwitchA#(config)
SwitchA(config)#ip vrf vpnc
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit
```

associate the vlan 1and vlan 2 respectively with vpnb and vpnc while configuring IP address

```
SwitchA(config)#in vlan1
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
SwitchA(config-if-Vlan1)#exit
SwitchA(config)#in vlan2
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
SwitchA(config-if-Vlan2)#exit
```

Configures the RIP examples associated with vpnb and vpnc respectively

```
SwitchA(config)#
SwitchA(config)#router rip
SwitchA(config-router)#address-family ipv4 vrf vpnb
SwitchA(config-router-af)#redistribute bgp
SwitchA(config-router-af)#network Vlan1
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router)#address-family ipv4 vrf vpnc
SwitchA(config-router-af)#redistribute bgp
SwitchA(config-router-af)#network Vlan2
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router)#
```

b) CE1 Layer 3 switch SwitchB

configure the IP address of Ethernet port E 1/2

```
SwitchB#config
SwitchB(config)# interface Vlan1
SwitchB(config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (config-if-Vlan1)exit
```

Initiate RIP protocol and configure the RIP segments

```
SwitchB(config)#router rip
SwitchB(config-router-rip)#network Vlan1
SwitchB(config-router-rip)#exit
```

c) CE2 Layer 3 switch SwitchC

Configure the IP address of Ethernet port E 1/2

```
SwitchC#config
SwitchC(config)# interface Vlan1
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC (config-if-vlan1)#exit
```

Initiate RIP protocol and configure the RIP segments

```
SwitchC(config)#router rip
SwitchC(config-router)#network Vlan1
SwitchC(config-router)#exit
```

1.4.4.3 Typical Examples of RIP aggregation function

The application topology as follows：

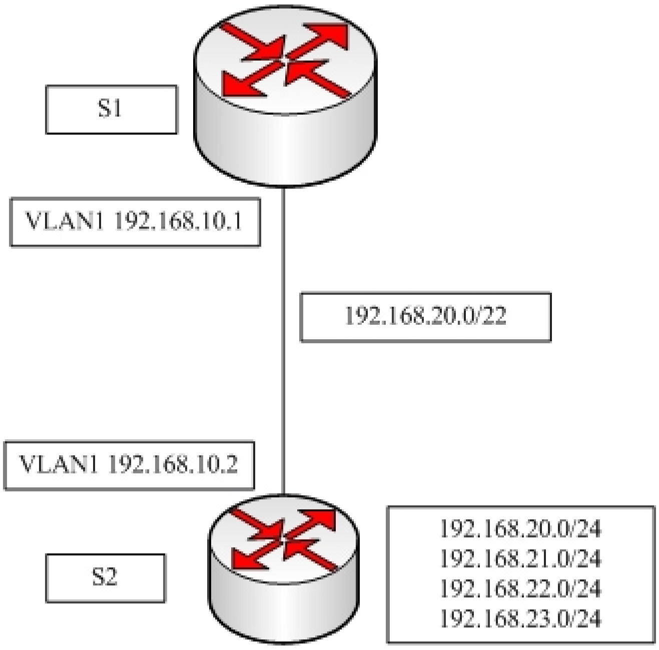

Fig 1-5 Typical application of RIP aggregation

As the above network topology, S2 is connected to S1 through interface vlan1, there are other 4 subnet routers of S2, which are 192.168.21.0/24, 192.168.22.0/24, 192.168.23.0/24, 192.168.24.0/24. S2 supports route aggregation, and to configure aggregation route 192.168.20.0/22 in interface vlan1 of S2, after that, sending router messages to S2 through vlan1, and put the four subnet routers aggregated to one router as 192.168.20.0/22, and send to S1, and not send subnet to neighbor. It can reduce the router table of S1, save the memory.

S1 configuration list：

```
Switch1(config)#router rip
Switch1(config-router) #network vlan 1
```

S2 configuration list：

```
Switch2(config)#router rip
Switch2(config-router)#network vlan 1
Switch2(config-router)#exit
Switch2(config)#in vlan 1
Switch2(Config-if-Vlan1)#ip rip agg 192.168.20.0/22
```

1.4.5 Troubleshooting Help Of RIP

The RIP protocol may not be working properly due to errors such as physical connection, configuration error when configuring and using the RIP protocol. So users should pay attention to following:

First ensure the physic connection is correct

Second, ensure the interface and chain protocol are UP (use show interface command)

Then initiate the RIP protocol (use router rip command) and configure the segment (use network command) and set RIP protocol parameter on corresponding interfaces, such as the option between RIP-I and RIP-II

After that, one feature of RIP protocol should be noticed ---the Layer 3 switch running RIP protocol sending route updating messages to all neighboring Layer 3 switches every 30 seconds. A Layer 3 switch is considered inaccessible if no route updating messages from the switch is received within 180 seconds, then the route to the switch will remains in the route table for 120 seconds before it is deleted. Therefore, if to delete a RIP route, this route item is assured to be deleted from route table after 300 seconds.

When exchanging routing messages with CE using RIP protocol on the PE router, we should first create corresponding VPN routing/transmitting examples to associate with corresponding interfaces. Then enter the RIP address family mode configuring corresponding parameters.

If the RIP routing problem remains unresolved, please use debug rip command to record the debug message in three minutes, and send them to our technical service center.

1.4.5.1 Monitor And Debug Command

1.4.5.1.1 debug rip redistribute message send

Command: debug rip redistribute message send

no debug rip redistribute message send

Function: To enable the debugging of sending messages for routing redistribution messages from OSPF or other external process for RIP. The no form of this command will disable the debugging messages.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug rip redistribute message send
Switch#no debug rip redistribute message send
```

1.4.5.1.2 debug rip redistribute route receive

Command: debug rip redistribute route receive

no debug rip redistribute route receive

Function: To enable debugging of received messages from NSM for RIP. The no form of this command will disable debugging of received messages from NSM for RIP.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug rip redistribute route receive
Switch#no debug rip redistribute route receive
```

1.4.5.1.3 show debugging rip

Command: show debugging rip

Function: Show RIP event debugging, RIP packet debugging and RIP nsm debugging status

Command Mode: Any mode

Example: Switch# show debugging rip

```
RIP debugging status:
RIP event debugging is on
RIP packet detail debugging is on
RIP NSM debugging is on
```

1.4.5.1.4 show ip protocols rip

Command: show ip protocols rip

Function: Show the RIP process parameter and statistics information

Command Mode: Any mode

Example:

```
show ip protocols rip
Routing Protocol is "rip"
Sending updates every 30 seconds with +/-50%, next due in 8 seconds
Timeout after 180 seconds, garbage collect after 120 seconds
Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing: static
  Default version control: send version 2, receive version 2
    Interface          Send   Recv    Key-chain
    Vlan1      2       2
  Routing for Networks:
    Vlan1
    Vlan2
  Routing Information Sources:
    Gateway             Distance   Last Update   Bad Packets   Bad Routes
    20.1.1.1                 120   00:00:31                0             0
  Distance: (default is 120)
```

Displayed information	Explanation
Sending updates every 30 seconds with +/-50%, next due in 8 seconds	Sending update every 30 secs
Timeout after 180 seconds, garbage collect after 120 seconds	The route time-out event period is 180 secs, the garbage collect time is 120 seconds
Outgoing update filter list for all interface is not set	Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set	Incoming update filter list for all interface is not set
Default redistribution metric is 1	Default redistribution metric is 1
Redistributing: static	Redistributing the static route into the RIP route
Default version control: send version 2, receive version 2 Interface Send Recv Key-chain Ethernet1/8 2 2	The configuration of interface receiving and sending packets. Receive version is 2, keychain 1 not configured.
Routing for Networks: Vlan1 Vlan2	The segment running RIP is the Vlan 1 and Vlan 2
Routing Information Sources: Gateway Distance Last Update Bad Packets Bad Routes 20.1.1.1 120 00:00:31 0 0	Routing information sources The badpacketand bad routes from the gateway 20.1.1.1 are all 0. 31 seconds have passed since the last route update. The manage distance is 120
Distance: (default is 120)	Default manage distance is 120

1.4.5.1.5 show ip rip

Command: show ip rip

Function: Show the routes in the RIP route data base

Command Mode: Any mode

Example:

```
show ip rip
Codes: R - RIP, K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS,
        B - BGP

   Network          Next Hop        Metric From          If      Time
R  12.1.1.0/24      20.1.1.1             2 20.1.1.1      Vlan1 02:51
R  20.1.1.0/24                           1               Vlan1
```

Amongst R stands for RIP route, namely a RIP route with the destination network address 12.1.1.0, the network prefix length as 24, next-hop address at 20.1.1.1. It is learnt from the Ethernet port E1/8 with a metric value of 2, and still has 2 minutes 51 seconds before time out.

1.4.5.1.6 show ip rip aggregate

Command: show ip rip aggregate

Function: To display the information of ipv4 aggregation route.

Parameter: None.

Command Mode: Admin Mode and Configuration Mode.

Default: None.

Usage Guide: This command is used to display which interface the aggregation route be configured, Metric, Count, Suppress and so on. If configured under global mode, then the interface display "----", "Metric" is metric. "Count" is the number of learned aggregation routes. "Suppress" is the times of aggregation.

Example: To display the information of ipv4 aggregation route.

```
Switch(Config-if-Vlan1)#show ip rip aggregate

 Aggregate information of rip

        Network             Aggregated Ifname      Metric Count Suppress
        192.168.0.0/16        Vlan1                   1      2      0
        192.168.4.0/22         ----                   1      2      0
        192.168.4.0/24         ----                   1      1      1
                              Vlan1                   1      1      1
```

Displayed information	Notes
Network	Route prefix and prefix length

Aggregated Ifname	To configure the interface name of the aggregation route. If the route aggregated globally , then display “---”.
Metric	Metric of aggregation route
Count	The number of learned aggregation routes
Suppress	The times of aggregated for aggregation route.

1.4.5.1.7 show ip rip database

Command: show ip rip database

Function: Show the routes in the RIP route database

Command Mode: Any mode

Example: Switch# show ip rip database

```
Codes: R - RIP, K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS,
          B -BGP
     Network              Next Hop           Metric From              If       Time
R   10.1.1.0/24                                  1                    Vlan1
R   20.1.1.0/24                                  1                    Vlan2
```

1.4.5.1.8 show ip rip database vrf

Command: show ip rip database vrf <*vrf-name*>

Function: This command display the RIP database messages related to the VPN routing/forwarding instances.

Parameter: Specifies the name of VPN routing/forwarding instances.

Command Mode: Any mode

Example: Switch# show ip rip database vrf IPI

```
Codes: R - RIP, K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS,
          B - BGP

     Network              Next Hop           Metric From              If       Time
R   10.1.1.0/24                                1                    Vlan1     00:46
```

1.4.5.1.9 show ip rip interface

Command: show ip rip interface [**<*ifname*>**]

Function: Show the RIP related messages

Parameter: **<*ifname*>** is the name of the interface to show the messages

Command Mode: Any mode

Example: Switch# show ip rip interface vlan 1

```
Vlan1 is up, line protocol is up
```

```
 Routing Protocol: RIP
   Receive RIP packets
   Send RIP packets
   Passive interface: Disabled
   Split horizon: Enabled with Poisoned Reversed
   IP interface address:
     10.1.1.1/24
```

1.4.5.1.10 show ip rip interface vrf

Command: show ip rip interface vrf <*vrf-name*>[<*ifname*>]

Function: This command shows RIP interface relevant to VPN routing/forwarding instances

Parameter: Specifies the name of VPN routing/forwarding instances

<ifname> is the name of the interfaces

Command Mode: Any mode

Example: Switch# show ip rip interface vrf IPI Vlan1

```
Ethernet1/1 is up, line protocol is up
  Routing Protocol: RIP
    VPN Routing/Forwarding: vpnb
    Receive RIP packets
    Send RIP packets
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      11.1.1.1/24
```

Displayed information	Explanations
Vlan1 is up, line protocol is up	Interface is up
Routing Protocol: RIP	The protocol running on the interface is RIP
VPN Routing/Forwarding: vpnb	Interface relates to the VPN routing/forwarding instances.
Receive RIP packets	The interface can receive RIP packets
Send RIP packets	The interface can send RIP packets
Passive interface: Disabled	Passive-interface disabled
Split horizon: Enabled with Poisoned Reversed	Configure a split horizon with poison reversed
IP interface address: 11.1.1.1/24	The IP address of the interface.

1.4.5.1.11 show ip rip redistribute

Command: show ip rip redistribute [vrf <*NAME*>]

Function: To display the routing information introduced from external process of RIP. **Parameters:** vrf name. If no parameter is appended, all the routing redistribution information of RIP for all vrf.

Default: Not shown by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip rip redistribute
```

1.4.5.1.12 show ip vrf

Command: show ip vrf [<*vrf-name*>]

Function: This command shows the RIP instances messages related to the VPN routing/forwarding instances

Parameter: Specifies the name of the VPN routing/forwarding instances

Command Mode: Any mode

Usage Guide: The command also exist in other routing protocols, when using this command, messages of other routing protocol processes related to this VPN routing/forwarding instances will also be displayed

Example: Switch# show ip vrf IPI

```
VRF IPI, FIB ID 1
Router ID: 11.1.1.1 (automatic)
Interfaces:
   Vlan1
!
VRF IPI; (id=1); RIP enabled Interfaces:
Ethernet1/8

Name                  Interfaces
IPI                     Vlan1

Name                          Default RD              Interfaces
IPI                                                    Vlan1
```

1.5 RIPng

1.5.1 Introduction to RIPng

RIPng is first introduced in ARPANET, this is a protocol dedicated to small, simple networks. RIPng is a distance vector routing protocol based on the Bellman-Ford algorithm. Network devices running vector routing protocol send 2 kind of information to the neighboring devices regularly:

- Number of hops to reach the destination network, or metrics to use or number of networks to pass.
- What is the next hop, or the director (vector) to use to reach the destination network.

Distance vector layer3 switches send all their route selecting tables to the neighbor layer3 switches at regular interval. A layer3 switch will build their own route selecting information table based on the information received from the neighbor layer3 switches. Then, it will send this information to its own neighbor layer3 switches. As a result, the route selection table is built on second hand information, route beyond 15 hops will be deemed as unreachable.

RIPng is an optional routing protocol based on UDP. Hosts using RIPng send and receive packets on UDP port 521. All layer3 switches running RIPng send their route table to all neighbor layer3 switches every 30 seconds for update.If no information from the partner is received in 180 seconds, then the device is deemed to have failed and the network connected to that device is considered to be unreachable. However, the route of that layer3 switch will be kept in the route table for another 120 seconds before deletion.

As layer3 switches running RIPng build route table with second hand information, infinite count may occur. For a network running RIPng routing protocol, when a RIPng route becomes unreachable, the neighboring RIPng layer3 switch will not send route update packets at once, instead, it waits until the update interval timeout (every 30 seconds) and sends the update packets containing that route. If before it receives the updated packet, its neighbors send packets containing the information about the failed neighbor, “infinite count” will be resulted. In other words, the route of unreachable layer3 switch will be selected with the metrics increasing progressively. This greatly affects the route selection and route aggregation time.

To avoid “infinite count”, RIPng provides mechanism such as “split horizon” and “triggered update” to solve route loop. “Split horizon” is done by avoiding sending to a gateway routes leaned from that gateway. There are two split horizon methods: “simple split horizon” and “poison reverse split horizon”. Simple split horizon deletes from the route to be sent to the neighbor gateways the routes learned from the neighbor gateways; poison reverse split horizon not only deletes the abovementioned routes, but set the costs of those routes to infinite. “Triggering update” mechanism defines whenever route metric changed by the gateway, the gateway advertise the update packets immediately other than wait for the 30 sec timer.

So far the RIPng protocol has got only one version----Version1: RIPng protocol is introduced in RFC 2080. RIPng transmits updating data packet by multicast data packet (multicast address FF02::9)

Each layer3 switch running RIPng has a route database, which contains all route entries for reachable destination, and route table is built based on this database. When a RIPng layer3 switch sent route update packets to its neighbor devices, the complete route table is included in the packets. Therefore, in a large network, routing data to be transferred and processed for each layer3 switch is quite large, causing degraded network performance.

Besides the above mentioned, RIPng protocol allows IPv6 route information discovered by the other routing protocols to be introduced to the route table.

The operation of RIPng protocol is shown below:

Enable RIPng The switch sends request packets to the neighbor layer3 switches by broadcasting; on receiving the request, the neighbor devices reply with the packets containing their local routing information.

The Layer3 switch modifies its local route table on receiving the reply packets and sends triggered update packets to the neighbor devices to advertise route update information. On receiving the triggered update package, the neighbor lay3 switches send triggered update packages to their neighbor lay3 switches. After a sequence of triggered update package broadcast, all layer3 switches get and maintain the latest route information.

In addition, RIPng layer3 switches will advertise its local route table to their neighbor devices every 30 seconds. On receiving the packets, neighbor devices maintain their local route table, select the best route and advertise the updated information to their own neighbor devices, so that the updated routes are globally valid. Moreover, RIPng uses a timeout mechanism for outdated route, that is, if a switch does not receive regular update packets from a neighbor within a certain interval (invalid timer interval), it considers the route from that neighbor invalid, after holding the route fro a certain interval (garbage collect timer interval), it will delete that route.

As a result of continuous development of IPv6 network, it has the network environment of nonsupport IPv6 sometimes, so it needs to do the IPv6 operation by tunnel. Therefore, our RIPng supports configuration on configure tunnel, and passes through nonsupport IPv6 network by unicast packet of IPv4 encapsulation.

1.5.2 RIPng Configuration Task List

1. Enable RIPng protocol (required)

（1） Enable/disable RIPng protocol

（2） Configure the interfaces running RIPng protocol

2. Configure RIPng protocol parameters (optional)

（1） Configure RIPng sending mechanism

1) Configure specified RIPng packets transmission address

（2） Configure RIPng routing parameters

1) Configure route introduction (default route metric,configure routes of the other protocols to be introduced in RIPng)

2) Configure the route deviation

3) Configure and apply route filter

4) Configure split horizon

（3） Configure other RIPng parameters

1） Configure timer for RIPng update, timeout and hold-down

（4） Delete the specified route in RIPng route table

3. RIPng aggregation configuration task list

(1) To configure aggregation route of ipv6 route mode

(2) To configure aggregation of ipv6 interface configuration mode

(3) To display ipv6 aggregation route information

4. Redistribution of OSPFv3 Routing to RIPng

（1）Enable Redistribution of OSPFv3 routing to RIPng

（2）Display the information about configuration of redistribution of OSPFv3 routing to RIPng

1. Enable RIPng protocol

Applying RIPng route protocol with basic configuration in ES4700 series is simple. Normally you only have to open the RIPng switch and configure the segments running RIPng, namely send and receive the RIPng data packet by default RIPng configuration.

Command	**Explanation**
Global mode	
[no] router IPv6 rip	Enables the RIPng protocol; the [no] router IPv6 rip command shuts the RIPng protocol.
Interface configuration mode	
[no] IPv6 router rip	configure the interface to run RIPng protocol; the **[no] IPv6 router rip** command set the interface not run RIPng protocol

2. Configure RIPng protocol parameters

（1）Configure RIPng sending mechanism

1）configure the RIPng data packets point-transmitting

Command	**Explanation**
Router configuration mode	
[no] neighbor *<IPv6-address> <ifname>*	Specify the IPv6 Link-local address and interface of the neighboring route needs point-transmitting; the **[no] neighbor *<IPv6-address> <ifname>*** command cancels the appointed router.

[no] passive-interface <*ifname*>	Block the RIPng multicast on specified port and the RIPng data packet is only transmittable among Layer 3 switch configured with neighbor. the **[no] passive-interface <*ifname*>** command cancels the function

（2）Configure RIPng routing parameters

1）configure route introduction (default route metric, configure routes of the other protocols to be introduced in RIPng)

Command	**Explanation**
Router configuration mode	
default-metric <*value*> **no default-metric**	Configure the default metric of distributed route; the **default-metric <*value*>** no default-metric command restores the default configuration 1
[no]redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]	Redistribute the routes distributed in other route protocols into the RIPng data packet; the **[no]redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]** command cancels the distributed route of corresponding protocols
[no]default-information originate	Generate a default route to the RIPng protocol; the **[no]default-information originate** command cancels the feature.

2）Configure the route offset

Command	**Explanation**
Router configuration mode	
[no] offset-list <access-list-number |access-list-name> {in|out} <number > [<ifname>]	Configure that provide a deviation value to the route metric value when the port sends or receives RIPng data packet; the **[no] offset-list<*access-list-number* |*access-list-name*> {in|out} <*number* > [<*ifname*>]**command removes the deviation table

3）configure and apply route filter and route aggregation

Command	**Explanation**
Router configuration mode	
[no] distribute-list {< ***access-list-number*** **|*access-list-name* > | prefix<*prefix-list-name*>} {in|out} [<*ifname*>]**	Set to filter the route when the interface sends and receives RIPng data packets. The **[no] distribute-list {<** ***access-list-number*** **|*access-list-name* > | prefix<*prefix-list-name*>} {in|out} [<*ifname*>]** command means do not set the route filter
[no]aggregate-address <*IPv6-address*>	Configure route aggregation, the **[no]aggregate-address <*IPv6-address*>** command cancels the route aggregation.

4）configure split horizon

Command	**Explanation**
Interface configuration mode	
IPv6 rip split-horizon [poisoned]	Configure that take the split-horizon when the port sends data packets, poisoned means with poison reverse
no IPv6 rip split-horizon	Cancel the split-horizon.

(3) Configure other RIPng protocol parameters

1）Configure timer for RIPng update, timeout and hold-down

Command	**Explanation**
Router configuration mode	
timers basic <*update*> <*invalid*> <*garbage*> **no timers basic**	Adjust the renew, timeout and garbage recycle RIPng timer, the no timers basic command restore the default configuration

(4) Delete the specified route in RIPng route table

Command	**Explanation**
Admin Mode	
clear IPv6 rip route {<*IPv6-address*>|kernel|static|connected|rip|ospf|isis|bgp|all}	the command deletes a specified route from the RIPng route table

3. RIPng aggregation configuration task list

(1) To configure ipv6 aggregation route globally

Command	**Explanation**
Router Configuration Mode	
ipv6 rip aggregate-address X:X::X:X/M **no ipv6 rip aggregate-address X:X::X:X/M**	To configure or delete ipv6 aggregation route globally.

(2) To configure ipv6 aggregation route on interface.

Command	**Explanation**
Interface Configuration Mode	
ipv6 rip aggregate-address X:X::X:X/M **no ipv6 rip aggregate-address X:X::X:X/M**	To configure or delete ipv6 aggregation route on interface.

(3) To display ipv6 aggregation route information

Command	**Explanation**
Admin Mode and Configuration Mode	
show ipv6 rip aggregate	To display ipv6 aggregation route information, such as aggregation interface, metric, numbers of aggregation route times of aggregation.

4. Redistribution of OSPFv3 Routing to RIPng

（1）To enable redistribution of OSPFv3 routing for RIPng

Command	Notes
Router ipv6 rip configuration mode	

redistribute ospf [<*process-tag*>] [metric<*value*>] [route-map<*word*>] **no redistribute ospf [<*process-tag*>]**	To enable or disable redistribution of OSPFv3 routing for RIPng.

（2）To display configuration information

Command	Notes
Admin mode and configuration mode	
show ipv6 rip redistribute	To display RIPng routing which is redistributed from other routing protocols.

（3）Debugging

Command	Notes
Admin mode	
debug ipv6 rip redistribute message send **no debug ipv6 rip redistribute message send** **debug ipv6 rip redistribute route receive** **no debug ipv6 rip redistribute route receive**	To enable or disable debugging messages sent to NSM by RIP(ng) for redistribution of OSPFv3 routing. To enable or disable debugging route messages received from NSM.

1.5.3 Commands For RIPng

1.5.3.1 clear ipv6 route

Command: clear ipv6 rip route { <*ipv6-address* >| kernel |static | connected |rip |ospf |isis | bgp |all }

Function: Clear specific route from the RIPng route table

Parameter: Clears the route exactly match with the destination address from the RIP route table ***<ipv6-address* >** is the destination address shown in hex notation with prefix length.

kernel delete kernel route from the RIPng route table

static delete static route from the RIPng route table

connected delete direct route from the RIPng route table

rip delete RIPng route from the RIPng route table only

ospf delete IPv6 OSPF route from the RIPng route table only

bgp delete IPv6 BGP route from the RIPng route table only

ISIS delete ivp6 isis route from the RIPng route table only

all delete all routes from the RIPng route table

Default: No default configuration

Command Mode: Admin mode

Usage Guide: All routes in the RIPng route table will be deleted by using this command with all parameters.

Example: Switch#clear ipv6 rip route 2001:1:1::/64

Switch#clear ipv6 rip route ospf

1.5.3.2 default-information originate

Command: default-information originate

no default-information originate

Function: Permit redistributing the network 0:: into RIPng. The "**no default-information originate**" disables this function

Parameter: None

Default: Disabled

Command Mode: Router mode

Example: Switch#config terminal

Switch(config)#router ipv6 rip

Switch(config-router)#default-information originate

1.5.3.3 default-metric

Command: default-metric <*value*>

no default-metric

Function: Set the default metric route value of the introduced route; the "**no default-metric**" restores the default value.

Parameter: <*value*> is the route metric value to be set, ranging between 1～16.

Default: Default route metric value is 1.

Command Mode: Router mode

Usage Guide: Default-metric command is used for setting the default route metric value of the routes from other routing protocols when distributed into the RIPng routes. When using the **redistribute** command for introducing routes from other protocols, the default route metric value specified by **default-metric** will be adopted if no specific route metric value is set.

Example: Set the default route metric value of the routes from other routing protocols when distributed into the RIPng routes as 3.

Switch(config-router)#default-metric 3

1.5.3.4 distance

Command: distance <*number*> [<*ipv6-address*>]
[<*access-list-name*|*access-list-number*>]
no distance [<*ipv6-address*>]

Function: Set the managing distance with this command. The "**no distance [<*A.B.C.D/M*>]**" command restores the default value to 120.

Parameter: <*number*> specifies the distance value, ranging between 1-255. **<*ipv6-address*>** is the local link address or its prefix. **<*access-list-name*|*access-list-number*>** specifies the access-list number or name applied.

Default: The default managing distance of RIP is 120.

Command Mode: Router mode and address-family mode.

Usage Guide: In case there are routes from two different routing protocols to the same destination, the managing distance is then used for selecting routes. The less the managing distance of the route protocol is , the more reliable will be the route acquired from the protocol.

Example:

```
Switch#config terminal
Switch(config)#router rip
Switch(config-router)#distance 8 fe80:1111::4200:21ff:fe00:11 mylist
```

1.5.3.5 ipv6 rip aggregate-address

Command: ipv6 rip aggregate-address X:X::X:X/M
no ipv6 rip aggregate-address X:X::X:X/M

Function: To configure IPv6 aggregation route. The no form of this command delete the IPv6 aggregation route.

Parameter: X:X::X:X/M：ipv6 address and prefix length.

Command Mode: Router Mode or Interface Configuration Mode.

Default: No aggregation route configured.

Usage Guide: If to configure aggregation route under router mode, RIPng protocol must be enabled. If configured under interface configuration mode, RIPng protocol may not be enabled, but the aggregation route can operation after the RIPng protocol be enabled on interface.

Example: To configure aggregation route as 2001:3f:ed8::99/64 globally.

```
Switch(config)#router rip
Switch(config-router)#ipv6 rip agg 2001:3f:ed8::99/64
```

1.5.3.6 ipv6 rip split-horizon

Command:ipv6 rip split-horizon [poisoned]
no ipv6 rip split-horizon

Function: Permit the split horizon. The "**no ipv6 rip split-horizon**" disables the split horizon

Parameter: [poisoned] configures split horizon with poison reverse.

Default: Split horizon with poison reverse

Command Mode: Interface Mode

Usage Guide: The split horizon is for preventing the routing loops, namely preventing the layer 3 switch from broadcasting a route at the interface from which the very route is learnt. The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example: Switch#config terminal

Switch(config)#interface Vlan1

Switch(Config-if-Vlan1)#ipv6 rip split-horizon poisoned

1.5.3.7 distribute-list

Command: distribute-list{*access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>|vlan <*vlan-id*>]

no distribute-list{*access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>|vlan <*vlan-id*>]

Function: This command uses access-list or prefix-list to filter the route renews messages sent and received. The "**no distribute-list{*access-list-name*> |prefix<*prefix-list-name*>} {in|out} [<*ifname*>|vlan <*vlan-id*>]**" command cancels this filter function

Parameter: <*access-list-name*> is the name or access-list number to be applied. **<*prefix-list-name*>** is the name of the prefix-list to be applied. **<*ifname*>** specifies the name of interface to be applied with route filtering

Default: Function disabled by RIPng by default

Command Mode: Router mode

Usage Guide: The filter will be applied to all interfaces if no specific interface is set.

Example:

Switch#config terminal

Switch(config)#router ipv6 rip

Switch(config-router)#distribute-list prefix myfilter in Vlan1

1.5.3.8 ipv6 router rip

Command: ipv6 router rip

no ipv6 router rip

Function: Enable RIPng on the interface. The "no ipv6 router rip" command disables RIPng on the interface.

Default: Not configured

Command Mode: Interface Mode

Usage Guide: The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example:

```
Switch#config terminal
Switch(config)#interface Vlan1
Switch(Config-if-Vlan1)#ipv6 router rip
```

1.5.3.9 neighbor

Command: neighbor <*ipv6-address*> <*ifname*> [vlan <*vlan-id*>]

no neighbor <*ipv6-address*> <*ifname*> [vlan <*vlan-id*>]

Function: Specify the destination address for fixed sending. The "**no neighbor <*ipv6-address*> <*ifname*> [vlan <*vlan-id*>]** " cancels the specified address defined and restores all trusted gateways

Parameter: <*ipv6-address*> is the IPv6 Link-local address specified for sending and shown in colon hex notation without the prefix length. **<*ifname*>** is the name of interface.

Default: Not sending to any fixed destination address

Command Mode: Router mode

Usage Guide: When used associating passive-interface command it would be able to send routing messages to specified neighbor only.

Example:

```
Switch#config terminal
Switch(config)#router ipv6 rip
Switch(config-router)#neighbor FE80:506::2 Vlan1
```

1.5.3.10 Offset-list

Command: offset-list <*access-list-number*|*access-list-name*> {in|out }<*number* >[<*ifname*> no neighbor <*ipv6-address*> <*ifname*> |vlan <*vlan-id*>]

no offset-list <*access-list-number*|*access-list-name*> {in|out }<*number* >[<*ifname*> no neighbor <*ipv6-address*> <*ifname*> |vlan <*vlan-id*>]

Function: Add an offset value on the routing metric value learnt by RIPng. The "**no offset-list <*access-list-number*|*access-list-name*> {in|out }<*number* >[<*ifname*> no neighbor <*ipv6-address*> <*ifname*> |vlan <*vlan-id*>]**" command disables this function

Parameter: < *access-list-number* |*access-list-name*> is the access-list or name to be applied. **<*number* >** is the additional offset value, ranging between 0-16; <*ifname*> is the name of specific interface

Default: The default offset value is the metric value of the interface defined by the system.

Command Mode: Router mode

Example:

```
Switch#config terminal
```

Switch(config)#router ipv6 rip

Switch(config-router)#offset-list 1 in 5 Vlan1

1.5.3.11 passive-interface

Command: passive-interface{<*ifname*>|vlan <*vlan-id*>}

no passive-interface{<*ifname*>|vlan <*vlan-id*>}

Function: Set the RIPng layer 3 switches to block RIPng broadcast on the specified interfaces, and only send the RIPng data packet to the layer 3 switch which is configured with neighbor.

Parameter: <***ifname***> is the specific interface name

Default: Not configured

Command Mode: Router mode

Example:

Switch#config terminal

Switch(config)#router ipv6 rip

Switch(config-router)#passive-interface Vlan1

1.5.3.12 redistribute

Command: redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]

no redistribute {kernel |connected| static| ospf| isis| bgp} [metric<*value*>] [route-map<*word*>]

Function: Introduce the routes learnt from other routing protocols into RIPng

Parameter: kernel introduce from kernel routes

connected introduce from direct routes

static introduce from static routes

ospf introduce from IPv6 OSPF routes

isis introduce from IPv6 ISIS routes

bgp introduce from IPv6 BGP routes

<*value*> is the metric value assigned to the introduced route, ranging between 0-16

<*word*> is the probe poining to the route map for introducing routes

Command Mode: Router mode

Example:

Switch#config terminal

Switch(config)#router ipv6 rip

Switch(config-router)#redistribute kernel route-map ipi

1.5.3.13 redistribute ospf

Command: redistribute ospf [<*process-tag*>] [metric<*value*>] [route-map<*word*>]

no redistribute ospf [<*process-tag*>]

Function: To redistribute routing information from external OSPFv3 processes to RIPng process. The no form of this command will remove the introduced OSPFv3 routing entries.

Parameters: process-tag is the string tag for OSPFv3 process with maximum length limited within 15 characters. If not specified, the default process will be used.

metric <value> is the weight metric for the introduced routing entries, limited between 0 and 16.

route-map <word> is the pointer to the introduced routing map.

Default: Not redistributed by default.

Command Mode: RIPng configuration mode.

Usage Guide: None.

Example: To redistribute ospfv3 abc routing ro ripng.

```
Switch(config)#router ipv6 rip
Switch(config-router)#redistribute ospf abc
```

1.5.3.14 route

Command: route <*ipv6-address*>

no route <*ipv6-address*>

Function: This command configures a static RIPng route. The“**no route <*ipv6-address*>**“ command deletes this route.

Parameter: Specifies this destination IPv6 address prefix and its length show in colon hex notation.

Usage Guide: The command add a static RIPng route, and is mainly used for debugging. Routes configured by this command will not appear in kernel route table but in the RIPng route database, however it could be located by using the show ipv6 rip command.

Command Mode: Router mode

Example:

```
Switch#config terminal
Switch(config)#router ipv6 rip
Switch(config-router)#route 3ffe:1234:5678::1/64
```

1.5.3.15 router ipv6 rip

Command: router ipv6 rip

no router ipv6 rip

Function: Enable RIPng routing process and entering RIPng mode; the “**no router ipv6 rip**” of this command disables the RIPng routing protocol.

Default: RIPng routing not running

Command Mode: Global mode

Usage Guide: This command is for enabling the RIPng routing protocol, this command should be enabled before performing other global configuration of the RIPng protocol.

Example: Enable the RIPng protocol mode

Switch(config)#router ipv6 rip

1.5.4 RIPng Configuration Examples

1.5.4.1 Typical RIP Examples

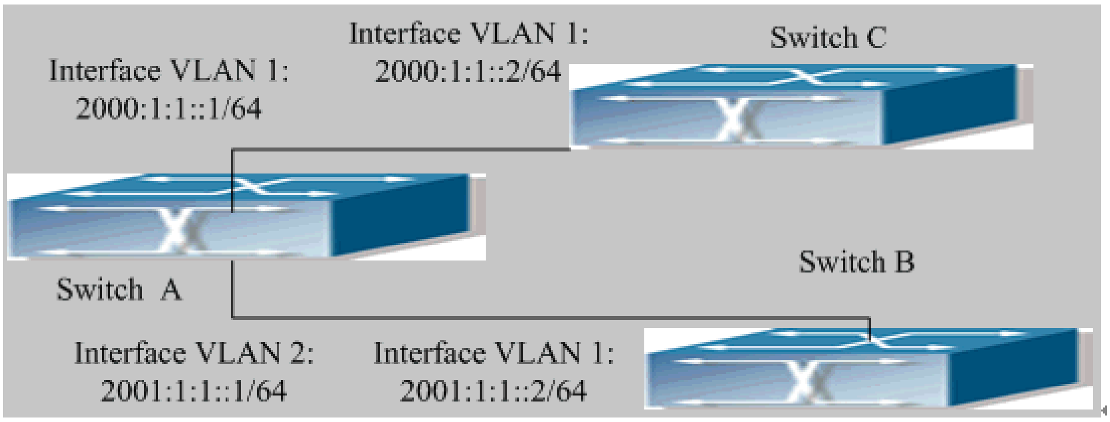

Fig　1-6 RIPng Example

As shown in the above figure, a network consists of three layer 3 switches. SwitchA and SwitchB connect to SwitchC through interface vlan1 and vlan2. All the three switches are running RIPng. Assume SwitchA（VLAN1:2001:1:1::1/64 and VLAN2:2001:1:1::1/64）exchange update information with SwitchB（VLAN1:2001:1:1::2/64） only, update information is not exchanged between SwitchA and SwitchC（VLAN1:20001:1::2/64）.

The configuration for SwitchA, SwitchB and SwitchC is shown below:

Layer 3 switch SwitchA

Enable RIPng protocol

SwitchA (config)#router IPv6 rip

SwitchA (config-router)#exit

Configure the IPv6 address in vlan1 and configure vlan1 to run RIPng

SwitchA#config

SwitchA (config)# interface Vlan1

SwitchA (config-if-Vlan1)# IPv6 address 2000:1:1::1/64

SwitchA (config-if-Vlan1)#IPv6 router rip

SwitchA (config-if-Vlan1)#exit

Configure the IPv6 address in vlan2 and configure vlan2 to run RIPng

```
SwitchA (config)# interface Vlan2
SwitchA (config-if-Vlan2)# IPv6 address 2001:1:1::1/64
SwitchA (config-if-Vlan2)#IPv6 router rip
SwitchA (config-if-Vlan2)#exit
```

Configure the interface vlan1 do not send RIPng messages to SwitchC

```
SwitchA (config)#
SwitchA (config-router)#passive-interface Vlan1
SwitchA (config-router)#exit
```

Layer 3 switch SwitchB

Enable RIPng protocol

```
SwitchB (config)#router IPv6 rip
SwitchB (config-router-rip)#exit
```

Configure the IPv6 address and interfaces of Ethernet port vlan1 to run RIPng

```
SwitchB #config
SwitchB (config)# interface Vlan1
SwitchB (config-if)# IPv6 address 2001:1:1::2/64
SwitchB (config-if)#IPv6 router rip
SwitchB (config-if)exit
```

Enable RIPng protocol

```
SwitchC (config)#router IPv6 rip
SwitchC (config-router-rip)#exit
```

Configure the IPv6 address and interfaces of Ethernet port vlan1 to run RIPng

```
SwitchC#config
SwitchC (config)# interface Vlan1
SwitchC (config-if)# IPv6 address 2000:1:1::2/64
SwitchC (config-if)#IPv6 router rip
SwitchC (config-if)exit
```

1.5.4.2 Typical examples of RIPng aggregation function

The application topology as follows：

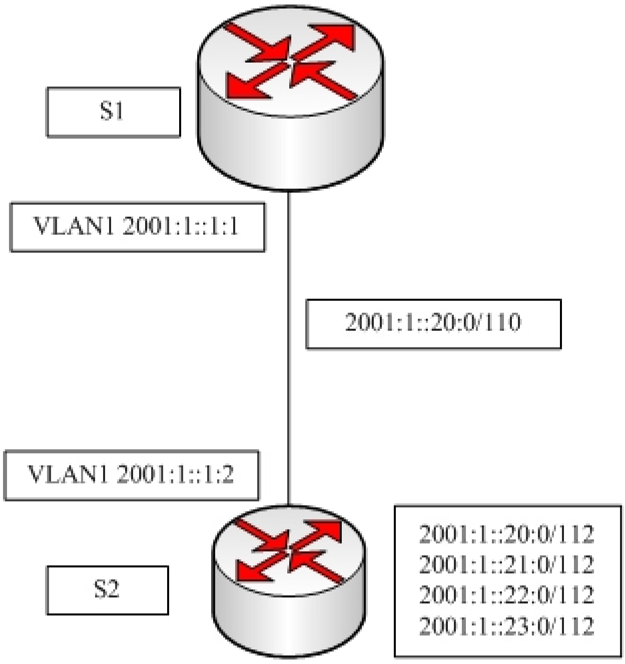

Fig 1-7 Typical application of RIPng aggregation

As the above network topology, S2 is connected to S1 through interface vlan1, there are other 4 subnet routers of S2, which are 2001:1::20:0/112, 2001:1::21:0/112, 2001:1::22:0/112, 2001:1::23:0/112. S2 supports route aggregation, and to configure aggregation route 2001:1::20:0/110 in interface vlan1 of S2, after that, sending router messages to S2 through vlan1, and put the four subnet routers aggregated to one router as 2001:1::20:0/110, and send to S1, and not send subnet to neighbor. It can reduce the router table of S1, save the memory.

S1 configuration list：

```
Switch1(config)#router ipv6 rip
Switch1(config-router)#network vlan 1
```

S2 configuration list：

```
Switch2(config)#router ipv6 rip
Switch2(config-router)#network vlan 1
Switch2(config-router)#exit
Switch2(config)#in vlan 1
Switch2(Config-if-Vlan1)#ipv6 rip agg 2001:1::20:0/110
```

1.5.5 RIPng Troubleshooting Help

The RIPng protocol may not be working properly due to errors such as physic connection, configuration error when configuring and using the RIPng protocol. So users should pay attention to the following:

first ensure the physic connection is correct and the IP Forwarding command is open

second, ensure the interface and link layer protocol are UP (use show interface command)

then initiate the RIPng protocol (use router IPv6 rip command) and configure the port (use IPv6 router command),and set RIPng protocol parameter on corresponding interfaces.

After that, a RIPng protocol feature should be noticed ---the Layer 3 switch running RIPng transmits the route updating messages every 30 seconds. A Layer 3 switch is considered inaccessible if no route updating messages from the switch are received within 180 seconds, then the route to the switch will remains in the route table for 120 seconds before it is deleted. Therefore, if to delete a RIPng route, this route item is assured to be deleted from route table after 300 seconds.

If the RIP routing problem remains unresolved, please use debug IPv6 rip command to record the debug message in three minutes, and send them to our technical service center.

1.5.5.1 Monitor And Debug Command

1.5.5.1.1 debug ipv6 rip

Command: debug ipv6 rip [events| nsm| packet[recv|send][detail]| all]
no debug ipv6 rip [events| nsm| packet[recv|send][detail]| all]

Function: For opening various debugging switches of RIPng, showing various debugging messages. The "**no debug ipv6 rip [events| nsm| packet[recv|send][detail]| all]**" command close the corresponding debugging switch.

Parameter: **events** shows the debugging message of RIPng events
nsm shows the communication messages between RIPng and NSM
packet shows the debugging messages of RIPng data packets
recv shows the messages of the received data packets
send shows the messages of the sent data packets
detail shows the messages of the data packets received or sent.

Default: Not enabled

Command Mode: Admin mode

Example: Switch#debug ipv6 rip packet

```
Switch#1970/01/01 21:15:08 IMI: SEND[Ethernet1/10]: Send to [ff02::9]:521
1970/01/01 21:15:08 IMI: SEND[Ethernet1/2]: Send to [ff02::9]:521
1970/01/01 21:15:09 IMI: RECV[Ethernet1/10]: Receive from [fe80::20b:46ff:fe57:8e60]:521
1970/01/01 21:15:09 IMI: RECV[Ethernet1/10]: 3000:1:1::/64 is filtered by access-list dclist
1970/01/01 21:15:09 IMI: RECV[Ethernet1/10]: 3ffe:1:1::/64 is filtered by access-list dclist
1970/01/01 21:15:15 IMI: RECV[Ethernet1/2]: Receive from [fe80::203:fff:fe01:257c]:521
```

1.5.5.1.2 debug ipv6 rip redistribute message send

Command: debug ipv6 rip redistribute message send

no debug ipv6 rip redistribute message send

Function: To enable the debugging of sending messages for routing redistribution messages from OSPFv3 or other external process for RIPng. The no form of this command will disable the debugging messages.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 rip redistribute message send
Switch#no debug ipv6 rip redistribute message send
```

1.5.5.1.3 debug ipv6 rip redistribute route receive

Command: debug ipv6 rip redistribute route receive

no debug ipv6 rip redistribute route receive

Function: To enable the debugging messages received from NSM for redistribution of routing information for RIPng. The no form of this command will disable the debugging information.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 rip redistribute route receive
Switch#no debug ipv6 rip redistribute route receive
```

1.5.5.1.4 show debugging ipv6 rip

Command: show debugging ipv6 rip

Function: Show RIPng debugging status for following debugging options: nsm debugging, RIPng event debugging, RIPng packet debugging and RIPng nsm debugging

Command Mode: All mode

Example: Switch#show debugging rip

```
RIPng debugging status:
RIPng event debugging is on
RIPng packet detail debugging is on
RIPng NSM debugging is on
```

1.5.5.1.5 show ipv6 rip aggregate

Command: show ipv6 rip aggregate

Function: To display the information of ipv6 aggregation route.

Parameter: None.

Command Mode: Admin Mode and Configuration Mode.

Default: None.

Usage Guide: This command is used to display which interface the aggregation route be configured, Metric, Count, Suppress and so on. If configured under global mode, then the interface display "----", "Metric" is metric. "Count" is the number of learned aggregation routes. "Suppress" is the times of aggregation.

Example: To display the information of ipv6 aggregation route.

```
Switch(config-router)#show ipv6 rip aggregate

 Aggregate information of ripng

        Network                   Aggregated Ifname      Metric Count Suppress
        2001::/16                 Vlan1                    1      2      0
        2001:1::/32                ----                    1      2      0
        2001:1:2::/60              Vlan1                   1      1      1
                                  ----                   1      1      1
```

Displayed information	Notes
Network	Route prefix and prefix length
Aggregated Ifname	To configure the interface name of the aggregation route. If the route aggregated globally , then display "---".
Metric	Metric of aggregation route
Count	The number of learned aggregation routes
Suppress	The times of aggregated for aggregation route.

1.5.5.1.6 show ipv6 rip interface

Command: show ipv6 rip interface

Function: Make sure the interface and line protocols is up,.

Command Mode: All mode

Example: Switch(config)#show ipv6 rip interface

```
Loopback is up, line protocol is up
  RIPng is not enabled on this interface
  Vlan1 is up, line protocol is up
  Routing Protocol: RIPng
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IPv6 interface address:
      3000:1:1::1/64
      fe80::203:fff:fe0c:cda/64
```

Displayed information	Explanations
Vlan1 is up, line protocol is up	Interface is Up
Routing Protocol: RIP	The routing protocol running on the interface is RIPng
Passive interface: Disabled	Passive-interface disabled
Split horizon:Enabled with Poisoned Reversed	The split horizon is enabled with poisoned reversed on the interface.
IP interface address: 3000:1:1::1/64 fe80::203:fff:fe01:429e/64	IPv6 address of the interface

1.5.5.1.7 show ipv6 rip redistribute

Command: show ipv6 rip redistribute

Function: Show the configuration information of redistributed other out routing to RIPng.

Parameter: None.

Default: Not shown by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ipv6 rip redistribute
```

1.5.5.1.8 show ipv6 protocols rip

Command: show ipv6 protocols rip

Function: Show the RIPng process parameters and statistic messages

Command Mode: All mode

Example: Switch(config)#show ipv6 protocols rip

```
Routing Protocol is "RIPng"
  Sending updates every 30 seconds with +/-50%, next due in 1 second
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Ethernet1/10 filtered by dclist
  Default redistribute metric is 1
  Redistributing: static
  Interface
    Vlan10
    Vlan2
  Routing for Networks:
```

Displayed information	Explanations
Sending updates every 30 seconds with +/-50%, next due in 1 seconds	Sending updates every 30 seconds
Timeout after 180 seconds, garbage collect after 120 seconds	The route timeout time is 180 seconds, the garbage collect time is 120 seconds
Outgoing update filter list for all interface is not set	Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set	Incoming update filter list for all interface is not set
Default redistribution metric is 1	Default redistribution metric is 1
Redistributing: static	Redistricting the static route into the RIPng routes
Interface Vlan10 Vlan2	The interfaces running RIPng is Vlan 10 and Vlan 2

1.5.5.1.9 show ipv6 rip

Command: show ipv6 rip

Function: Show RIPng Routing

Command Mode: All mode

Example: Switch#show ipv6 rip

```
Codes: R - RIP, K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS,
          B - BGP, a - aggregate, s - suppressed
   Network                          Next Hop                      If       Met Tag   Time
R   2000:1:1::/64                 ::                            Vlan2     1     0
R   2001:1:1::/64                 fe80::203:fff:fe01:257c       Vlan2     2     0 02:40
R   3000:1:1::/64                 ::                            Vlan10     1     0
R   3010:1:1::/64                 ::                              --        1     0
```

Amongst R stands for RIPng route, namely a RIPng route with the destination network address 2001:1:1::/64, next-hop address at fe80::203:fff:fe01:257c. It is learnt from the Ethernet port VLAN2 with a metric value of 2, and still has 2 minutes 40 seconds before time out.

1.5.5.1.10 show ipv6 rip database

Command: show ipv6 rip database

Function: Show messages related to RIPng database

Command Mode: All mode

Example: Switch#show ipv6 rip database

1.6 OSPF

1.6.1 Introduction to OSPF

OSPF is abbreviation for Open Shortest Path First. It is an interior dynamic routing protocol for autonomous system based on link-state. The protocol creates a link-state database by exchanging link-states among layer3 switches, and then uses the Shortest Path First algorithm to generate a route table basing on that database.

Autonomous system (AS) is a self-managed interconnected network. In large networks, such as the Internet, a giant interconnected network is broken down to autonomous systems. Big enterprise networks connecting to the Internet are independent AS, since the other host on the Internet are not managed by those AS and they don’t share interior routing information with the layer3 switches on the Internet.

Each link-state Layer3 switch can provide information about the topology with its neighboring Layer3 switches.

- The network segment (link) connecting to the layer3 switch
- State of the connecting link

Link-state information is flooded throughout the network so that all Layer3 switches can get firsthand information. Link-state Layer3 switches will not broadcast all information contained in their route tables; instead, they only send changed link-state information. Link-state Layer3 switches establish neighborhood by sending “HELLO” to their neighbors, then link-state advertisements (LSA) will be sent among neighboring Layer3 switches. Neighboring Layer3 switch copy the LSA to their routing table and transfer the information to the rest part of the network. This process is referred to as “flooding”. In this way, firsthand information is sent throughout the network to provide accurate map for creating and updating routes in the network.

Link-state routing protocols use cost instead of hops to decide the route. Cost is assigned automatically or manually. According to the algorithm in link-state protocol, cost can be used to calculate the hop number for packets to pass, link bandwidth, and current load of the link. The administrator can even add weight for better assessment of the link-state.

1) When a link-state layer3 switch enters a link-state interconnected network, it sends a HELLO packet to get to know its neighbors and establish neighborhood.

2) The neighbors respond with information about the links they are connecting and the related costs.

3) The originate layer3 switch uses this information to build its own routing table

4) Then, as part of the regular update, layer3 switch send link-state advertisement (LSA) packets to its neighboring layer3 switches. The LSA include links and related costs of that layer3 switch.

5) Each neighboring layer3 switch copies the LSA packet and passes it to the next neighbor (i.e. flooding).

6) Since routing database is not recalculated before layer3 switch forwards LSA flooding, the converging time is greatly reduced.

One major advantage of link-state routing protocols is the fact that infinite counting is impossible, this is because of the way link-state routing protocols build up their routing table. The second advantage is that converging in a link-state interconnected network is very fast, once the routing topology changes, updates will be flooded throughout the network very soon. Those advantages release some layer3 switch resources, as the process ability and bandwidth used by bad route information are minor.

The features of OSPF protocol include the following: OSPF supports networks of various scales, several hundreds of layer3 switches can be supported in an OSPF network. Routing topology changes can be quickly found and updating LSAs can be sent immediately, so that routes converge quickly. Link-state information is used in shortest path algorithm for route calculation, eliminating loop route. OSPF divides the autonomous system into areas, reducing database size, bandwidth occupation and calculation load. (According to the position of layer3 switches in the autonomous system, they can be grouped as internal area switches, area border switches, AS border switches and backbone switches). OSPF supports load balance and multiple routes to the same destination of equal costs. OSPF supports 4 level routing mechanisms (process routing according to the order of intra-area path, inter-area path, type 1 external path and type 2 external path). OSPF supports IP subnet and redistribution of routes from the other routing protocols, and interface-based packet verification. OSPF supports sending packets in multicast.

Each OSPF layer3 switch maintains a database describing the topology of the whole autonomous system. Each layer3 switch gathers the local status information, such as available interface, reachable neighbors, and sends link-state advertisement (sending out link-state information) to exchange link-state information with other OSPF layer3 switches to form a link-state database describing the whole autonomous system. Each layer3 switch builds a shortest path tree rooted by itself according to the link-state database, this tree provides the routes to all nodes in an autonomous system. If two or more layer3 switches exist (i.e. multi-access network), "designated layer3 switch” and “backup designated layer3 switch” will be selected. Designated layer3 switch is responsible for spreading link-state of the network. This concept helps reducing the traffic among the Layer3 switches in multi-access network.

OSPF protocol requires the autonomous system to be divided into areas. That is to divide the autonomous system into 0 area (backbone area) and non-0 areas. Routing information between areas are further abstracted and summarized to reduce the bandwidth required in the network. OSPF uses four different kinds of routes; they are intra-area route, inter-area route, type 1 external route and type 2 external route, in the order of highest priority to lowest. The route inside an area and between areas describe the internal network structure of an autonomous system, while external routes describe how to select the routing information to destination outside the autonomous system. The first type of external route corresponds to the information introduced by OSPF from the other interior routing protocols, the costs of those routes are comparable with the costs of OSPF routes; the second type of external route corresponds to the information introduced by OSPF from the other external routing protocols, but the costs of those routes are far greater than that of OSPF routes, so OSPF route cost is ignored when calculating route costs.

OSPF areas are centered with the Backbone area, identified as Area 0, all the other areas must be connected to Area 0 logically, and Area 0 must be continuous. For this reason, the concept of virtual link is introduced to the backbone area, so that physically separated areas still have logical connectivity to the backbone area. The configurations of all the layer3 switches in the same area must be the same.

In conclusion, LSA can only be transferred between neighboring Layer3 switches, OSPF protocol includes 5 types of LSA: router LSA, network LSA, network summary LSA to the other areas, ASBR summary LSA and AS external LSA. They can also be called type1 LSA, type2 LSA, type3 LSA, type4 LSA, and type5 LSA. Router LSA is generated by each layer3 switch inside an OSPF area, and is sent to all the other neighboring layer3 switches in the same area; network LSA is generated by the designated layer3 switch in the OSPF area of multi-access network, and is sent to all other neighboring layer3 switches in this area. (In order to reduce traffic on layer3 switches in the multi-access network, “designated layer3 switch” and “backup designated layer3 switch” should be selected in the multi-access network, and the network link-state is broadcasted by the designated layer3 switch); network summary LSA is generated by border switches in an OSPF area , and is transferred among area border layer3 switches; AS external LSA is generated by layer3 switches on external border of AS, and is transferred throughout the AS.

As to autonomous systems mainly advertises external link-state, OSPF allow some areas to be configured as STUB areas to reduce the size of the topology database. Type4 LSA (ASBR summary LSA) and type5 LSA (AS external LSA) are not allowed to flood into/through STUB areas. STUB areas must use the default routes, the layer3 switches on STUB area edge advertise the default routes to STUB areas by type 3 summary LSA, those default routes only flood inside STUB area and will not get out of STUB area. Each STUB area has a corresponding default route, the route from a STUB area to AS external destination must rely on the default route of that area.

The following simply outlines the route calculation process of OSPF protocol:

1） Each OSPF-enabled layer3 switch maintains a database (LS database) describing the link-state of the topology structure of the whole autonomous system. Each layer3 switch generates a link-state advertisement according to its surrounding network topology structure (router LSA), and sends the LSA to other layer3 switches through link-state update (LSU) packets. Thus each layer3 switches receives LSAs from other layer3 switches, and all LSAs are combined to the link-state database.

2） Since a LSA is the description of the network topology structure around a layer3 switch, the LS database is the description of the network topology structure of the whole network. The layer3 switches can easily create a weighted vector map according to the LS database. Obviously, all layer3 switches in the same autonomous system will have the same network topology map.

3） Each layer3 switch uses the shortest path first (SPF) algorithm to calculate a tree of shortest path rooted by itself. The tree provides the route to all the nodes in the autonomous system, leaf nodes consist of the external route information. The external route can be marked by the layer3 switch broadcast it, so that additional information about the autonomous system can be recorded. As a result, the route table of each layer3 switch is different.

OSPF protocol is developed by the IETF, the OSPF v2 widely used now is fulfilled according to the content described in RFC2328.

1.6.2 OSPF Configuration Task List

The OSPF configuration for Edge-core series switches may be different from the configuration procedure to switches of the other manufacturers. It is a two-step process:

1、Enable OSPF in the Global Mode;

2、Configure OSPF area for the interfaces.

The configuration task list is as follows:

1. Enable/disable OSPF protocol (required)

（1）Enable/disable OSPF protocol (required)

（2）Configure the Router-id of the layer3 switch running OSPF (optional)

（3）Configure the network scope for running OSPF (optional)

（4）Configure the area for the interface (required)

2. Configure OSPF protocol parameters (optional)

（1）Configure OSPF packet sending mechanism parameters

1）Configure OSPF packet verification

2）Set the OSPF interface to receive only

3）Configure the cost for sending packets from the interface

4）Configure OSPF packet sending timer parameter (timer of broadcast interface sending HELLO packet to poll, timer of neighboring layer3 switch invalid timeout, timer of LSA transmission delay and timer of LSA retransmission.

（2）Configure OSPF route introduction parameters

1）Configure default parameters (default type, default tag value, default cost

2）Configure the routes of the other protocols to introduce to OSPF

（3）Configure OSPF importing the routes of other OSPF processes

1) Enable the function of OSPF importing the routes of other OSPF processes

2) Display relative information

3) Debug

（4）Configure other OSPF protocol parameters

1）Configure OSPF routing protocol priority

2）Configure cost for OSPF STUB area and default route

3）Configure OSPF virtual link

4）Configure the priority of the interface when electing designated layer3 switch (DR).

5）Configure to keep a log for OSPF adjacency changes or not

3. Disable OSPF protocol

1. Enable OSPF protocol

Basic configuration of OSPF routing protocol on ES4700 series switches is quite simple, usually only enabling OSPF and configuration of the OSPF area for the interface are required. The OSPF protocol parameters can use the default settings. If OSPF protocol parameters need to be modified, please refer to “2. Configure OSPF protocol parameters”.

Command	**Explanation**
Global mode	
[no] router ospf [process <*id*>]	Enables OSPF protocol; the “**no router ospf**” command disables OSPF protocol (required)
OSPF protocol configuration mode	

router-id <*router_id*> **no router-id**	Configures the ID number for the layer3 switch running OSPF; the "**no router id**" command cancels the ID number. The IP address of an interface is selected to be the layer3 switch ID. (optional)
[no] network {**<*network*> <*mask*> | <*network*>/<*prefix*>} area <*area_id*>**	Configure certain segment to certain area, the no **[no] network** {**<*network*> <*mask*> | <*network*>/<*prefix*>} area <*area_id*>** command cancels this configuration. (required)
[no] passive-interface {*IFNAME* | ethernet *IFNAME* | Vlan <*ID*>}	Sets an interface to receive only, the [no] passive-interface {*IFNAME* | ethernet *IFNAME* | Vlan <*ID*>} command cancels this configuration.

2. Configure OSPF protocol parameters

（1）Configure OSPF packet sending mechanism parameters

1）Configure OSPF packet verification

2）Set the OSPF interface to receive only

3）Configure the cost for sending packets from the interface

Command	**Explanation**
Interface configuration mode	
ip ospf authentication **{ message-digest | null}** **no ip ospf authentication**	Configures the authentication method by the interface to accept OSPF packets; the no ip ospf authentication command restores the default settings.
ip ospf authentication-key LINE **no ip ospf authentication-key**	Configure the key of the authentication process of OSPF data packets receiving for the interfaces; the no action of this command restores the default settings.
ip ospf cost <*cost* > **no ip ospf cost**	Sets the cost for running OSPF on the interface; the "**no ip ospf cost**" command restores the default setting.

4）Configure OSPF packet sending timer parameter (timer of broadcast interface sending HELLO packet to poll, timer of neighboring layer3 switch invalid timeout, timer of LSA transmission delay and timer of LSA retransmission.

Command	**Explanation**
Interface configuration mode	

ip ospf hello-interval <*time*> **no ip ospf hello-interval**	Sets interval for sending HELLO packets; the "**no ip ospf hello-interval**" command restores the default setting.
ip ospf dead-interval <*time* > **no ip ospf dead-interval**	Sets the interval before regarding a neighbor layer3 switch invalid; the "**no ip ospf dead-interval**" command restores the default setting.
ip ospf transit-delay <*time*> **no ip ospf transit-delay**	Sets the delay time before sending link-state broadcast; the "**no ip ospf transmit-delay**" command restores the default setting.
ip ospf retransmit <*time*> **no ip ospf retransmit**	Sets the interval for retransmission of link-state advertisement among neighbor layer3 switches; the "**no ip ospf retransmit**" command restores the default setting.

（2）Configure OSPF route introduction parameters

Configure the routes of the other protocols to introduce to OSPF.

Command	**Explanation**
OSPF protocol configuration mode	
redistribute { bgp | connected | static | rip | kernel} [metric-type { 1 | 2 }] [tag <*tag*>] [metric <*cost_value*>] [router-map <WORD>] **no redistribute { bgp | connected | static | rip | kernel }**	Distribute other protocols to find routing and static routings as external routing messages the **no redistribute { bgp | connected | static | rip | kernel }** command cancels the distributed external messages.

（3）Configure OSPF importing the routes of other OSPF processes

1) Enable the function of OSPF importing the routes of other OSPF processes

Command	Explanation
Router ospf mode	
redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}][route-map<*word*>] **no redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}][route-map<*word*>]**	Enable or disable the function of OSPF importing the routes of other OSPF processes.

2) Display relative information

Command	Explanation
Any mode	
show ip ospf [<*process-id*>] redistribute	Display the configuration information of the OSPF process importing other outside routes.

3) Debug

Command	Explanation
Admin mode	
debug ospf redistribute message send **no debug ospf redistribute message send** **debug ospf redistribute route receive** **no debug ospf redistribute route receive**	Enable/disable debugging of sending command from OSPF process redistributed to other ospf process routing. Enable/disable debugging of received routing message from nsm for OSPF process.

（4）Configure other OSPF protocol parameters

1）configure how to calculate OSPF spf algorithm time

2）configure the LSA limit in the OSPF link state database

3）Configure various OSPF parameters

Command	**Explanation**
OSPF protocol configuration mode	
timers spf <*interval*> **no timers spf**	configure the SPF timer of OSPF; the **no timers spf** command restores the default settings
overflow database {<*max-LSA*> [hard | soft] | external <*max-LSA*> <*recover time*>} **no overflow database [external < *max-LSA* > < *recover time* >]**	Configure the LSA limit in current OSPF process database; the **no overflow database [external < *max-LSA* > < *recover time* >]** command restores the default settings.

area <id> {authentication [message-digest] | default-cost <*cost*> | filter-list {access | prefix} <*WORD*> {in | out} | nssa [default-information-originate | no-redistribution | no-summary | translator-role] | range <*range*> | stub [no-summary] | virtual-link <*neighbor*>}
no area <id> {authentication | default-cost | filter-list {access | prefix} <*WORD*> {in | out} | nssa [default-information-originate | no-redistribution | no-summary | translator-role] | range <*range*> | stub [no-summary] | virtual-link <*neighbor*>}
	Configure the parameters in OSPF area (STUB area, NSSA area and virtual links); the **no area <id> {authentication | default-cost | filter-list {access | prefix} <*WORD*> {in | out} | nssa [default-information-originate | no-redistribution | no-summary | translator-role] | range <*range*> | stub [no-summary] | virtual-link <*neighbor*>}** command restores the default settings.

4）Configure the priority of the interface when electing designated layer3 switch (DR).

command	**explanation**
interface configuration mode	
ip ospf priority <*priority*> **no ip ospf priority**	Sets the priority of the interface in “designated layer3 switch” election; the **no ip ospf priority** command restores the default setting.

5）Configure to keep a log for OSPF adjacency changes or not

Command	Explanation
OSPF Protocol Configuration Mode	
log-adjacency-changes detail **no log-adjacency-changes detail**	Configure to keep a log for OSPF adjacency changes or not.

3. Disable OSPF protocol

command	**explanation**
Global mode	
no router ospf [process <*id*>]	Disables OSPF routing protocol

1.6.3 Command For OSPF

1.6.3.1 area authentication

Command: area <*id*> authentication [message-digest]

no area <*id*> authentication

Function: Configure the authentication mode of the OSPF area; the "**no area <*id*> authentication**" command restores the default value.

Parameter: **<*id*>** is the area number which could be shown in digit, ranging between 0～4294967295, or in IP address. **message-digest** is proved by MD5 authentication, or be proved by simple plaintext authentication if not choose this parameter.

Default: No authentication.

Command Mode: OSPF protocol mode.

Usage Guide: Set the authentication mode to plaintext authentication or MD5 authentication. The authentication mode is also configurable under interface mode of which the priority is higher than those in the area. It is required to use ip ospf authentication-key to set the password while no authentication mode configured at the interface and the area is plaintext authentication, and use ip ospf message-digest key command to configure MD5 key if is MD5 authentication. The area authentication mode could not affect the authentication mode of the interface in this area.

Example: Set the authentication mode in area 0 to MD5.

Switch(config-router)#area 0 authentication message-digest

1.6.3.2 area default-cost

Command: area <*id*> default-cost <*cost*>

no area <*id*> default-cost

Function: Configure the cost of sending to the default summary route in stub or NSSA area; the "**no area <*id*> default-cost**" command restores the default value.

Parameter: **<*id*>** is the area number which could be shown as digits 0～4294967295, or as an IP address; **<*cost*>** ranges between <0-16777215>.

Default: Default OSPF cost is 1.

Command Mode: OSPF protocol mode.

Usage Guide: The command is only adaptive to the ABR router connected to the stub area or NSSA area.

Example:

Set the default-cost of area 1 to 10

Switch(config-router)#area 1 default-cost 10

1.6.3.3 area filter-list

Command: [no] area <*id*> filter-list {access|prefix} {in|out}

Function: Configure the filter broadcasting summary routing on the ABR; the "**[no] area <*id*> filter-list {access|prefix} {in|out}**" command restores the default value.

Parameter: **<*id*>** is the area number which could be shown in digits ranging between 0～4294967295, or as an IP address; access-list is appointed for use in access, so is prefix-list for prefix; ***<name>*** is the name of the filter, the length of which is between 1-256; in means from other areas to this area, out means from this area to other areas.

Default: No filter configured.

Command Mode: OSPF protocol mode.

Usage Guide: This command is used for restraining routes from specific area from spreading between this area and other areas.

Example: Set a filter on the area 1.

```
Switch(config)#access-list 1 deny 172.22.0.0 0.0.0.255
Switch(config)#access-list 1 permit any-source
Switch(config)#router ospf 100
Switch(config-router)#area 1 filter-list access 1 in
```

1.6.3.4 area nssa

Command: area <*id*> nssa [TRANSLATOR| no-redistribution |DEFAULT-ORIGINATE |no-summary]

no area <*id*> nssa [TRANSLATOR| no-redistribution |DEFAULT-ORIGINATE |no-summary]

Function: Set the area to Not-So-Stubby-Area (NSSA) area.

Parameter: **<*id*>** is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

TRANLATOR = translator-role {candidate|never|always}, specifies the LSA translation mode for routes: **candidate** means if the router is elected translator, Type 7 LSA can be translated to Type-5 LSA, the default is **candidate.**

never means the router will never translate Type 7 LSA to Type 5 LSA.

always means the route always translate Type 7 LSA to Type 5 LSA.

no-redistribution means never distribute external-LSA to NSSA.

DEFAULT-ORIGINATE=default-information-originate [metric <0-16777214>] [metric-type <*1-2*>], generate the Type-7 LSA.

metric <*0-16777214*> specify the metric value.

metric-type <*1-2*> specifies the metric value type of external-LSA , default value is 2.

no-summary shows not injecting area route to the NSSA.

Default: No NSSA area defined by default.

Command Mode: OSPF protocol mode.

Usage Guide: The same area can not be both NSSA and stub at the same time.

Example: Set area 3 to NSSA.

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#area 0.0.0.51 nssa
```

Switch(config-router)#area 3 nssa default-information-originate metric 34 metric-type 2 translator-role candidate no-redistribution

1.6.3.5 area range

Command: area <*id*> range <*address*> [advertise| not-advertise| substitute]
no area <*id*> range <*address*>

Function: Aggregate OSPF route on the area border. The "**no area <*id*> range <*address*>**" cancels this function.

Parameter: **<*id*>** is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

<*address*>=<*A.B.C.D/M*> specifies the area network prefix and its length.

advertise: Advertise this area, which is the default.

not-advertise : Not advertise this area.

substitute= substitute <*A.B.C.D/M*>: advertise this area as another prefix.

<*A.B.C.D/M*>: Replace the network prefix to be advertised in this area.

Default: Not set.

Command Mode: OSPF protocol mode.

Usage Guide: Use this command to aggregate routes inside an area. If the network IDs in this area are not configured continuously, a summary route can be advertised by configuring this command on ABR. This route consists of all single networks belong to specific range.

Example:

```
Switch # config terminal
Switch (config)# router ospf 100
Switch (config-router)# area 1 range 192.16.0.0/24
```

1.6.3.6 area stub

Command: area <*id*> stub [no-summary]
no area <*id*> stub [no-summary]

Function: Define a area to a stub area. The "**no area <*id*> stub [no-summary]**" command cancels this function.

Parameter: **<*id*>** is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

no-summary: The area border routes stop sending link summary announcement to the stub area.

Default: Not defined.

Command Mode: OSPF protocol mode.

Usage Guide: Configure area stub on all routes in the stub area. There are two configuration commands for the routers in the stub area: stub and default-cost. All routers connected to the stub area should be configured with area stub command. As for area border routers connected to the stub area, their introducing cost is defined with area default-cost command.

Example:

```
Switch # config terminal
Switch (config)# router ospf 100
Switch (config-router)# area 1 stub
```

1.6.3.7 area virtual-link

Command: area <*id*> virtual-link A.B.C.D {AUTHENTICATION|AUTH_KEY|INTERVAL}

no area <*id*> virtual-link A.B.C.D [AUTHENTICATION|AUTH_KEY|INTERVAL]

Function: Configure a logical link between two backbone areas physically divided by non-backbone area. The "**no area <*id*> virtual-link A.B.C.D [AUTHENTICATION|AUTH_KEY|INTERVAL]**" command removes this virtual-link.

Parameter: <*id*> is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

AUTHENTICATION = authentication [message-digest[message-digest-key <1-255> md5 <LINE>] |null|AUTH_KEY].

authentication : Enable authentication on this virtual link.

message-digest: Authentication with MD-5.

null : Overwrite password or packet summary with null authentication.

AUTH_KEY= authentication-key <*key*>.

<key>: A password consists of less than 8 characters.

INTERVAL= [dead-interval | hello-interval | message-digest-key<1-255>md5<LINE> | retransmit-interval | transmit-delay] <value>.

<*value*>:The delay or interval seconds, ranging between 1~65535.

<*dead-interval*>: A neighbor is considered offline for certain dead interval without its group messages which the default is 40 seconds.

<*hello-interval*>: The time interval before the router sends a hello group message, default is 10 seconds.

<message-digest-key>: Authentication key with MD-5.

<*retransmit-interval*>: The time interval before a router retransmitting a group message, default is 5 seconds.

<*transmit-delay*>: The time delay before a router sending a group messages, default is 1 second.

Default: None.

Command Mode: OSPF protocol mode.

Usage Guide: In the OSPF all non-backbone areas will be connected to a backbone area. If the connection to the backbone area is lost, virtual link will repair this connection. You can configure virtual link between any two backbone area routers connected with the public non-backbone area. The protocol treat routers connected by virtual links as a point-to-point network.

Example:

```
Switch#config terminal
Switch(config) #router ospf 100
Switch(config-router) #area 1 virtual-link 10.10.11.50 hello 5 dead 20
```

1.6.3.8 auto-cost reference-bandwidth

Command: auto-cost reference-bandwidth <*bandwith*>
no auto-cost reference-bandwidth

Function: This command sets the way in which OSPF calculate the default metric value. The "**no auto-cost reference-bandwidth**" command only configures the cost to the interface by types.

Parameter: <*bandwith*> reference bandwidth in Mbps, ranging between 1~*4294967.*

Default: Default bandwidth is 100Mbps.

Command Mode: OSPF protocol mode.

Usage Guide: The interface metric value is acquired by divide the interface bandwith with reference bandwidth. This command is mainly for differentiate high bandwidth links. If several high bandwidth links exist, their cost can be assorted by configuring a larger reference bandwidth value.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#auto-cost reference-bandwidth 50
```

1.6.3.9 compatible RFC1583

Command: compatible RFC1583
no compatible RFC1583

Function: This command configures to RFC1583 compatible. The "**no compatible RFC1583**" command close the compatibility.

Default: RFC 2328 compatible by default.

Command Mode: OSPF protocol mode.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#compatible RFC1583
```

1.6.3.10 clear ip ospf process

Command: clear ip ospf [<*process-id*>] process

Function: Use this command to clear and restart OSPF routing processes. One certain OSPF process will be cleared by specifying the process ID, or else all OSPF processes will be cleared.

Default: No default configuration.

Command Mode: Admin mode.

Example:

```
Switch#clear ip ospf process
```

1.6.3.11 distance

Command: distance {<*value*>|ROUTEPARAMETER}

no distance ospf

Function: Configure OSPF manage distance base on route type. The "**no distance ospf**" command restores the default value.

Parameter: <*value*>, OSPF routing manage distance, ranging between 1~235.

ROUTEPARAMETER= ospf {ROUTE1|ROUTE2|ROUTE3}.

ROUTE1= external <external-distance>, Configure the distance learnt from other routing area.

<external-distance>distance value, ranging between 1~255.

ROUTE2= inter-area <inter-distance>, configure the distance value from one area to another area.

<inter-distance> manage distance value, ranging between 1~255.

ROUTE3= intra-area <intra-distance> Configure all distance values in one area.

<intra-distance> Manage distance value, ranging between 1~255.

Default: Default distance value is 110.

Command Mode: OSPF protocol mode.

Usage Guide: Manage distance shows the reliability of the routing message source. The distance value may range between 1~255. the larger the manage distance value is, the lower is its reliability.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#distance ospf inter-area 20 intra-area 10 external 40
```

1.6.3.12 distribute-list

Command: distribute-list <*access-list-name*> out {kernel |connected| static| rip| isis| bgp}

[no] distribute-list out {kernel |connected| static| rip| isis| bgp}

Function: Filter network in the routing update. The "**[no] distribute-list out {kernel |connected| static| rip| isis| bgp}**" command disables this function.

Parameter: < *access-list-name*> is the access-list name to be applied.

out: Filter the sent route update.

kernel Kernel route.

connected Direct route.

static Static route.

rip RIP route.

isis ISIS route.

bgp BGP route.

Default: None.

Command Mode: OSPF protocol mode.

Usage Guide: When distributing route from other routing protocols into the OSPF routing table, we can use this command.

Example: Example below is the advertisement based on the access-list list 1 of the BGP route.

```
Switch#config terminal
Switch(config)#access-list l1 permit 172.10.0.0 0.0.255.255
Switch(config)#router ospf 100
Switch(config-router)#distribute-list 1 out bgp
Switch(config-router)#redistribute bgp
```

1.6.3.13 host area

Command: host <*host-address*> area <*area-id*> [cost <*cost*>]

no host <*host-address*> area <*area-id*> [cost <*cost*>]

Function: Use this command to set a stub host entire belongs to certain area. The "**no host <*host-address*> area <*area-id*> [cost <*cost*>]**" command cancels this configuration.

Parameter: ***<host-address>*** is host IP address show in dotted decimal notation,

<area-id> area ID shown in dotted decimal notation or integer ranging between 0~4294967295.

<cost> specifies the entire cost, which is a integer ranging between 0~65535 and defaulted at 0.

Default: No entire set.

Command Mode: OSPF protocol mode.

Usage Guide: With this command you can advertise certain specific host route out as stub link. Since the stub host belongs to special router in which setting host is not important.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#host 172.16.10.100 area 1
Switch(config-router)#host 172.16.10.101 area 2 cost 10
```

1.6.3.14 ip ospf authentication

Command: ip ospf [<*ip-address*>] authentication [message-digest|null]

no ip ospf [<*ip-address*>] authentication

Function: Specify the authentication mode required in sending and receiving OSPF packets on the interfaces; the "**no ip ospf [<*ip-address*>] authentication**" command cancels the authentication.

Parameter: <*ip-address*> is the interface IP address, shown in dotted decimal notation.

message-digest: Use MD5 authentication.

null: no authentication applied, which resets the password or MD5 authentication applied on the interface.

Default: Authentication not required in receiving OSPF packets on the interface.

Command Mode: Interface Mode.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf authentication message-digest
```

1.6.3.15 ip ospf authentication-key

Command: ip ospf [<*ip-address*>] authentication-key <LINE>

no ip ospf [<*ip-address*>] authentication

Function: Specify the authentication key required in sending and receiving OSPF packet on the interface; the "**no ip ospf [<*ip-address*>] authentication**" cancels the authentication key.

Parameter: <*ip-address*> is the interface IP address shown in dotted decimal notation; **<LINE>** specifies the key required in the plaintext authentication.

Default: Authentication not required in receiving OSPF packets on the interface.

Command Mode: Interface Mode.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf authentication-key password
```

1.6.3.16 ip ospf cost

Command: ip ospf [<*ip-address*>] cost <*cost*>

no ip ospf [<*ip-address*>] cost

Function: Specify the cost required in running OSPF protocol on the interface; the "**no ip ospf [<*ip-address*>] cost**" command restores the default value.

Parameter: <*ip-address*> is the interface IP address shown in dotted decimal notation.

<*cost* > is the cost of OSPF protocol ranging between 1~65535.

Default: Default OSPF cost on the interface is auto-figure out based bandwidth.

Command Mode: Interface Mode.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf cost 3
```

1.6.3.17 ip ospf database-filter

Command: ip ospf [<*ip-address*>] database-filter all out

no ip ospf [<*ip-address*>] database-filter

Function: The command opens LSA database filter switch on specific interface; the "**no ip ospf [<*ip-address*>] database-filter**" command closes the filter switch.

Parameter: <*ip-address*> is the interface IP address shown in dotted decimal notation.

all: All LSAs.

out: Sent LSAs.

Default: Filter switch Closed.

Command Mode: Interface Mode.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf database-filter all out
```

1.6.3.18 ip ospf dead-interval

Command: ip ospf [<*ip-address*>] dead-interval <*time* >

no ip ospf [<*ip-address*>] dead-interval

Function: Specify the dead interval for neighboring layer 3 switch; the "**no ip ospf [<*ip-address*>] dead-interval**" command restores the default value.

Parameter: <*ip-address*> is the interface IP address shown in dotted decimal notation.

<*time* > is the dead interval length of the neighboring layer 3 switches, shown in seconds and ranging between 1~65535.

Default: The default dead interval is 40 seconds (normally 4 times of the hellow-interval).

Command Mode: Interface Mode.

Usage Guide: If no Hello data packet received after the **dead-interval** period then this layer 3 switch is considered inaccessible and invalid. This command modifies the dead interval value of neighboring layer 3 switch according to the actual link state. The set **dead-interval** value is written into the Hello packet and transmitted. To ensure the normal operation of the OSPF protocol, the dead-interval between adjacent layer 3 switches should be in accordance or at least 4 times of the **hello-interval** value.

Example:

```
Switch#config terminal
```

```
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf dead-interval 80
```

1.6.3.19 ip ospf disable all

Command: ip ospf disable all

no ip ospf disable all

Function: Stop OSPF group process on the interface.

Command Mode: Interface Mode.

Usage Guide: This command resets the network area command and stops group process on specific interface.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf disable all
```

1.6.3.20 ip ospf hello-interval

Command: ip ospf [<*ip-address*>] hello-interval <*time*>

no ip ospf [<*ip-address*>] hello-interval

Function: Specify the hello-interval on the interface; the "**no ip ospf [<*ip-address*>] hello-interval**" restores the default value.

Parameter: <*ip-address*> is the interface IP address shown in dotted decimal notation;

<*time*> is the interval sending HELLO packet, shown in seconds and ranging between 1～65535.

Default: The hello-interval on the interface is 10 seconds.

Command Mode: Interface Mode.

Usage Guide: HELLO data packet is the most common packet which is periodically sent to adjacent layer 3 switch to discover and maintain adjacent relationship, elect DR and BDR. The user set **hello-interval** value will be written into the HELLO packet and transmitted. The less the **hello-interval** value is, the sooner the network topological structure is discovered as well larger the cost. The ensure the normal operation of OSPF protocol the **hello-interval** parameter between the layer 3 switches adjacent to the interface must be in accordance.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf hello-interval 20
```

1.6.3.21 ip ospf message-digest-key

Command: ip ospf [<*ip-address*>] message-digest-key <*key_id*> MD5 <*LINE*>

no ip ospf [<*ip-address*>] message-digest-key <*key_id*>

Function: Specify the key id and value of MD5 authentication on the interface; the "**no ip ospf [<*ip-address*>] message-digest-key <*key_id*>**" restores the default value.

Parameter: *<ip-address>* is the interface IP address show in dotted decimal notation; ***<key_id>*** ranges between 1-255; ***<LINE>*** is the OSPF key.

Default: MD5 key not configured.

Command Mode: Interface Mode.

Usage Guide: MD5 key encrypted authentication is used for ensure the safety between the OSPF routers on the network. Same key id and key should be configured between neighbors when using this command or else no adjacent relationship will not be created. The last configuration of this command will overwrite the previous one to prevent the system from communicating with the former key id.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf message-digest-key 2 MD5 yourpassword
```

1.6.3.22 ip ospf mtu

Command: ip ospf mtu <*mtu*>

no ip ospf mtu

Function: Specify the mtu value of the interface as the OSPF group structure according; the "**no ip ospf mtu**" command restores the default value.

Parameter: *<mtu >* is the interface mtu value ranging between 576~65535.

Default: Use the interface mtu acquired from the kernel.

Command Mode: Interface Mode.

Usage Guide: The interface value configured by this command is only used by OSPF protocol other than updated into kernel.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf mtu 1480
```

1.6.3.23 ip ospf mtu-ignore

Command: ip ospf <*ip-address*> mtu-ignore

no ip ospf <*ip-address*> mtu-ignore

Function: Use this command so that the mtu size is not checked when switching DD; the "**no ip ospf <*ip-address*> mtu-ignore**" will ensure the mtu size check when performing DD switch.

Parameter: <*ip-address*> is the interface IP address show in dotted decimal notation.

Default: Check mtu size in DD switch.

Command Mode: Interface Mode.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf mtu-ignore
```

1.6.3.24 ip ospf network

Command: ip ospf network {broadcast|non-broadcast|point-to-point|point-to-multipoint}
no ip ospf network

Function: This command configure the OSPF network type of the interface; the "**no ip ospf network**" command restores the default value.

Parameter: broadcast: Set the OSPF network type to broadcast.

non-broadcast: Set the OSPF network type to NBMA.

point-to-point: Set the OSPF network type to point-to-point.

point-to-multipoint: Set the OSPF network type to point-to-multipoint.

Default: The default OSPF network type is broadcast.

Command Mode: Interface Mode.

Example: The configuration below set the OSPF network type of the interface vlan 1 to point-to-point.

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf network point-to-point
```

1.6.3.25 ip ospf priority

Command: ip ospf [<*ip-address*>] priority <*priority*>
no ip ospf [<*ip-address*>] priority

Function: Configure the priority when electing "Defined layer 3 switch" at the interface. The "**no ip ospf [<*ip-address*>] priority**" command restores the default value.

Parameter: **<*ip-address*>** is the interface IP address show in dotted decimal notation
<*priority*> is the priority of which the valid value ranges between 0～255.

Default: The default priority when electing DR is 1.

Command Mode: Interface Mode.

Usage Guide: When two layer 3 switches connected to the same segments both want to be the "Defined layer 3 switch", the priority will decide which one should be chosen. Normally the one with higher priority will be elected, or the one with larger router-id number if the priorities are the same. A layer 3 switch with a priority equal to 0 will not be elected as "Defined layer 3 switch" or

"Backup Defined layer 3 switch".

Example: Configure the priority of DR electing. Configure the interface vlan 1 to no election right, namely set the priority to 0.

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf priority 0
```

1.6.3.26 ip ospf retransmit-interval

Command: ip ospf [<*ip-address*>] retransmit-interval <*time*>
no ip ospf [<*ip-address*>] retransmit-interval

Function: Specify the retransmit interval of link state announcements between the interface and adjacent layer 3 switches. The "**no ip ospf [<*ip-address*>] retransmit-interval**" command restores the default value.

Parameter: <*ip-address*> is the interface IP address show in dotted decimal notation.

<*time*> is the retransmit interveral of link state announcements between the interface and adjacent layer 3 switches, shown in seconds ang raning between 1～65535.

Default: Default retransmit interval is 5 seconds.

Command Mode: Interface Mode.

Usage Guide: When a layer 3 switch transmits LSA to its neighbor, it will maintain the link state announcements till confirm from the object side is received. If the confirm packet is not received within the interval, the LSA will be retransmitted. The retransmit interval must be larger than the time it takes to make a round between two layer 3 switches.

Example: Configure the LSA retransmit interval of interface vlan 1 to 10 seconds.

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf retransmit-interval 10
```

1.6.3.27 ip ospf transmit-delay

Command: ip ospf [<*ip-address*>] transmit-delay <*time*>
no ip ospf [<*ip-address*>] transmit-delay

Function: Set the transmit delay value of LSA transmitting; the "**no ip ospf [<*ip-address*>] transmit-delay**" restores the default value.

Parameter: <*ip-address*> is the interface IP address show in dotted decimal notation.

<*time*> is the transmit delay value of link state announcements between the interface and adjacent layer 3 switches, shown in seconds ang raning between 1～65535.

Default: Default transmit delay value of link state announcements is 1 second.

Command Mode: Interface Mode.

Usage Guide: The LSA ages with time in the layer 3 switches, but not in the network transmitting process. By adding the **transit-delay** prior to sending the LSA, the LSA will be sent before aged.

Example: Set the LSA transmit delay of interface vlan1 to 3 seconds.

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip ospf transmit-delay 3
```

1.6.3.28 key

Command: key <*keyid*>

no key <*keyid*>

Function: This command is for managing and adding keys in the key chain. The “**no key <*keyid*>**“ command deletes one key.

Parameter: <*keyid*> is key ID, ranging between 0-2147483647.

Command Mode: Keychain mode

Usage Guide: The command permits entering the keychain-key mode and set the passwords corresponding to the keys.

Example: Switch#config terminal

```
Switch(config)#key chain mychain
Switch(config-keychain)#key 1
Switch(config-keychain-key)#
```

1.6.3.29 key chain

Command: key chain <*name-of-chain*>

no key chain < *name-of-chain* >

Function: This command is for entering a keychain manage mode and configure a keychain. The “**no key chain < *name-of-chain* >**“ delete one keychain.

Parameter: <*name-of-chain*> is the name string of the keychain the length of which is not specifically limited.

Command Mode: Global Mode

Example: Switch#config terminal

```
Switch(config)#key chain mychain
Switch(config-keychain)#
```

1.6.3.30 log-adjacency-changes detail

Command：log-adjacency-changes detail

no log-adjacency-changes detail

Function：Configure to keep a log for OSPF adjacency changes or not.

Parameter: None.

Default: Don't l keep a log for OSPF adjacency changes by default.

Command Mode: OSPF Protocol Configuration Mode

Usage Guide: When this command is configured, the OSPF adjacency changes information will be recorded into a log.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#log-adjacency-changes detail
```

1.6.3.31 max-concurrent-dd

Command: max-concurrent-dd <*value*>
no max-concurrent-dd

Function: This command set the maximum concurrent number of dd in the OSPF process; the "**no max-concurrent-dd**" command restores the default.

Parameter: <*value*> ranges between ***<1-65535>***, which is the capacity of processing the concurrent dd data packet.

Default: Not set, no concurrent dd limit.

Command Mode: OSPF protocol mode.

Usage Guide: Specify the max concurrent number of dd in the OSPF process.

Example: Set the max concurrent dd to 20.

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#max-concurrent-dd 20
```

1.6.3.32 neighbor

Command: neighbor A.B.C.D [<*cost*>| priority <*value*> | poll-interval <*value*>]
no neighbor A.B.C.D [<*cost*>| priority <*value*> | poll-interval <*value*>]

Function: This command configures the OSPF router connecting NBMA network. the "**no neighbor A.B.C.D [<*cost*>| priority <*value*> | poll-interval <*value*>]**" command removes this configuration.

Parameter: <*cost*>, OSPF neighbor cost value ranging between 1-65535; **priority <*value*>** , neighbor priority defaulted at 0 and ranges between 0-255; **poll-interval <*value*>**, 120s by default, which the polling time before neighbor relationship come into shape , ranging between 1-65535.

Default: No default configuration.

Command Mode: OSPF protocol mode.

Usage Guide: Use this command on NBMA network to configure neighbor manually. Every known non-broadcasting neighbor router should be configured with a neighbor entry. The configured neighbor address should be the main address of the interface. The poll-interval should be much larger than the hello-interval.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#neighbor 1.2.3.4 priority 1 poll-interval 90
Switch(config-router)#neighbor 1.2.3.4 cost 15
```

1.6.3.33 network area

Command: network NETWORKADDRESS area <*area-id*>

no network NETWORKADDRESS area <*area-id*>

Function: This command enables OSPF routing function one the interface with IP address matched with the network address. The "**no network NETWORKADDRESS area <*area-id*>**" command removes the configuration and stop OSPF on corresponding interface.

Parameter: NETWORKADDRESS = A.B.C.D/M | A.B.C.D X.Y.Z.W, Shown with the network address prefix or the mask. Wildcast mask if shown in mask; <***area-id***> is the ip address or area number shown in point divided demical system, if shown in demcial integer, it ranges between 0~4294967295.

Default: No default.

Command Mode: OSPF protocol mode.

Usage Guide: When certain segment belongs to certain area, interface the segment belongs will be in this area, starting hello and database interaction with the connected neighbor.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#network 10.1.1.0/24 area 1
```

1.6.3.34 ospf abr-type

Command: ospf abr-type {cisco|ibm|shortcut|standard}

no ospf abr-type

Function: Use this command to configure a OSPF ABR type. The "**no ospf abr-type**" command restores the default value.

Parameter: cisco, Realize through cisco ABR; **ibm**, Realize through ibm ABR; **shortcut**, Specify a shortcut-ABR; **standard**, Realize with standard（RFC2328）ABR.

Default: Cisco by default.

Command Mode: OSPF protocol mode.

Usage Guide: For Specifying the realizing type of abr. This command is good for interactive operation among different OSPF realizing method and is especially useful in the multiple host environment.

Example: Configure abr as standard.

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#ospf abr-type standard
```

1.6.3.35 ospf router-id

Command: ospf router-id <*address*>

no ospf router-id

Function: Specify a router ID for the OSPF process. The "**no ospf router-id**" command cancels the ID number.

Parameter: <*address*>, IPv4 address format of router-id.

Default: No default configuration.

Command Mode: OSPF protocol mode.

Usage Guide: The new router-id takes effect immediately.

Example: Configure router-id of ospf 100 to 2.3.4.5.

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#ospf router-id 2.3.4.5
```

1.6.3.36 overflow database

Command: overflow database <*maxdbsize* > [{hard|soft}]

no overflow database

Function: This command is for configuring the max LSA number. The "**no overflow database**" command cancels the limit.

Default: Not configured.

Parameter: < ***maxdbsize*** >Max LSA numbers, ranging between 0~4294967294

soft: Soft limit, warns when border exceeded.

hard: Hard limit, directly close ospf instance when border exceeded.

If there is not soft or hard configured, the configuration is taken as hard limit.

Command Mode: OSPF protocol mode.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#overflow database 10000 soft
```

1.6.3.37 overflow database external

Command: overflow database external [<*maxdbsize* > <*maxtime*>]

no overflow database external [<*maxdbsize* > <*maxtime*>]

Function: The command is for configuring the size of external link database and the waiting time before the route exits overflow state. The "**no overflow database external [<*maxdbsize* > <*maxtime*>]**" restores the default value.

Parameter: < ***maxdbsize*** > size of external link database, ranging between 0~4294967294 , defaulted at 4294967294.

< *maxtime* > the seconds the router has to wait before exiting the database overflow, ranging between 0~65535.

Command Mode: OSPF protocol mode.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#overflow database external 5 3
```

1.6.3.38 passive-interface

Command: passive-interface<*ifname*>

no passive-interface<*ifname*>

Function: Configure that the hello group not sent on specific interfaces. The "**no passive-interface<*ifname*>**" command cancels this function.

Parameter: <***ifname***> is the specific name of interface.

<*ip-address*> IP address of the interface, shown in dotted decimal notation.

Default: Not configured.

Command Mode: OSPF protocol mode.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#passive-interface vlan1
```

1.6.3.39 redistribute

Command: redistribute {kernel |connected| static| rip| isis| bgp} [metric<*value*>] [metric-type {1|2}][route-map<*word*>][tag<*tag-value*>]

no redistribute {kernel |connected| static| rip| isis| bgp} [metric<*value*>] [metric-type {1|2}][route-map<*word*>][tag<*tag-value*>]

Function: Introduce route learnt from other routing protocols into OSPF.

Parameter: kernel introduce from kernel route.

connected introduce from direct route.

static introduce from static route.

rip introduce from the RIP route.

isis introduce from ISIS route.

bgp introduce from BGP route.

metric <*value*> is the introduced metric value, ranging between 0-16777214.

metric-type {1|2} is the metric value type of the introduced external route, which can be 1 or 2, and it is 2 by default.

route-map <*word*> point to the probe of the route map for introducing route.

tag<*tag-value*> external identification number of the external route, ranging between 0~4294967295, defaulted at 0.

Command Mode: OSPF protocol mode.

Usage Guide: Learn and introduce other routing protocol into OSPF area to generate AS-external_LSAs.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#redistribute bgp metric 12
```

1.6.3.40 redistribute ospf

Command: redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>]

no redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>]

Function: To redistribute of process id routing to this process. The no command is to deletes the redistribution of process id routing to this process. When input the optional parameters of metric, metric type and routermap, then restores default configuration.

Parameter: process-id is OSPF process id, 0 by default.

metric <*value*> is the metric for redistributed routing, range between 0 to 16777214.

metric-type {1|2} is the metric type for redistributed routing, only can be 1 or 2, and 2 by default.

route-map <*word*> is the pointer to the introduced routing map.

Default: Not redistributed any OSPF routing by default.

Command Mode: OSPF protocol mode.

Usage Guide: When process-id is not input, that means OSPF routing will be redistributed by default. (Process-id is 0)

Example:

```
Switch(config-router)#redistribute ospf
```

1.6.3.41 router ospf

Command: router ospf <process_id> <*vrf-name*>

no router ospf <process_id> <*vrf-name*>

Function: This command is for relating the OSPF process and a specific VPN. All configuration commands will be related to this VPN after the configuration succeeded. The "**no router ospf <process_id> <*vrf-name*>**" command deletes the VPN routing/forwarding instance related OSPF instances.

Parameter: <*process_id*> specifies the id of the OSPF process to be created. **<*vrf-name*>** specifies the name of VPN routing/forwarding instance.

Command Mode: Global mode.

Usage Guide: This command is only used for PE router. A VPN routing/forwarding instance should be generated with ip vrf command before using this command, then with this command you can relate OSPF instances to this VPN routing/forwarding instance.

Example:

```
Switch# config terminal
Switch(config)# router ospf 100 VRF1
Switch(config-router)#network 10.1.1.0/24 area 0
```

1.6.3.42 default-information originate

Command: default-information originate [always|METRIC|METRICTYPE|ROUTEMAP]

no default-information originate

Function: This command create a default external route to OSPF route area; the "**no default-information originate**" closes this feature.

Parameter: always: Whether default route exist in the software or not, the default route is always advertised.

METRIC = metric <*value*>: Set the metric value for creating default route, **<*value*>** ranges between 0~*16777214* , default metric value is 0.

METRICTYPE = metric-type {1|2} set the OSPF external link type of default route.

1 Set the OSPF external type 1 metric value.

2 Set the OSPF external type 2 metric value.

ROUTEMAP = route-map <*WORD*>.

<WORD> specifies the route map name to be applied.

Default: Default metric value is 10, default OSPF external link type is 2.

Command Mode: OSPF protocol mode.

Usage Guide: When introducing route into OSPF route area with this command , the system will behaves like an ASBR.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#default-information originate always metric 23 metric-type 2 route-map
myinfo
```

1.6.3.43 default-metric

Command: default-metric <*value*>

no default-metric

Function: The command set the default metric value of OSPF routing protocol; the "**no default-metric**" returns to the default state.

Parameter: <*value*>, metric value, ranging between 0~*16777214.*

Default: Built-in, metric value auto translating.

Command Mode: OSPF protocol mode.

Usage Guide: When the default metric value makes the metric value not compatible, the route introducing still goes through. If the metric value can not be translated, the default value provides alternative option to carry the route introducing on. This command will result in that all introduced route will use the same metric value. This command should be used associating redistribute.

Example:

```
Switch#config terminal
Switch(config)#router ospf 100
Switch(config-router)#default-metric 100
```

1.6.3.44 summary-address

Command: summary-address <*A.B.C.D/M*> [{not-advertise|tag<*tag-value*>}]

Function: Summarize or restrain external route with specific address scope.

Parameter: <*A.B.C.D/M*> address scope, shown in dotted decimal notation IPv4 address plus mask length.

not-advertised restrain the external routes.

tag<*tag-value*> is the identification label of the external routes, which ranges between 0~4294967295, and is defaulted at 0.

Command Mode: OSPF protocol mode.

Usage Guide: When routes are introduced into OSPF from other routing protocols, it is required to advertise every route in a external LSA. This command is for advertise one summary route for those introduced routes contained in specific network address and masks, which could greatly reduces the size of the link state database.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#summary-address 172.16.0.0/16 tag 3
```

1.6.3.45 timers spf

Command: timers spf <*spf-delay*> <*spf-holdtime*>

no timers spf

Function: Adjust the value of the route calculating timer. The "**no timers spf**" command restores relevant values to default.

Parameter: <*spf-delay*> 5 seconds by default.

<*spf-holdtime*> 10 seconds by default.

Command Mode: OSPF protocol mode.

Usage Guide: This command configures the delay time between receiving topology change and SPF calculation, further configured the hold item between two discontinuous SPF calculation.

Example:

```
Switch#config terminal
Switch(config)#router ospf
Switch(config-router)#timers spf 5 10
```

1.6.4 OSPF Example

1.6.4.1 Configuration Example of OSPF

Scenario 1: OSPF autonomous system.

This scenario takes an OSPF autonomous system consists of five ES4700 series layer3 switches for example, where layer3 SwitchC SwitchA and SwitchE make up OSPF area 0, layer3 switch SwitchB and SwitchC form OSPF area 1 (assume vlan1 interface of layer3 switch SwitchA belongs to area 0), layer3 switch SwitchD forms OSPF area 2 (assume vlan2 interface of layer3 SwitchE belongs to area 0). SwtichA and SwitchE are backbone layer3 switches, SwtichB and SwitchD are area border layer3 switches, and SwitchC is the inside-area layer3 switch.

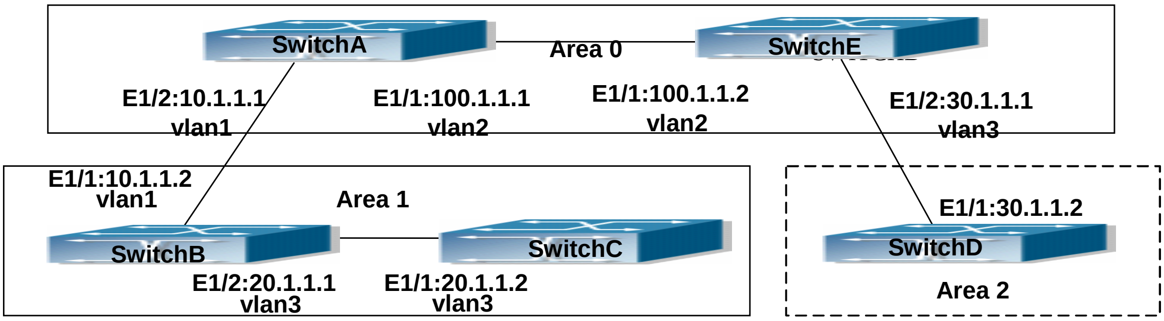

Fig 1-8 Network topology of OSPF autonomous system.

The configuration for layer3 switch SwitchA and SwitchE is shown below:

Layer 3 switch switchA

Configuration of the IP address for interface vlan1

```
SwitchA#config
SwitchA(config)# interface vlan 1
SwitchA(config-if-vlan1)# ip address 10.1.1.1 255.255.255.0
SwitchA(config-if-vlan1)#exit
```

Configuration of the IP address for interface vlan2

Configure the IP address of interface vlan2

```
SwitchA(config)# interface vlan 2
SwitchA(config-if-vlan2)# ip address 100.1.1.1 255.255.255.0
SwitchA (config-if-vlan2)#exit
```

Enable OSPF protocol, configure the area number for interface vlan1 and vlan2.

```
SwitchA(config)#router ospf
SwitchA(config-router)#network 10.1.1.0/24 area 0
SwitchA(config-router)#network 100.1.1.0/24 area 0
SwitchA(config-router)#exit
SwitchA(config)#exit
```

Layer 3 switch SwitchB

Configure the IP address for interface vlan1 and vlan2Configure the IP address of interface vlan1 and vlan2

```
SwitchB#config
SwitchB(config)# interface vlan 1
SwitchB(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB(config-if-vlan1)#exit
SwitchB(config)# interface vlan 3
SwitchB(config-if-vlan3)# ip address 20.1.1.1 255.255.255.0
SwitchB(config-if-vlan3)#exit
```

Enable OSPF protocol, configure the OSPF area interfaces vlan1 and vlan3 in

```
SwitchB(config)#router ospf
SwitchB(config-router)# network 10.1.1.0/24 area 0
SwitchB(config-router)# network 20.1.1.0/24 area 1
SwitchB(config-router)#exit
SwitchB(config)#exit
```

Layer 3 switch SwitchC

Configuration of the IP address for interface vlan3

```
SwitchC#config
SwitchC(config)# interface vlan 3
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC(config-if-vlan3)#exit
```

Enable OSPF protocol, configure the OSPF area interfaces vlan3 resides in.

Initiate the OSPF protocol, configure the OSPF area to which interface vlan3 belongs

```
SwitchC(config)#router ospf
SwitchC(config-router)# network 20.1.1.0/24 area 1
SwitchC(config-router)#exit
SwitchC(config)#exit
```

Layer 3 switch SwitchD

Configuration of the IP address for interface vlan3

```
SwitchD#config
SwitchD(config)# interface vlan 3
SwitchD(config-if-vlan3)# ip address30.1.1.2 255.255.255.0
SwitchD(config-if-vlan3)#exit
```

Enable OSPF protocol, configure the OSPF area interfaces vlan3 resides in. SwitchD(config)#router ospf

```
SwitchD(config-router)# network 30.1.1.0/24 area 0
SwitchD(config-router)#exit
SwitchD(config)#exit
```

Layer 3 switch SwitchE

Configuration of the IP address for interface vlan2

```
SwitchE#config
SwitchE(config)# interface vlan 2
SwitchE(config-if-vlan2)# ip address 100.1.1.2 255.255.255.0
SwitchE(config-if-vlan2)#exit
```

Configuration of the IP address for interface vlan3

```
SwitchE(config)# interface vlan 3
SwitchE(config-if-vlan3)# ip address 30.1.1.1 255.255.255.0
SwitchE(config-if-vlan3)#exit
```

Enable OSPF protocol, configure the number of the area in which interface vlan2 and vlan3 reside in.

```
SwitchE(config)#router ospf
SwitchE(config-router)# network 30.1.1.0/24 area 0
SwitchE(config-router)# network 100.1.1.0/24 area 0
SwitchE(config-router)#exit
SwitchE(config)#exit
```

Scenario 2: Typical OSPF protocol complex topology.

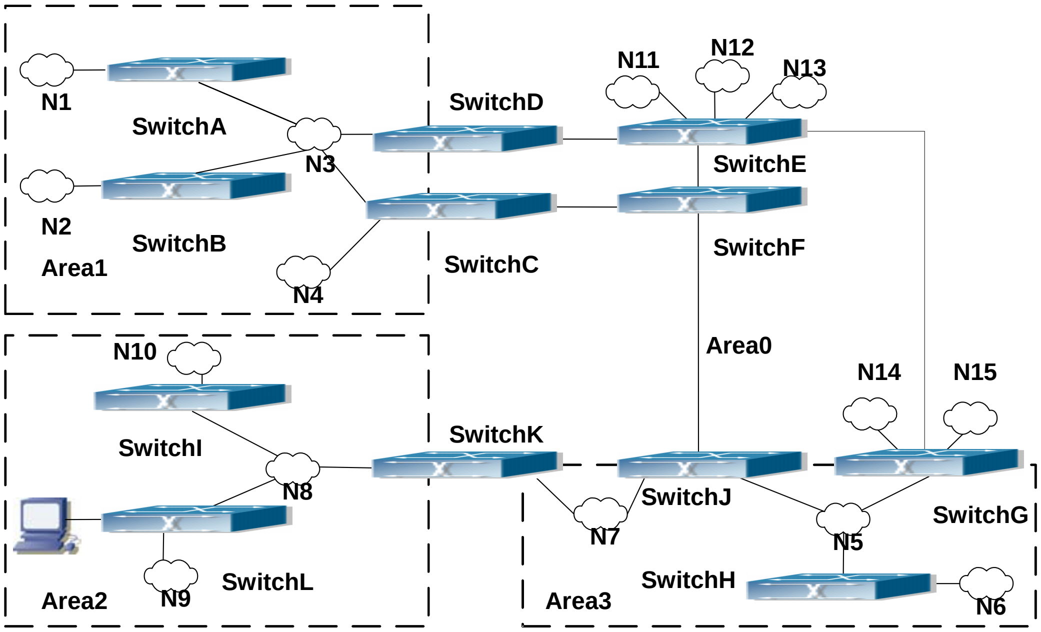

Fig　1-9 Typical complex OSPF autonomous system.

This scenario is a typical complex OSPF autonomous system network topology. Area1 include network N1-N4 and layer3 switch SwitchA-SwitchD, area2 include network N8-N10, host H1 and layer3 switch SwitchH, area3 include N5-N7 and layer3 switch SwitchG, SwitchG SwitchJ and SwitchK, and network N8-N10 share a summary route with host H1(i.e. area3 is defined as a STUB area). Layer3 switch SwitchA, SwitchB, SwitchE, SwitchF, SwitchG, SwitchH, SwitchL are in-area layer3 switches, SwitchC, SwitchD, SwitchG, SwitchJ and SwitchK are edge layer3 switches of the area, SwitchE and SwitchG are edge layer3 switches of the autonomous system.

To area1, layer3 switches SwitchA and SwitchB are both in-area switches, area edge switches SwitchC and SwitchD are responsible for reporting distance cost to all destination outside the area, while they are also responsible for reporting the position of the AS edge layer3 switches SwitchE and SwitchG, AS external link-state advertisement from SwitchE and SwitchG are flooded throughout the whole autonomous system. When ASE LSA floods in area 1, those LSAs are included in the area 1 database to get the routes to network N11 and N15.

In addition, layer3 switch SwitchC and SwitchD must summary the topology of area 1 to the backbone area (area 0, all non-0 areas must be connected via area 0, direct connections are not allowed), and advertise the networks in area 1 (N1-N4) and the costs from SwitchC and SwitchD to those networks. As the backbone area is required to keep connected, there must be a virtual link between backbone layer3 switch SwitchJ and SwitchK. The area edge layer3 switches exchange summary information via the backbone layer3 switch, each area edge layer3 switch listens to the summary information from the other edge layer3 switches.

Virtual link can not only maintain the connectivity of the backbone area, but also strengthen the backbone area. For example, if the connection between backbone layer3 switch SwitchG and SwitchJ is cut down, the backbone area will become incontinuous. The backbone area can become more robust by establishing a virtual link between backbone layer3 switches SwitchG and SwitchJ. In addition, the virtual link between SwitchG and SwitchJ provide a short path from area 3 to layer3 switch SwitchG.

Take area 1 as an example. Assume the IP address of layer3 switch SwitchA is 10.1.1.1, IP address of layer3 switch SwitchB interface VLAN2 is 10.1.1.2, IP address of layer3 switch SwitchC interface VLAN2 is 10.1.1.3, IP address of layer3 switch SwitchD interface VLAN2 is 10.1.1.4. SwitchA is connecting to network N1 through Ethernet interface VLAN1 (IP address 20.1.1.1); SwitchB is connecting to network N2 through Ethernet interface VLAN1 (IP address 20.1.2.1); SwitchC is connecting to network N4 through Ethernet interface VLAN3 (IP address 20.1.3.1). All the three addresses belong to area 1. SwitchC is connecting to layer3 switch SwitchF through Ethernet interface VLAN1 (IP address 10.1.5.1); SwitchD is connecting to layer3 switch SwitchE through Ethernet interface VLAN1 (IP address 10.1.6.1); both two addresses belong to area 1. Simple authentication is implemented among layer3 switches in area1, edge layer3 switches of area 1 authenticate with the area 0 backbone layer3 switches by MD5 authentication.

The followings are just configurations for all layer3 switches in area 1, configurations for layer3 switches of the other areas are omitted. The following are the configurations of SwitchA SwitchB.SwitchC and SwitchD:

1)SwitchA:

Configure IP address for interface vlan2

```
SwitchA#config
SwitchA(config)# interface vlan 2
SwitchA(config-If-Vlan2)# ip address 10.1.1.1 255.255.255.0
SwitchA(config-If-Vlan2)#exit
```

Enable OSPF protocol, configure the area number for interface vlan2.

```
SwitchA(config)#router ospf
SwitchA(config-router)#network 10.1.1.0/24 area 1
SwitchA(config-router)#exit
```

Configure simple key authentication.

```
SwitchA(config)#interface vlan 2
SwitchA(config-If-Vlan2)#ip ospf authentication
SwitchA(config-If-Vlan2)#ip ospf authentication-key DCS
SwitchA(config-If-Vlan2)exit
```

Configure IP address and area number for interface vlan1.

```
SwitchA(config)# interface vlan 1
SwitchA(config-If-Vlan1)#ip address 20.1.1.1 255.255.255.0
SwitchA(config-If-Vlan1)#exit
SwitchA(config)#router ospf
SwitchA(config-router)#network 20.1.1.0/24 area 1
SwitchA(config-router)#exit
```

2)SwitchB:

Configure IP address for interface vlan2

```
SwitchB#config
SwitchB(config)# interface vlan 2
SwitchB(config-If-Vlan2)# ip address 10.1.1.2 255.255.255.0
SwitchB(config-If-Vlan2)#exit
```

Enable OSPF protocol, configure the area number for interface vlan2.

```
SwitchB(config)#router ospf
SwitchB(config-router)#network 10.1.1.0/24 area 1
SwitchB(config-router)#exit
SwitchB(config)#interface vlan 2
```

Configure simple key authentication.

```
SwitchB(config)#interface vlan 2
SwitchB(config-If-Vlan2)#ip ospf authentication
SwitchB(config-If-Vlan2)#ip ospf authentication-key DCS
SwitchB(config-If-Vlan2)#exit
```

Configure IP address and area number for interface vlan1.

```
SwitchB(config)# interface vlan 1
SwitchB(config-If-Vlan1)#ip address 20.1.2.1 255.255.255.0
SwitchB(config-If-Vlan1)#exit
SwitchB(config)#router ospf
SwitchB(config-router)#network 20.1.2.0/24 area 1
SwitchB(config-router)#exit
SwitchB(config)#exit
SwitchB#
```

3)SwitchC:

Configure IP address for interface vlan2

```
SwitchC#config
SwitchC(config)# interface vlan 2
SwitchC(config-If-Vlan2)# ip address 10.1.1.3 255.255.255.0
SwitchC(config-If-Vlan2)#exit
```

Enable OSPF protocol, configure the area number for interface vlan2

```
SwitchC(config)#router ospf
SwitchC(config-router)#network 10.1.1.0/24 area 1
SwitchC(config-router)#exit
```

Configure simple key authentication

```
SwitchC(config)#interface vlan 2
SwitchC(config-If-Vlan2)#ip ospf authentication
SwitchC(config-If-Vlan2)#ip ospf authentication-key DCS
SwitchC(config-If-Vlan2)#exit
```

Configure IP address and area number for interface vlan3

```
SwitchC(config)# interface vlan 3
SwitchC(config-If-Vlan3)#ip address 20.1.3.1 255.255.255.0
SwitchC(config-If-Vlan3)#exit
SwitchC(config)#router ospf
SwitchC(config-router)#network 20.1.3.0/24 area 1
SwitchC(config-router)#exit
```

Configure IP address and area number for interface vlan 1

```
SwitchC(config)# interface vlan 1
SwitchC(config-If-Vlan1)#ip address 10.1.5.1 255.255.255.0
SwitchC(config-If-Vlan1)#exit
SwitchC(config)#router ospf
SwitchC(config-router)#network 10.1.5.0/24 area 0
SwitchC(config-router)#exit
```

Configure MD5 key authentication.

```
SwitchC(config)#interface vlan 1
SwitchC (config-If-Vlan1)#ip ospf authentication message-digest
SwitchC (config-If-Vlan1)#ip ospf authentication-key DCS
SwitchC (config-If-Vlan1)#exit
SwitchC(config)#exit
```

4)SwitchD:

Configure IP address for interface vlan2

```
SwitchD#config
SwitchD(config)# interface vlan 2
SwitchD(config-If-Vlan2)# ip address 10.1.1.4 255.255.255.0
SwitchD(config-If-Vlan2)#exit
```

Enable OSPF protocol, configure the area number for interface vlan2.

```
SwitchD(config)#router ospf
SwitchD(config-router)#network 10.1.1.0/24 area 1
SwitchD(config-router)#exit
```

Configure simple key authentication.

```
SwitchD(config)#interface vlan 2
SwitchD(config-If-Vlan2)#ip ospf authentication
SwitchD(config-If-Vlan2)#ip ospf authentication-key DCS
SwitchD(config-If-Vlan2)#exit
```

Configure the IP address and the area number for the interface vlan 1

```
SwitchD(config)# interface vlan 1
SwitchD(config-If-Vlan1)# ip address 10.1.6.1 255.255.255.0
SwitchD(config-If-Vlan1)exit
SwitchD(config)#router ospf
SwitchD(config-router)#network 10.1.6.0/24 area 0
SwitchD(config-router)#exit
```

Configure MD5 key authentication

```
SwitchD(config)#interface vlan 1
SwitchD(config-If-Vlan1)#ip ospf authentication message-digest
SwitchD(config-If-Vlan1)#ip ospf authentication-key DCS
SwitchD(config-If-Vlan1)exit
SwitchD(config)#exit
```

1.6.4.2 Configuration Examples of OSPF VPN

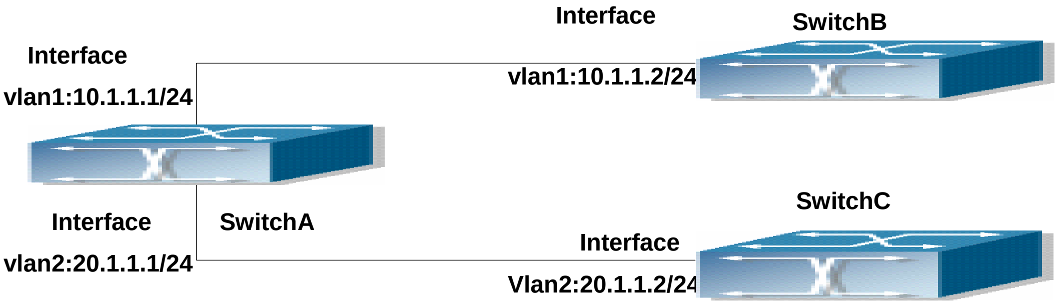

Fig 1-10 OSPF VPN Example

The above figure shows that a network consists of three Layer 3 switches in which the switchA as PE, SwitchB and SwitchC as CE1 and CE2. The PE is connected to CE1 and CE2 through vlan1 and vlan2. The routing messages are exchanged between PE and CE through OSPF protocol.

a) SwitchA, the Layer 3 switch as PE

Configure VPN route/transmitting examples vpnb and vpnc

```
SwitchA#config
SwitchA(config)#ip vrf vpnb
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit
SwitchA#(config)
SwitchA(config)#ip vrf vpnc
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit
```

Associate the vlan 1 and vlan 2 respectively with vpnb and vpnc while configuring IP address

```
SwitchA(config)#in vlan1
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
SwitchA(config-if-Vlan1)#exit
SwitchA(config)#in vlan2
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
SwitchA(config-if-Vlan2)#exit
```

Configure OSPF examples associated with vpnb and vpnc respectively

```
SwitchA(config)#
SwitchA(config)#router ospf 100 vpnb
SwitchA(config-router)#network 10.1.1.0/24 area 0
SwitchA(config-router)#redistribute bgp
SwitchA(config-router)#exit
SwitchA(config)#router ospf 200 vpnc
SwitchA(config-router)#network 20.1.1.0/24 area 0
SwitchA(config-router)#redistribute bgp
```

b) The Layer 3 switch SwitchB of CE1:

Configure the IP address of Ethernet E 1/2

```
SwitchB#config
SwitchB(config)# interface Vlan1
SwitchB(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (config-if-vlan1)exit
```

Enable OSPF protocol and configuring OSPF segments

```
SwitchB(config)#router ospf
SwitchB(config-router-rip)#network 10.1.1.0/24 area 0
SwitchB(config-router-rip)#exit
```

c) The Layer 3 switch SwitchC of CE2

Configure the IP address of Ethernet E 1/2

```
SwitchC#config
SwitchC(config)# interface Vlan1
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

SwitchC (config-if-vlan1)#exit

Initiate OSPF protocol and configuring OSPF segments

SwitchC(config)#router ospf

SwitchC(config-router)#network 20.1.1.0/24 area 0

SwitchC(config-router)#exit

1.6.5 OSPF Troubleshooting Help

The OSPF protocol may not be working properly due to errors such as physic connection, configuration error when configuring and using the OSPF protocol. So users should pay attention to following:

First ensure the physic connection is correct

Second, ensure the interface and link protocol are UP (use show interface command)

Configure different IP address from different segment on each interface

Then initiate OSPF protocol (use router-ospf command) and configure the OSPF area on corresponding interface

After that, a OSPF protocol feature should be checked---the OSPF backbone area should be continuous and apply virtual link to ensure it is continuous. if not; all non 0 areas should only be connected to other non 0 area through 0 area; a border Layer 3 switch means that one part of the interfaces of this switch belongs to 0 area, the other part belongs to non 0 area; Layer 3 switch DR should be specified for multi-access network such as broadcast network.

If the OSPF routing problem remains unresolved after checking and debugging, please use debug ospf packet/events commands and record the debug messages in three minutes ,then send it to our technical service center.

1.6.5.1 Monitor And Debug Command

1.6.5.1.1 debug ospf events

Command: [no]debug ospf events [abr|asbr|lsa|nssa|os|router|vlink]

Function: Open debugging switches showing various OSPF events messages; the "**[no]debug ospf events [abr|asbr|lsa|nssa|os|router|vlink**]" command closes the debugging switch.

Default: Closed

Command Mode: Admin and global mode

Example: Switch#debug ospf events router

1.6.5.1.2 debug ospf ifsm

Command: [no]debug ospf ifsm [status|events|timers]

Function: Open debugging switches showing the OSPF interface states; the "**[no]debug ospf**

ifsm [status|events|timers]” command closes this debugging switches.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf ifsm events

1.6.5.1.3 debug ospf lsa

Command: [no]debug ospf lsa [generate|flooding|install|maxage|refresh]

Function: Open debugging switches showing showing link state announcements; the “**[no]debug ospf lsa [generate|flooding|install|maxage|refresh]**” closes the debugging switches.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf lsa generate

1.6.5.1.4 debug ospf nfsm

Command: [no]debug ospf nfsm [status|events|timers]

Function: Open debugging switches showing showing OSPF neighbor state machine; the “**[no]debug ospf nfsm [status|events|timers]**”command closes this debugging switch.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf nfsm events

1.6.5.1.5 debug ospf nsm

Command: [no]debug ospf nsm [interface|redistribute]

Function: Open debugging switches showing showing OSPF NSM, the “**[no]debug ospf nsm [interface|redistribute]**” command closes this debugging switch.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf nsm interface

1.6.5.1.6 debug ospf packet

Command: [no]debug ospf packet [dd|detail|hello|ls-ack|ls-request|ls-update|recv|detail]

Function: Open debugging switches showing showing OSPF packet messages; the “**[no]debug ospf packet [dd|detail|hello|ls-ack|ls-request|ls-update|recv|detail]**” command closes this debugging switch.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf packet hello

1.6.5.1.7 debug ospf route

Command: [no]debug ospf route [ase|ia|install|spf]

Function: Open debugging switches showing showing OSPF related routes; the "**[no]debug ospf route [ase|ia|install|spf]**" command closes this debugging switch.

Default: Closed

Command Mode: Admin mode and global mode

Example: Switch#debug ospf route spf

1.6.5.1.8 debug ospf redistribute message send

Command: debug ospf redistribute message send

no debug ospf redistribute message send

Function: To enable/disable debugging of sending command from OSPF process redistributed to other ospf process routing.

Parameter: None.

Default: Disabled.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

Switch#debug ospf redistribute message send

1.6.5.1.9 debug ospf redistribute route receive

Command: debug ospf redistribute route receive

no debug ospf redistribute route receive

Function: To enable/disable debugging of received routing message from nsm for OSPF process.

Parameter: None.

Default: Disabled.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

Switch#debug ospf redistribute route receive

1.6.5.1.10 show ip ospf

Command: show ip ospf [<*process-id*>]

Function: Display OSPF main messages.

Parameter: ***<process-id>*** is the process ID, ranging between 0~65535

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ip ospf
 Routing Process "ospf 0" with ID 192.168.1.1
 Process bound to VRF default
 Process uptime is 2 days 0 hour 30 minutes
 Conforms to RFC2328, and RFC1583Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of non-default external LSA 0
 External LSA database is unlimited.
 Number of LSA originated 0
 Number of LSA received 0
 Number of areas attached to this router: 1
Area 0 (BACKBONE) (Inactive)
Number of interfaces in this area is 0(0)
Number of fully adjacent neighbors in this area is 0
Area has message digest authentication
SPF algorithm executed 0 times
Number of LSA 0. Checksum Sum 0x000000
Routing Process "ospf 10" with ID 0.0.0.0
 Process bound to VRF test
 Process uptime is 4 days 23 hours 51 minutes
 Conforms to RFC2328, and RFC1583Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x000000
```

```
  Number of opaque AS LSA 0. Checksum Sum 0x000000
  Number of non-default external LSA 0
  External LSA database is unlimited.
  Number of LSA originated 0
  Number of LSA received 0
  Number of areas attached to this router: 1
Area 0 (BACKBONE) (Inactive)
Number of interfaces in this area is 0(0)
Number of fully adjacent neighbors in this area is 0
Area has no authentication
SPF algorithm executed 0 times
Number of LSA 0. Checksum Sum 0x000000
```

1.6.5.1.11 show ip ospf border-routers

Command: show ip ospf [<*process-id*>] border-routers

Function: Display the intra-domain route entries for the switch to reach ABR and ASBR of all instances.

Parameter: ***<process-id>*** is the process ID, ranging between 0~65535

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ip ospf border-routers
OSPF process 0 internal Routing Table
Codes: i - Intra-area route, I - Inter-area route
i 10.15.0.1 [10] via 10.10.0.1, Vlan1, ASBR, Area 0.0.0.0
i 172.16.10.1 [10] via 10.10.11.50, Vlan2, ABR, ASBR, Area 0.0.0.0
```

1.6.5.1.12 show ip ospf database

Command: show ip ospf [<*process-id*>] database[{

adv-router [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| asbr-summary[{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}] |

externel [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| network [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| nssa-external [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}] |

opaque-area [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| opaque-as [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

|opaque-link [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| router [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

| summary [{<*linkstate_id*>| self-originate |adv-router <*advertiser_router*>}]

|self-originate | max-age }]

Function: Display the OSPF link state data base messages.

Parameter: ***<process-id>*** is the process ID, ranging between 0~65535

<*linkstate_id*> Link state ID, shown in point divided demical system

<*advertiser_router*> is the ID of Advertising router, shown in point divided demcial IP address format

Default: Not displayed

Command Mode: All modes

Usage Guide: According to the output messages of this command, we can view the OSPF link state database messages

Example:

```
Switch#show ip ospf database
                    Router Link States (Area 0.0.0.2)
Link ID           ADV Router        Age Seq#          CkSum Link count
192.168.1.2       192.168.1.2       254 0x80000031 0xec21 1
192.168.1.3       192.168.1.3       236 0x80000033 0x0521 2
                    Net Link States (Area 0.0.0.2)
Link ID           ADV Router        Age Seq#          CkSum
20.1.1.2          192.168.1.2       254 0x8000002b 0xece4
                    Summary Link States (Area 0.0.0.2)
Link ID           ADV Router        Age Seq#           CkSum    Route
6.1.0.0           192.168.1.2        68 0x8000002b    0x5757   6.1.0.0/22
6.1.1.0           192.168.1.2       879 0x8000002a    0xf8bc    6.1.1.0/24
22.1.1.0          192.168.1.2       308 0x8000000c    0xc8f0    22.1.1.0/24
                    ASBR-Summary Link States (Area 0.0.0.2)
Link ID           ADV Router        Age Seq#          CkSum
192.168.1.1       192.168.1.2      1702 0x8000002a 0x89c7
                    AS External Link States
Link ID           ADV Router        Age Seq#           CkSum    Route
2.2.2.0           192.168.1.1      1499 0x80000056    0x3a63   E2 2.2.2.0/24 [0x0]
2.2.3.0           192.168.1.1      1103 0x8000002b    0x0ec3   E2 2.2.3.0/24 [0x0]
```

1.6.5.1.13 show ip ospf interface

Command: show ip ospf interface <*interface*>

Function: Display the OSPF interface messages.

Parameter: ***<interface> is the*** name of interface

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ip ospf interface
Loopback is up, line protocol is up
OSPF not enabled on this interface
Vlan1 is up, line protocol is up
Internet Address 10.10.10.50/24, Area 0.0.0.0
Process ID 0, Router ID 10.10.11.50, Network Type BROADCAST, Cost: 10
Transmit Delay is 5 sec, State Waiting, Priority 1
No designated router on this network
No backup designated router on this network
Timer intervals configured, Hello 35, Dead 35, Wait 35, Retransmit 5
Hello due in 00:00:16
Neighbor Count is 0, Adjacent neighbor count is 0
```

1.6.5.1.14 show ip ospf neighbor

Command: show ip ospf [<*process-id*>] neighbor [{<*neighbor_id*> |all |detail [all] |interface<*ifaddress*>}]

Function: Display the OSPF adjacent point messages.

Parameter: <*process-id*> is the process ID ranging between 0~65535

<*neighbor_id*> is the dotted decimal notation neighbor ID

all: Display messages of all neighbors

detail: Display detailed messages of all neighbors

<*ifaddress*> Interface IP address

Default: Not displayed

Command Mode: All modes

Usage Guide: OSPF neighbor state can be checked by viewing the output of this command

Example:

```
Switch#show ip ospf neighbor
OSPF process 0:
Neighbor ID     Pri   State          Dead Time   Address       Interface
192.168.1.1       1    Full/Backup    00:00:32   6.1.1.1        Vlan1
192.168.1.3       1    Full/DR        00:00:36   20.1.1.3       Vlan2
192.168.1.3       1    Full/ -        00:00:30   20.1.1.3      VLINK2
```

Displayed information	Explanation
Neighbor ID	ID Neighbor ID
Priority	Priority
State	Neighbor relation state
Dead time	Neighbor dead time

Address	Interface Address
Interface	Interface name

1.6.5.1.15 show ip ospf route

Command: show ip ospf [<*process-id*>] route

Function: Display the OSPF routing table messages.

Parameter: ***<process-id>*** is the process ID ranging between 0~65535

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ip ospf route
O   10.1.1.0/24 [10] is directly connected, Vlan1, Area 0.0.0.0
O   10.1.1.4/32 [10] via 10.1.1.4, Vlan1, Area 0.0.0.0
IA 11.1.1.0/24 [20] via 10.1.1.1, Vlan1, Area 0.0.0.0
IA 11.1.1.2/32 [20] via 10.1.1.1, Vlan1, Area 0.0.0.0
IA 12.1.1.0/24 [20] via 10.1.1.2, Vlan1, Area 0.0.0.0
IA 12.1.1.2/32 [20] via 10.1.1.2, Vlan1, Area 0.0.0.0
O   13.1.1.0/24 [10] is directly connected, Vlan4, Area 0.0.0.3
O   14.1.1.0/24 [10] is directly connected, Vlan5, Area 0.0.0.4
IA 15.1.1.0/24 [20] via 13.1.1.2, Vlan4, Area 0.0.0.3
IA 15.1.1.2/32 [20] via 13.1.1.2, Vlan4, Area 0.0.0.3
E1 100.1.0.0/16 [21] via 10.1.1.1, Vlan1
E1 100.2.0.0/16 [21] via 10.1.1.1, Vlan1
```

1.6.5.1.16 show ip ospf redistribute

Command: show ip ospf [<*process-id*>] redistribute

Function: To display the routing message redistributed from external process of OSPF.

Parameter: ***<process-id>*** is the process ID ranging between 0~65535.

Default: None.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip ospf redistribute
        ospf process 1 redistribute information：
          ospf process 2
          ospf process 3
          bgp
        ospf process 2 redistribute information：
```

```
          ospf process 1
          bgp
        ospf process 3 redistribute information：
          ospf process 1
          bgp

Switch#show ip ospf 2 redistribute
        ospf process 2 redistribute information：
          ospf process 1
bgp
```

1.6.5.1.17 show ip ospf virtual-links

Command: show ip ospf [<*process-id*>] virtual-links

Function: Display the OSPF virtual link message.

Parameter: ***<process-id>*** is the process ID ranging between 0~65535

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ip ospf virtual-links
Virtual Link VLINK0 to router 10.10.0.9 is up
Transit area 0.0.0.1 via interface Vlan1
Transmit Delay is 1 sec, State Point-To-Point,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:02
Adjacency state Full
Virtual Link VLINK1 to router 10.10.0.123 is down
Transit area 0.0.0.1 via interface Vlan1
Transmit Delay is 1 sec, State Down,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in inactive
Adjacency state Down
```

1.6.5.1.18 show ip route process-detail

Command: show ip route [database] process-detail

Function: To display IP router with specific process id. (process id or tag)

Parameter: If database parameter is included, then display all routes, If no parameter is included, then only display operation route.

Default: Not redistributed any OSPF routing by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip route database process-detail
Codes: K - kernel, 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, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       > - selected route, * - FIB route, p - stale info

C        *> 127.0.0.0/8 is directly connected, Loopback
O            192.168.2.0/24 [110/10] is directly connected, Vlan2, 00:06:13,process 12
C        *> 192.168.2.0/24 is directly connected, Vlan2
```

1.6.5.1.19 show ip route vrf process-detail

Command: show ip route vrf <*name*> [database] process-detail

Function: To display IP router with specific process id. (process id or tag)

Parameter: If database parameter is included, then display all routes, If no parameter is included, then only display operation route.

Default: Not redistributed any OSPF routing by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip route vrf tt database process-detail
Codes: K - kernel, 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, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       > - selected route, * - FIB route, p - stale info

C        *> 127.0.0.0/8 is directly connected, Loopback
O            192.168.2.0/24 [110/10] is directly connected, Vlan2, 00:06:13,process 12
C        *> 192.168.2.0/24 is directly connected, Vlan2
```

1.6.5.1.20 show ipv6 route process-detail

Command: show ipv6 route [database] process-detail

Function: To display IP router with specific process id. (process id or tag)

Parameter: If database parameter is included, then display all routes, If no parameter is included, then only display operation route.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ipv6 route database process-detail
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
        I - IS-IS, B - BGP
        > - selected route, * - FIB route, p - stale info
Timers: Uptime

C*> ::1/128 via ::, Loopback, 00:29:53
O     2001::/64 [110/10] via ::, Vlan1, 00:01:07 ,process aaa
C*> 2001::/64 via ::, Vlan1, 00:02:54
O*> 2006::/64 [110/10] via ::, Vlan1, 00:01:07, process aaa
O*> 2008::/64 [110/20] via fe80::203:fff:fe01:2542, Vlan1, 00:00:54, process bbb
```

1.6.5.1.21 show ip protocols

Command: show ip protocols

Function: Display the running routing protocol messages.

Default: None

Command Mode: All modes

Example:

Switch#show ip protocols

Use “show ip protocols” command will show the messages of the routing protocol running on current layer 3 switch

For example, the displayed messages are:

```
Routing Protocol is "ospf 0"
Invalid after 0 seconds, hold down 0, flushed after 0
Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing:
  Routing for Networks:
    10.1.1.0/24
    12.1.1.0/24
  Routing Information Sources:
    Gateway              Distance         Last Update
  Distance: (default is 110)
    Address              Mask                  Distance List
Routing Protocol is "bgp 0"
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Neighbor(s):
  Address                FiltIn FiltOut DistIn DistOut Weight RouteMap
  Incoming Route Filter:
```

1.7 OSPFv3

1.7.1 Introduction to OSPFv3

OSPFv3（Open Shortest Path First） is the third version for Open Shortest Path First, and it is the IPv6 version of OSPF Protocol. It is an interior dynamic routing protocol for autonomous system based on link-state. The protocol creates a link-state database by exchanging link-states among layer3 switches, then uses the Shortest Path First algorithm to generate a route table basing on that database.

Autonomous system (AS) is a self-managed interconnected network. In large networks, such as the Internet, a giant interconnected network is broken down to autonomous systems. Big enterprise networks connecting to the Internet are independent AS, since the other host on the Internet are not managed by those AS and they don’t share interior routing information with the layer3 switches on the Internet.

Each link-state layer3 switch can provide information about the topology with its neighboring layer3 switches.

- The network segment (link) connecting to the layer3 switch
- State of the connecting link

Link-state information is flooded throughout the network so that all layer3 switches can get first hand information. Link-state layer3 switches will not broadcast all information contained in their route tables; instead, they only send changed link-state information. Link-state layer3 switches establish neighborhood by sending “HELLO” to their neighbors, then link-state advertisements (LSA) will be sent among neighboring layer3 switches. Neighboring layer3 switch copy the LSA to their routing table and transfer the information to the rest part of the network. This process is referred to as “flooding”. In this way, firsthand information is sent throughout the network to provide accurate map for creating and updating routes in the network. Link-state routing protocols use cost instead of hops to decide the route. Cost is assigned automatically or manually. According to the algorithm in link-state protocol, cost can be used to calculate the hop number for packets to pass, link bandwidth, and current load of the link, The administrator can even add weight for better assessment of the link-state.

1) When a link-state layer3 switch enters a link-state interconnected network, it sends a HELLO packet to get to know its neighbors and establish neighborhood.

2) The neighbors respond with information about the links they are connecting and the related costs.

3) The originate layer3 switch uses this information to build its own routing table.

4) Then, as part of the regular update, layer3 switch send link-state advertisement (LSA) packets to its neighboring layer3 switches. The LSA include links and related costs of that layer3 switch.

5) Each neighboring layer3 switch copies the LSA packet and passes it to the next neighbor (i.e. flooding).

6) Since routing database is not recalculated before layer3 switch forwards LSA flooding, the converging time is greatly reduced.

One major advantage of link-state routing protocols is the fact that infinite counting is impossible, this is because of the way link-state routing protocols build up their routing table. The second advantage is that converging in a link-state interconnected network is very fast, once the routing topology changes, updates will be flooded throughout the network very soon. Those advantages release some layer3 switch resources, as the process ability and bandwidth used by bad route information are minor.

The features of OSPFv3 protocol include the following: OSPFv3 supports networks of various scales, several hundreds of layer3 switches can be supported in an OSPFv3 network. Routing topology changes can be quickly found and updating LSAs can be sent immediately, so that routes converge quickly. Link-state information is used in shortest path algorithm for route calculation, eliminating loop route. OSPFv3 divides the autonomous system intro areas, reducing database size, bandwidth occupation and calculation load. (According to the position of layer3 switches in the autonomous system, they can be grouped as internal area switches, area edge switches, AS edge switches and backbone switches). OSPFv3 supports load balance and multiple routes to the same destination of equal costs. OSPFv3 supports 4 level routing mechanisms (process routing according to the order of route inside an area, route between areas, first category external route and second category external route). OSPFv3 support IP subnet and redistribution of routes from the other routing protocols, and interface-based packet verification. OSPFv3 supports sending packets in multicast.

Each OSPFV3 layer3 switch maintains a database describing the topology of the whole autonomous system. Each layer3 switch gathers the local status information, such as available interface, reachable neighbors, and sends link-state advertisement (sending out link-state information) to exchange link-state information with other OSPFv3 layer3 switches to form a link-state database describing the whole autonomous system. Each layer3 switch builds a shortest path tree rooted by itself according to the link-state database, this tree provide the routes to all nodes in an autonomous system. If two or more layer3 switches exist (i.e. multi-access network), "designated layer3 switch” and “backup designated layer3 switch” will be selected. Designated layer3 switch is responsible for spreading link-state of the network. This concept helps reducing the traffic among the Layer3 switches in multi-access network.

OSPFv3 protocol requires the autonomous system to be divided into areas. That is to divide the autonomous system into 0 area (backbone area) and non-0 areas. Routing information between areas are further abstracted and summarized to reduce the bandwidth required in the network. OSPFv3 uses four different kinds of routes: they are the route inside the area, route between areas, first category external route and second category external route, in the order of highest priority to lowest. The route inside an area and between areas describe the internal network structure of an autonomous system, while external routes describe external routes describe how to select the routing information to destination outside the autonomous system. The first type of external route corresponds to the information introduced by OSPFv3 from the other interior routing protocols, the costs of those routes are comparable with the costs of OSPFv3 routes; the second type of external route corresponds to the information introduced by OSPFv3 from the other external routing protocols, but the costs of those routes are far greater than that of OSPFv3 routes, so OSPFv3 route cost is ignored when calculating route costs.

OSPFv3 areas are centered with the Backbone area, identified as the Area 0, all the other areas must be connected to Area 0 logically, and Area 0 must be continuous. For this reason, the concept of virtual link is introduced to the backbone area, so that physically separated areas still have logical connectivity to the backbone area. The configurations of all the layer3 switches in the same area must be the same.

In one word, LSA can only be transferred between neighboring Layer3 switches, and OSPFv3 protocol includes seven kinds of LSA: link LSA,intra-area prefix LSA, router LSA, network LSA, inter-area prefix LSA, inter-area router LSA and Autonomous system external LSA. Router LSA is generated by each Layer 3 switch in an OSPF area, and is sent to all other neighboring Layer 3 switch in this area; network LSA is generated by designated Layer 3 switch in the OSPF area of multi-access network and is sent to all other neighboring layer3 switches in this area.(To reduce data traffic among each Layer 3 switches in the multi-access network, “designated layer3 switch” and “backup designated layer3 switch” should be selected in the multi-access network, and the network link-state is broadcasted by designated Layer 3 switch); the inter-area prefix LSA and inter-area router LSA are generated by OSPF area border Layer 3 switches and transferred among those switches. The autonomic system external LSA is generated by autonomic system external border Layer 3 switches and transferred in the whole autonomic system. Link LSA is generated by Layer 3 switch on the link and sent to other Layer 3 switches on the link. Internal-area prefix LSA is generated by designated layer3 switch of each link in this area, and flooded to the whole area.

For autonomous system focused on external link-state announcement, OSPFv3 allow some areas to be configured as STUB areas in order to reduce the size of topological database. Router LSA, network LSA, inter-area prefix LSA, link LSA, internal-area prefix LSA are permitted to advertise to STUB area. Default route must be used in STUB area, Layer 3 switches on the area border of STUB area announces to default routes of STUB area by inter-area prefix LSA; these default routes only flood in STUB area, not outside of STUB area. Each STUB area has a corresponding default route, the route from STUB area to AS external destination depends only on default route of this area.

The following simply outlines the route calculation process of OSPFv3 protocol:

1）Each OSPF-enabled layer3 switch maintains a database (LS database) describing the link-state of the topology structure of the whole autonomous system. Each layer3 switch generates a link-state advertisement according to its surrounding network topology structure (router LSA), and sends the LSA to other layer3 switches through link-state update (LSU) packets. Thus, each layer3 switches receives LSAs from other layer3 switches, and all LSAs combined to the link-state database.

2）Since a LSA is the description of the network topology structure around a layer3 switch, the LS database is the description of the network topology structure of the whole network. The layer3 switches can easily create a weighted vector map according to the LS database. Obviously, all layer3 switches in the same autonomous system will have the same network topology map.

3）Each layer3 switch uses the shortest path first (SPF) algorithm to calculate a tree of shortest path rooted by itself. The tree provides the route to all the nodes in the autonomous system, leaf nodes consist of the external route information. The external route can be marked by the layer3 switch broadcast it, so that additional information about the autonomous system can be recorded. As a result, the route table of each layer3 switch is different.

OSPFv3 protocol is developed by the IETF, the OSPF v3 used now is fulfilled according to the content described in RFC2328 and RFC2740.

As a result of continuous development of IPv6 network, it has the network environment of nonsupport IPv6 sometimes, so it needs to do the IPv6 operation by tunnel. Therefore, our OSPFv3 supports configuration on configure tunnel, and passes through nonsupport IPv6 network by unicast packet of IPv4 encapsulation.

1.7.2 OSPFv3 Configuration Task List

1. Enable OSPFv3 (required)
   （1） Enable/disable OSPFv3(required)
   （2） Configure the router-id number of the layer3 switch running OSPFv3 (required)
   （3） Configure the network scope for running OSPFv3 (optional)
   （4） enable OSPFv3 on the interface (required)
2. Configure OSPFv3 auxiliary parameters (optional)
   （1） Configure OSPFv3 packet sending mechanism parameters
      1) Set the OSPFv3 interface to receive only
      2) Configure the cost for sending packets from the interface
      3) Configure OSPFv3 packet sending timer parameter (timer of broadcast interface sending HELLO packet to poll, timer of neighboring layer3 switch invalid timeout, timer of LSA transmission delay and timer of LSA retransmission.
   （2） Configure OSPFv3 route introduction parameters
      1) Configure default parameters (default type, default tag value, default cost)
      2) Configure the routes of the other protocols to introduce to OSPFv3
   （3） Configure OSPFv3 importing the routes of other OSPFv3 processes
      1) Enable the function of OSPFv3 importing the routes of other OSPFv3 processes
      2) Display relative information
      3) Debug
   （4） Configure other OSPFv3 protocol parameters
      1) Configure OSPFv3 routing protocol priority
      2) Configure cost for OSPFv3 STUB area and default route
      3) Configure OSPFv3 virtual link
      4) Configure the priority of the interface when electing designated layer3 switch
3. Close OSPFv3 Protocol

1. Enable OSPFv3 Protocol

It is very simple to run the basic configurations of OSPFv3 routing protocol on the Layer 3 switch of ES4700 series , normally only enabling OSPFv3, implement OSPFv3 interface, the default value is defined to OSPFv3 protocol parameters. Refer to 2. Configure OSPF auxiliary parameters, if the OSPFv3 protocol parameters need to be modified.

Commands	Explanation
Global mode	
[no] router IPv6 ospf <*tag*>	The command initializes ospfv3 routing process and enter ospfv3 mode to configure ospfv3 routing process. The **[no] router IPv6 ospf** <*tag*> command stops relative process. (required)

OSPFv3 Protocol Configure Mode	
router-id <*router_id*> **no router-id**	Configure router for ospfv3 process. The **no router-id** command returns ID to 0.0.0.0 (required)
[**no**] **passive-interface<*ifname*>**	Configure an interface receiving without sending. The [**no**] **passive-interface**<*ifname*>command cancels configuration.
Interface Configuration Mode	
[no] IPv6 router ospf {**area <*area-id*> [instance-id <*instance-id*> | tag <*tag*> [instance-id <*instance-id*>]] | tag <*tag*> area <*area-id*> [instance-id <*instance-id*>]}**	Implement ospfv3 routing on the interface. The **[no] IPv6 router ospf** {**area <*area-id*>** [**instance-id <*instance-id*> | tag <*tag*> [instance-id <*instance-id*>]] | tag <*tag*> area <*area-id*>** [**instance-id <*instance-id*>**]} command cancels configuration.

2. Configure OSPFv3 parameters

（**1**）**Configure OSPFv3 packet sending mechanism parameters**

1）Set the OSPF interface to receive only

2）Configure the cost for sending packets from the interface

Commands	Explanation
Interface Configuration Mode	
IPv6 ospf cost <*cost*> [instance-id <*id*>] **no IPv6 ospf cost [instance-id <*id*>]**	Appoint interface to implement required cost of OSPFv3 protocol. The **no IPv6 ospf cost [instance-id <*id*>]** restores the default setting

3）Configure OSPFv3 packet sending timer parameter (timer of broadcast interface sending HELLO packet to poll, timer of neighboring layer3 switch invalid timeout, timer of LSA transmission delay and timer of LSA retransmission.

Commands	Explanation
Interface Configuration Mode	
IPv6 ospf hello-interval <*time*> [instance-id <*id*>] **no IPv6 ospf hello-interval [instance-id <*id*>]**	Sets interval for sending HELLO packets; the “**no IPv6 ospf hello-interval [instance-id <*id*>]**” command restores the default setting.

IPv6 ospf dead-interval <*time*> [instance-id <*id*>] **no IPv6 ospf dead-interval [instance-id <*id*>]**	Sets the interval before regarding a neighbor layer3 switch invalid; the "**no IPv6 ospf dead-interval [instance-id <*id*>]**" command restores the default setting.
IPv6 ospf transit-delay <*time*> [instance-id <*id*>] **no IPv6 ospf transit-delay [instance-id <*id*>]**	Sets the delay time before sending link-state broadcast; the "**no IPv6 ospf transit-delay [instance-id <*id*>]**" command restores the default setting.
IPv6 ospf retransmit <*time*> [instance-id <*id*>] **no IPv6 ospf retransmit [instance-id <*id*>]**	.Sets the interval for retransmission of link-state advertisement among neighbor layer3 switches; the "**no IPv6 ospf retransmit [instance-id <*id*>]**" command restores the default setting.

（2）Configure OSPFv3 route introduction parameters

Configure OSPFv3 route introduction parameters

Commands	Explanation
Router IPv6 OSPF mode	
[no]redistribute {kernel |connected| static| rip| isis| bgp} [metric<*value*>] [metric-type {1|2}][route-map<*word*>]	Introduces other protocol discovery routing and static routing regarded as external routing message. The **[no]redistribute {kernel |connected| static| rip| isis| bgp} [metric<*value*>] [metric-type {1|2}][route-map<*word*>]** command cancels imported external routing message.

（3）Configure OSPFv3 importing the routes of other OSPFv3 processes

1) Enable the function of OSPFv3 importing the routes of other OSPFv3 processes

Command	Explanation
Router IPv6 OSPF mode	

redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>] **no redistribute ospf [<*process-id*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>]**	Enable or disable the function of OSPFv3 importing the routes of other OSPFv3 processes.

2) Display relative information

Command	Explanation
Any mode	
show ipv6 ospf [<*process-id*>] redi stribute	Display the configuration information of the OSPFv3 process importing other outside routes.

3) Debug

Command	Explanation
Admin mode	
debug ipv6 ospf redistribute message send **no debug ipv6 ospf redistribute message send** **debug ipv6 ospf redistribute route receive** **no debug ipv6 ospf redistribute route receive**	Enable/disable debugging of sending command from OSPFv3 process redistributed to other OSPFv3 process routing. Enable/disable debugging of received routing message from nsm for OSPFv3 process.

（4）Configure Other Parameters of OSPFv3 Protocol

1） Configure OSPFv3 STUB Area & Default Routing Cost

2） Configure OSPFv3 Virtual Link

Commands	Explanation
OSPFv3 Protocol Configuration Mode	
timers spf <*spf-delay*> <*spf-holdtime*> **no timers spf**	Configure OSPFv3 SPF timer. The **no timers spf** command recovers default value.

area <*id*> stub [no-summary] **no area <*id*> stub [no-summary]** **area <*id*> default-cost <*cost*>** **no area <*id*> default-cost** **area <*id*> virtual-link A.B.C.D [instance-id <*instance-id*> INTERVAL]** **no area <*id*> virtual-link A.B.C.D [|INTERVAL]**	Configure parameters in OSPFv3 area (STUB area, Virtual link). The no command restores default value.

4）Configure the priority of the interface when electing designated layer3 switch (DR).

Commands	Explanation
Interface Configuration Mode	
IPv6 ospf priority <*priority*> [instance-id <*id*>] **no IPv6 ospf priority [instance-id <*id*>]**	Sets the priority of the interface in "designated layer3 switch" election; the "**no IPv6 ospf priority [instance-id <*id*>]**" command restores the default setting.

3. Disable OSPFv3 Protocol

Commands	Explanation
Global mode	
no router IPv6 ospf [<*tag*>]	Disable OSPFv3 Routing Protocol

1.7.3 Command For OSPFV3

1.7.3.1 area default cost

Command: area <*id*> default-cost <*cost*>

no area <*id*> default-cost

Function: Configure the cost of sending to the default summary route in stub or NSSA area; the "**no area <*id*> default-cost**" command restores the default value.

Parameter: <*id*> is the area number which could be shown as digits 0～4294967295, or as an IP address; **<*cost*>** ranges between <0-16777215>

Default: Default OSPFv3 cost is 1

Command Mode: OSPFv3 protocol mode

Usage Guide: The command is only adaptive to the ABR router connected to the stub area or NSSA area

Example: Set the default-cost of area 1 to 10

Switch(config-router)#area 1 default-cost 10

1.7.3.2 area range

Command: area <*id*> range <ipv6*address*> [advertise| not-advertise|]

no area <*id*> range <ipv6*address*>

Function: Aggregate OSPF route on the area border. The "**no area <*id*> range <*address*>**" cancels this function

Parameter: <*id*> is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

<ipv6*address*>=<*X:X::X:X/M*>,Specifies the area ipv6 network prefix and its length

advertise: Advertise this area

not-advertise : Not advertise this area

If both are not set, this area is defaulted for advertising

Default: Function not configured

Command Mode: OSPFv3 protocol mode

Usage Guide: Use this command to aggregate routes inside an area. If the network IDs in this area are not configured continuously, a summary route can be advertised by configuring this command on ABR. This route consists of all single networks belong to specific range.

Example:

Switch # config terminal

Switch (config)# router ipv6 ospf

Switch (config-router)# area 1 range 2000::/3

1.7.3.3 area stub

Command: area <*id*> stub [no-summary]

no area <*id*> stub [no-summary]

Function: Define a area to a stub area. The "**no area <*id*> stub [no-summary]**" command cancels this function

Parameter: <*id*> is the area number which could be digits ranging between 0～4294967295, and also as an IPv4 address.

no-summary: The area border routes stop sending link summary announcement to the stub area

Default: Not defined

Command Mode: OSPFv3 protocol mode

Usage Guide: Configure area stub on all routes in the stub area. There are two configuration commands for the routers in the stub area: stub and default-cost. All routers connected to the stub area should be configured with area stub command. As for area border routers connected to the stub area, their introducing cost is defined with area default-cost command.

Example:

```
Switch # config terminal
Switch (config)# router ipv6 ospf
Switch (config-router)# area 1 stub
```

1.7.3.4 area virtual-link

Command: area <*id*> virtual-link A.B.C.D [instance-id <*instance-id*> | INTERVAL <*value*>]

no area <*id*> virtual-link A.B.C.D [instance-id <*instance-id*> | |INTERVAL]

Function: Configure a logical link between two backbone areas physically divided by non-backbone area. The "**no area <*id*> virtual-link A.B.C.D [instance-id <*instance-id*> | |INTERVAL]**" command removes this virtual-link.

Parameter: <*id*>is the area number which could be digits ranging between 0～4294967295, and also as an IP address.

<***instance-id***> is the interface instance ID ranging between 0~255 and defaulted at 0

***INTERVAL*= [*dead-interval*|*hello-interval*|*retransmit-interval*|*transmit-delay*]**

<*value*>: The delay or interval seconds, ranging between 1~65535

<*dead-interval*>: A neighbor is considered offline for certain dead interval without its group messages which the default is 40 seconds.

<*hello-interval*>: The time interval before the router sends a hello group message, default is 10 seconds

<*retransmit-interval*>: The time interval before a router retransmitting a group message, default is 5 seconds

<*transmit-delay*>: The time delay before a router sending a group messages, 1 second by default

Default: No default configuration.

Command Mode: OSPFv3 protocol mode

Usage Guide: In the OSPF all non-backbone areas will be connected to a backbone area. If the connection to the backbone area is lost, virtual link will repair this connection. You can configure virtual link between any two backbone areas routers connected with the public non-backbone area. The protocol treat routers connected by virtual links as a point-to-point network

Example:

```
Switch#config terminal
Switch(config) #router ipv6 ospf
Switch(config-router) #area 1 virtual-link 10.10.11.50 hello 5 dead 20
Switch(config-router) #area 1 virtual-link 10.10.11.50 instance-id 1
```

1.7.3.5 abr-type

Command: abr-type {cisco|ibm| standard}

no abr-type [cisco|ibm| standard]

Function: Configure an OSPF ABR type with this command. The “**no abr-type [cisco|ibm| standard]**” command restores the default

Parameter: cisco, realize by cisco ABR; **ibm**, realize by ibm ABR; **shortcut**, specify a shortcut-ABR; **standard**, realize with standard（RFC2328）ABR.

Default: Cisco configured by default

Command Mode: OSPFv3 protocol mode

Usage Guide: For Specifying the realizing type of abr. This command is good for interactive operation among different OSPF realizing method and is especially useful in the multiple host environment.

Example: Configure abr as standard

```
Switch#config terminal
Switch(config)#router ipv6 ospf
Switch(config-router)#abr-type standard
```

1.7.3.6 default-metric

Command: default-metric <*value*>

no default-metric

Function: The command set the default metric value of OSPF routing protocol; the “**no default-metric**” returns to the default state.

Parameter: <*value*>, metric value, ranging between 1~16777214.

Default: Built-in, metric value auto translating.

Command Mode: OSPF protocol mode.

Usage Guide: When the default metric value makes the metric value not compatible, the route introducing still goes through. If the metric value can not be translated, the default value provides alternative option to carry the route introducing on. This command will result in that all introduced route will use the same metric value. This command should be used associating redistribute.

Example:

```
Switch#config terminal
Switch(config)#router ipv6 ospf
Switch(config-router)#default-metric 100
```

1.7.3.7 ipv6 ospf cost

Command: ipv6 ospf cost <*cost*> [instance-id <*id*>]

no ipv6 ospf　cost [instance-id <*id*>]

Function: Specify the cost required in running OSPF protocol on the interface; the “**no ipv6 ospf cost [instance-id <*id*>]**” command restores the default value

Parameter: ***<id>*** is the interface instance ID, ranging between 0~255, defaulted at 0

<*cost* > is the cost of OSPF protocol ranging between 1~65535

Default: Default OSPF cost on the interface is 10

Command Mode: Interface Mode

Usage Guide: The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 ospf cost 3
```

1.7.3.8 ipv6 ospf dead-interval

Command: ipv6 ospf dead-interval <*time* > [instance-id <*id*>]

no ipv6 ospf dead-interval [instance-id <*id*>]

Function: Specify the dead interval for neighboring layer 3 switch; the "**no ipv6 ospf dead-interval [instance-id <*id*>]**" command restores the default value.

Parameter: ***<id>*** is the interface instance ID, ranging between 0~255, defaulted at 0

<*time* > is the length of the adjacent layer 3 switch, in seconds, ranging between 1～65535

Default: The default dead interval is 40 seconds (normally 4 times of the hello-interval).

Command Mode: Interface Mode

Usage Guide: If no HELLO data packet received after the **dead-interval** period then this layer 3 switch is considered inaccessible and invalid. This command modifies the dead interval value of neighboring layer 3 switch according to the actual link state. The set **dead-interval** value is written into the Hello packet and transmitted. To ensure the normal operation of the OSPF protocol, the dead-interval between adjacent layer 3 switches should be in accordance or at least 4 times of the **hello-interval** value. The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 ospf dead-interval 80
```

1.7.3.9 ipv6 ospf display route single-line

Command: [no] ipv6 ospf display route single-line

Function: show ipv6 ospf route change the display results of show ipv6 ospf route command.The "**[no] ipv6 ospf display route single-line**" restores to default display mode

Default: Not configured

Command Mode: Global Mode

Usage Guide: The show ipv6 ospf route command displays the same route in several lines. This command will strict that one route will be displayed in one line

Example:

Switch#config terminal

Switch(config)#ipv6 ospf display route single-line

1.7.3.10 ipv6 ospf hello-interval

Command: ipv6 ospf hello-interval <*time*> [instance-id <*id*>]

no ipv6 ospf hello-interval [instance-id <*id*>]

Function: Specify the hello-interval on the interface; the "**no ipv6 ospf hello-interval [instance-id <*id*>]**" restores the default value

Parameter: <*id*> is the interface instance ID, ranging between 0~255, defaulted at 0

<*time* > is the length of the adjacent layer 3 switch, in seconds, ranging between 1～65535

Default: Default HELLO packet sending interval is 10 seconds.

Command Mode: Interface Mode

Usage Guide: HELLO data packet is the most common packet which is periodically sent to adjacent layer 3 switch to discover and maintain adjacent relationship, elect DR and BDR. The user set **hello-interval** value will be written into the HELLO packet and transmitted. The less the **hello-interval** value is, the sooner the network topological structure is discovered as well larger the cost. The ensure the normal operation of OSPF protocol the **hello-interval** parameter between the layer 3 switches adjacent to the interface must be in accordance. The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example:

Switch#config terminal

Switch(config)#interface vlan 1

Switch(Config-if-Vlan1)#ipv6 ospf hello-interval 20

1.7.3.11 ipv6 ospf priority

Command: ipv6 ospf priority <*priority*> [instance-id <*id*>]

no ipv6 ospf priority[instance-id <*id*>]

Function: Configure the priority when electing "Defined layer 3 switch" at the interface. The "**no ipv6 ospf [<*ip-address*>] priority**" command restores the default value

Parameter: <id> is the interface instance ID, ranging between 0~255, and defaulted at 0**<*priority*>** is the priority of which the valid value ranges between 0～255.

Default: The default priority when electing DR is 1.

Command Mode: Interface Mode

Usage Guide: When two layer 3 switches connected to the same segments both want to be the

"Defined layer 3 switch", the priority will decide which one should be chosen. Normally the one with higher priority will be elected, or the one with larger router-id number if the priorities are the same. A layer 3 switch with a priority equal to 0 will not be elected as "Defined layer 3 switch" or "Backup Defined layer 3 switch". The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example: Configure the priority of DR electing. Configure the interface vlan 1 to no election right, namely set the priority to 0.

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 ospf priority 0
```

1.7.3.12 ipv6 ospf retransmit-interval

Command: ipv6 ospf retransmit-interval <*time*> [instance-id <*id*>]

no ipv6 ospf retransmit-interval [instance-id <*id*>]

Function: Specify the retransmit interval of link state announcements between the interface and adjacent layer 3 switches. The "**no ipv6 ospf retransmit-interval [instance-id <*id*>]**" command restores the default value

Parameter: <*id*> is the interface instance ID, ranging between 0~255, defaulted at 0

<*time*> is the retransmit interveral of link state announcements between the interface and adjacent layer 3 switches, shown in seconds ang raning between 1～65535

Default: Default retransmit interval is 5 seconds

Command Mode: Interface Mode

Usage Guide: When a layer 3 switch transmits LSA to its neighbor, it will maintain the link state announcements till confirm from the object side is received. If the confirm packet is not received within the interval, the LSA will be retransmitted. The retransmit interval must be larger than the time it takes to make a round between two layer 3 switches. The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example: Configure the LSA retransmit interval of interface vlan 1 to 10 seconds

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 ospf retransmit-interval 10
```

1.7.3.13 ipv6 ospf transmit-delay

Command: ipv6 ospf transmit-delay <*time*> [instance-id <*id*>]

no ipv6 ospf transmit-delay [instance-id <*id*>]

Function: Configure the LSA sending delay time on the interface. The "**no ipv6 ospf transmit-delay [instance-id <*id*>]**" command restores to the default

Parameter: <*id*> is the instance ID ranging between 0~255 and defaulted at 0

<time> is the delay time of sending LSA on the interface, which is shown in seconds and ranged between 1～65535.

Default: The default delay time of send LSA on the interface is 1 second by default.

Command Mode: Interface Mode

Usage Guide: The LSA ages by time in the layer 3 switches but not in the transmission process. So by increasing the **transmit-delay** before sending LSA so that it will be sent out. The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example: Set the interface vlan 1 LSA sending delay to 3 seconds

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 ospf transmit-delay 3
```

1.7.3.14 ipv6 router ospf

Command: [no] ipv6 router ospf {area <*area-id*> [instance-id <*instance-id*>]| tag <*tag*> area <*area-id*> [instance-id <*instance-id*>]}

Function: Enable ospf route on the interface; the "**no ipv6 router ospf {area <*area-id*> [instance-id <*instance-id*>]| tag <*tag*> area <*area-id*> [instance-id <*instance-id*>]}**" command cancels this configuration.

Parameter: <*area-id*> is an area ID which could be shown in digits ranging between 0～4294967295, or an IPv4 address.

<*instance-id*> is the interface instance ID ranging between 0~255 and defaulted at 0.

<*tag*> ospfv3 process identifier

Default: Not configured.

Command Mode: Interface Mode.

Usage Guide: To enable this command on the interface, the area id must be configured. The instance ID and instance tag are optional. The ospfv3 process allows one routing instance for each instance ID. The route can be enabled on a interface with a instance ID. If the instance IDs are different, several OSPF process can be run on one interface. However different OSPF process should not use the same instance ID The command can configure on IPv6 tunnel interface, but it is successful configuration to only configure tunnel carefully.

Example:

```
Switch#config terminal
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 router ospf area 1 tag IPI instance-id 1
```

1.7.3.15 max-concurrent-dd

Command: max-concurrent-dd <*value*>

no max-concurrent-dd

Function: Configure with this command the current dd max concurrent number in the OSPF processing. The “**no max-concurrent-dd**” command restores the default

Parameter: ***<value>*** ranges between ***<1-65535>,*** the capacity of concurrent dd data packet processing.

Default: No default configuration. No dd concurrent limit

Command Mode: OSPFv3 protocol mode

Usage Guide: Specify the current dd max concurrent number in the OSPF processing

Example: Set the max concurrent dd to 20

```
Switch#config terminal
Switch(config)#router ipv6 ospf
Switch(config-router)#max-concurrent-dd 20
```

1.7.3.16 passive-interface

Command: [**no**] **passive-interface{<*ifname*>|vlan <*vlan-id*>}**

Function: Configure that the hello group not sent on specific interfaces. The “**no passive-interface{<*ifname*>|vlan <*vlan-id*>}** “ command cancels this function

Parameter: <***ifname***> is the specific name of interface

Default: Not configured

Command Mode: OSPFv3 protocol mode

Example: Switch#config terminal

```
Switch(config)#router ipv6 ospf
Switch(config-router)#passive-interface vlan1
```

1.7.3.17 redistribute

Command: [no]redistribute {kernel |connected| static| rip| isis| bgp} [metric<*value*>] [metric-type {1|2}][route-map<*word*>]

Function: Introduce route learnt from other routing protocols into OSPFv3.

Parameter: kernel Introduct from kernel route

connected Introduce from direct route

static Introduce from static route

rip Introduce from the RIP route

isis Introduce from ISIS route

bgp Introduce from BGP route

metric <*value*> is the introduced metric value, ranging between 0-16777214

metric-type {1|2} is the metric value type of the introduced external route, which can be 1 or 2, and it is 2 by default

route-map <*word*> targets to the probe of the route map for introducing route

Command Mode: OSPFv3 protocol mode.

Usage Guide: Learn and introduce other routing protocol into OSPFv3 area to generate AS-external_LSAs

Example: Switch#config terminal

Switch(config)#router ipv6 ospf

Switch(config-router)#redistribute bgp metric 12 metric-type 1

1.7.3.18 redistribute ospf

Command: redistribute ospf [<*process-tag*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>]

no redistribute ospf [<*process-tag*>] [metric<*value*>] [metric-type {1|2}] [route-map<*word*>]

Function: To redistribute routing information form process-tag to this command. The no command is to deletes the redistribution of process-tag routing to this process. When input the optional parameters of metric, metric type and routermap, then restores default configuration.

Parameter: process-tag is the process id of IPv6 OSPF process, NULL by default.

metric <*value*> is the metric for redistributed routing, range between 0 to 16777214.

metric-type {1|2} is the metric type for redistributed routing, only can be 1 or 2, and 2 by default.

route-map <*word*> is the pointer to the introduced routing map.

Default: Not redistributed any OSPFv3 routing by default.

Command Mode: OSPFv3 protocol mode.

Usage Guide: When process-id is not input, that means ospfv3 routing will be redistributed by default. (Process-tag is NULL) When the no command input the optional parameters of metric, metric-type and routermap, then restores default configuration. When not input any optional parameters that means to delete the router of redistributed process.

Example:

Switch(config)#router ipv6 ospf

Switch(config-router)#redistribute ospf

1.7.3.19 router-id

Command: router-id<*router-id*>

no router-id

Function: Configure router ID for ospfv3 process. The "**no router-id**" restores ID to 0.0.0.0

Parameter: <*router-id*>> is the router ID shown in IPv4 format

Default: 0.0.0.0 by default

Usage Guide: If the router-id is 0.0.0.0, the ospfv3 process can not be normally enabled. It is required to configure a router-id for ospfv3

Command Mode: OSPFv3 protocol mode

Example: Switch#config terminal

Switch(config)#router ipv6 ospf

Switch(config-router)#router-id 192.168.2.1

1.7.3.20 router ipv6 ospf

Command: [no] router ipv6 ospf [<*tag*>]

Function: This command initializes the ospfv3 routing prcess and enters ospfv3 mode for configuring the ospfv3 routing process. The "**[no] router ipv6 ospf [<*tag*>]**" command stops relevant process

Parameter: <*tag*> ospfv3 is the process mark which could be random strings made up of characters and digits

Command Mode: Global mode

Usage Guide: To let he ospfv3 routing process work properly, this command must be configured and ospfv3 must at least be enabled on one interface. When the tag configured by the ipv6 router ospf area command under interface mode matches with the tag of ospf process, the ospfv3 process is enabled on this interface.

Example: Switch#config terminal

Switch(config)#router ipv6 ospf IPI

1.7.3.21 timers spf

Command: timers spf <*spf-delay*> <*spf-holdtime*>

no timers spf

Function: Adjust route calculation timer value. The "**no timers spf**" restores the relevant value to default

Parameter: <*spf-delay*> 5 seconds by default

<*spf-holdtime*> 10 seconds by default

Command Mode: OSPFv3 protocol mode

Usage Guide: In this command the delay time between receiving topology change and SPF calculation, and further configured the hold time between two discontinuous SPF calculations.

Example: Switch#config terminal

Switch(config)#router ipv6 ospf

Switch(config-router)#timers spf 5 10

1.7.4 OSPFv3 Examples

Examples 1:OSPF autonomous system.

This scenario takes an OSPF autonomous system consists of five ES4700 series layer3 switches for example, where layer3 switch SwitchA and SwitchE make up OSPF area 0, layer3 switch SwitchB and SwitchC form OSPF area 1 (assume vlan1 interface of layer3 switch SwitchA belongs to area 0), layer3 switch SwitchD forms OSPF area2 (assume vlan2 interface of layer3 SwitchE belongs to area 0). Swtich1 and SwitchE are backbone layer3 switches, Swtich2 and SwitchD are area edge layer3 switches, and SwitchC is the in-area layer3 switch.

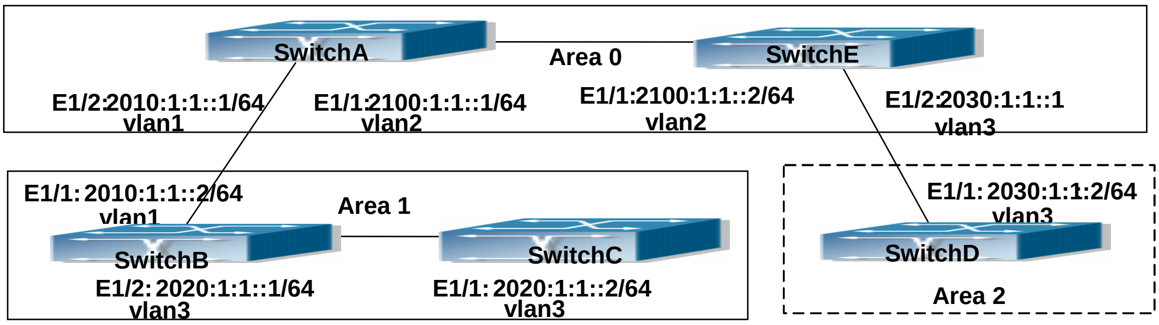

Fig 1-11 Network topology of OSPF autonomous system.

The configuration for layer3 switch SwitchA and SwitchE is shown below:

Layer3 switch SwitchA:

! Enable OSPFv3 protocol, configure router ID

```
SwitchA(config)#router IPv6 ospf
SwitchA (config-router)#router-id 192.168.2.1
```

Configure interface vlan1 IPv6 address and affiliated OSPFv3 area

```
SwitchA#config
SwitchA(config)# interface vlan 1
SwitchA(config-if-vlan1)# IPv6 address 2010:1:1::1/64
SwitchA(config-if-vlan1)# IPv6 router ospf area 0
SwitchA(config-if-vlan1)#exit
```

Configure interface vlan2 IP address and affiliated OSPFv3 area

```
SwitchA(config)# interface vlan 2
SwitchA(config-if-vlan2)# IPv6 address 2100:1:1::1/64
SwitchA(config-if-vlan2)# IPv6 router ospf area 0
SwitchA (config-if-vlan2)#exit
SwitchA(config)#exit
SwitchA#
```

Layer 3 switch SwitchB:

Enable OSPFv3 protocol, configure router ID

```
SwitchB(config)#router IPv6 ospf
SwitchB (config-router)#router-id 192.168.2.2
```

Configure interface vlan1 address, vlan2 IPv6 address and affiliated OSPFv3 area

```
SwitchB#config
SwitchB(config)# interface vlan 1
SwitchB(config-if-vlan1)# IPv6 address 2010:1:1::2/64
SwitchB(config-if-vlan1)# IPv6 router ospf area 0
SwitchB(config-if-vlan1)#exit
SwitchB(config)# interface vlan 3
SwitchB(config-if-vlan3)# IPv6 address 2020:1:1::1/64
SwitchB(config-if-vlan3)# IPv6 router ospf area 1
SwitchB(config-if-vlan3)#exit
SwitchB(config)#exit
SwitchB#
```

Layer 3 switch SwitchC:

```
! Enable OSPFv3 protocol, configure router ID
SwitchC(config)#router IPv6 ospf
SwitchC(config-router)#router-id 192.168.2.3
```

Configure interface vlan3 IPv6 address and affiliated OSPFv3 area

```
SwitchC#config
SwitchC(config)# interface vlan 3
SwitchC(config-if-vlan3)# IPv6 address 2020:1:1::2/64
SwitchC(config-if-vlan3)# IPv6 router ospf area 1
SwitchC(config-if-vlan3)#exit
SwitchC(config)#exit
SwitchC#
```

Layer 3 switch SwitchD:

```
!   Enable OSPFv3 protocol, configure router ID
SwitchD(config)#router IPv6 ospf
SwitchD(config-router)#router-id 192.168.2.4
```

Configure interface vlan3 IPv6 address and affiliated OSPFv3 area

```
SwitchD#config
SwitchD(config)# interface vlan 3
SwitchD(config-if-vlan3)# IPv6 address 2030:1:1::2/64
SwitchD(config-if-vlan3)# IPv6 router ospf area 0
SwitchD(config-if-vlan3)#exit
SwitchD(config)#exit
SwitchD#
```

Layer 3 switch SwitchE:

```
!   Startup OSPFv3 protocol, configure router ID
```

```
SwitchE(config)#router IPv6 ospf
SwitchE(config-router)#router-id 192.168.2.5
Configure interface IPv6 address and affiliated OSPFv3 area
SwitchE#config
SwitchE(config)# interface vlan 2
SwitchE(config-if-vlan2)# IPv6 address 2100:1:1::2/64
SwitchE(config-if-vlan2)# IPv6 router ospf area 0
SwitchE(config-if-vlan2)#exit
Configure interface vlan3 IPv6 address and affiliated area
SwitchE(config)# interface vlan 3
SwitchE(config-if-vlan3)# IPv6 address 2030:1:1::1/64
SwitchE(config-if-vlan3)# IPv6 router ospf area 0
SwitchE(config-if-vlan3)#exit
SwitchE(config)#exit
SwitchE#
```

1.7.5 OSPFv3 Troubleshooting Help

In the process of configuring and implementing OSPFv3, physical connection, configuration false probably leads to OSPFv3 protocol doesn’t work. Therefore, the customers should give their attention to it.

First of all, to ensure correct physical connection, firstly;

Secondly, to ensure interface and link protocol are UP (execute show interface instruction);

And configure IPv6 address of the different net segment on every interface.

To startup OSPFv3 protocol (execute router IPv6 OSPF instruction), and configure affiliated OSPFv3 area on relative interface.

And then, consider OSPFv3 protocol characteristic —— OSPFv3 backbone area (area 0) must be continuous. If it doesn’t ensure that virtual link is implemented continuously, all of not area 0 only can be connected by area 0 and other not area 0, not directly connected by not area 0; The border Layer 3 switch is a part of this Layer 3 switch interface belongs to area 0, and another part of interface belongs to not area 0; for multi-access net etc like broadcast, Layer 3 switch DR needs vote and appoint; for each OSPFv3 process must not configure router ID of 0.0.0.0 address.

If OSPFv3 routing problem still can’t be solved by debugging, please use debug instructions like debug IPv6 OSPF packet/events etc, and copy DEBUG information in 3 minutes, then send them to our technical service center.

1.7.5.1 Monitor And Debug Command

1.7.5.1.1 debug ipv6 ospf ifsm

Command: [no]debug ipv6 ospf ifsm [status|events|timers]

Function: Open debugging switches showing the OSPF interface states; the "**[no]debug ospf ifsm [status|events|timers]**" command closes this debugging switches.

Default: Closed.

Command Mode: Admin mode.

Example:

```
Switch#debug ipv6 ospf ifsm
1970/01/01 01:11:44 IMI: IFSM[Vlan1]: Hello timer expire
1970/01/01 01:11:44 IMI: IFSM[Vlan2]: Hello timer expire
```

1.7.5.1.2 debug ipv6 ospf lsa

Command: [no]debug ipv6 ospf lsa [generate|flooding|install|maxage|refresh]

Function: Open debugging switches showing showing link state announcements; the "**[no]debug ospf lsa [generate|flooding|install|maxage|refresh]**" closes the debugging switches.

Default: Closed.

Command Mode: Admin mode.

1.7.5.1.3 debug ipv6 ospf nfsm

Command: [no]debug ipv6 ospf nfsm [status|events|timers]

Function: Open debugging switches showing showing OSPF neighbor state machine; the "**[no]debug ipv6 ospf nfsm [status|events|timers]**"command closes this debugging switch.

Default: Closed.

Command Mode: Admin mode.

```
Switch#debug ipv6 ospf nfsm
1970/01/01 01:14:07 IMI: NFSM[192.168.2.3-000007d4]: LS update timer expire
1970/01/01 01:14:07 IMI: NFSM[192.168.2.1-000007d3]: LS update timer expire
1970/01/01 01:14:08 IMI: NFSM[192.168.2.1-000007d3]: Full (HelloReceived)
1970/01/01 01:14:08 IMI: NFSM[192.168.2.1-000007d3]: nfsm_ignore called
1970/01/01 01:14:08 IMI: NFSM[192.168.2.1-000007d3]: Full (2-WayReceived)
```

1.7.5.1.4 debug ipv6 ospf nsm

Command: [no]debug ipv6 ospf nsm [interface|redistribute]

Function: Open debugging switches showing showing OSPF NSM, the "**[no]debug ipv6 ospf nsm [interface|redistribute]**" command closes this debugging switch.

Default: Closed.

Command Mode: Admin mode.

1.7.5.1.5 debug ipv6 ospf packet

Command: [no]debug ipv6 ospf packet [dd|detail|hello|ls-ack|ls-request|ls-update|recv|send]

Function: Open debugging switches showing OSPF packet messages; the "**[no]debug ipv6 ospf packet [dd|detail|hello|ls-ack|ls-request|ls-update|recv|send]**" command closes this debugging switch.

Default: Closed.

Command Mode: Admin mode.

1.7.5.1.6 debug ipv6 ospf route

Command: [no]debug ipv6 ospf route [ase|ia|install|spf]

Function: Open debugging switches showing OSPF related routes; the "**[no]debug ipv6 ospf route [ase|ia|install|spf]**" command closes this debugging switch.

Default: Closed.

Command Mode: Admin mode.

1.7.5.1.7 debug ipv6 ospf events

Command: [no]debug ipv6 ospf events [abr|asbr|os|router|vlink]

Function: Open debugging switches showing OSPF events. The "no debug ipv6 ospf events [abr|asbr|os|router|vlink]" command closes this debugging switch.

Default: Closed.

Command Mode: Admin mode.

Example:

```
Switch#debug ipv6 ospf events
1970/01/01 01:10:35 IMI: ROUTER[Process:(null)]: GC timer expire
```

1.7.5.1.8 debug ipv6 ospf redistribute message send

Command: debug ipv6 ospf redistribute message send
no debug ipv6 ospf redistribute message send

Function: To enable/disable debugging of sending command from IPv6 OSPF process redistributed to other IPv6 OSPF process routing.

Parameter: None.

Default: Disabled.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 ospf redistribute message send
```

1.7.5.1.9 debug ipv6 ospf redistribute route receive

Command: debug ipv6 ospf redistribute route receive

no debug ipv6 ospf redistribute route receive

Function: To enable/disable debugging of received routing message from nsm for IPv6 OSPF process.

Parameter: None.

Default: Disabled.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 ospf redistribute route receive
```

1.7.5.1.10 show ipv6 ospf

Command: show ipv6 ospf [<*tag*>]

Function: Display OSPF global and area messages

Parameter: ***<tag>*** is the process tag which is a character string

Default: Not displayed

Command Mode: All modes

Example:

```
Routing Process "OSPFv3 (*null*)" with ID 192.168.2.2
  SPF schedule delay 5 secs, Hold time between SPFs 10 secs
  Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
  Number of external LSA 0. Checksum Sum 0x0000
  Number of AS-Scoped Unknown LSA 0
  Number of LSA originated 6
  Number of LSA received 14
  Number of areas in this router is 1
  Area BACKBONE(0)
  Number of interfaces in this area is 2
  SPF algorithm executed 6 times
  Number of LSA 8.   Checksum Sum 0x43D52
  Number of Unknown LSA 0
```

1.7.5.1.11 show ipv6 ospf database

Command: show ipv6 ospf [<*tag*>] database

[router   [adv-router <*advertiser_router*>]

|network   [adv-router <*advertiser_router*>]

| intra-prefix [adv-router <*advertiser_router*>]

| link   [adv-router <*advertiser_router*>]

| external   [adv-router <*advertiser_router*>]

| inter-prefix [adv-router <*advertiser_router*>]

| inter-router [adv-router <*advertiser_router*>]]

Function: Display the OSPF link state data base message

Parameter: ***<tag>*** is the process tag which is a character string

<advertiser_router> is the ID of Advertising router, shown in IPv4 address format

Default: Not displayed

Command Mode: All modes

Example: According to the output messages of this command, we can view the OSPF link state database messages

Use show ipv6 ospf database command will be able to show LSA messages of the OSPF routing protocol

For Example, the displayed messages are:

```
                 OSPFv3 Router with ID (192.168.2.2) (Process *null*)
                       Link-LSA (Interface Vlan1)
Link State ID   ADV Router        Age  Seq#        CkSum   Prefix
0.0.7.211         192.168.2.2     1409 0x80000001 0x6dda        1
0.0.7.212         192.168.2.3     1357 0x80000001 0x248e        1
                       Link-LSA (Interface Vlan2)
Link State ID   ADV Router        Age  Seq#        CkSum   Prefix
0.0.7.211         192.168.2.1     1450 0x80000001 0xa565        1
0.0.7.212         192.168.2.2     1399 0x80000001 0x4305        1
                       Router-LSA (Area 0.0.0.0)
Link State ID   ADV Router        Age  Seq#        CkSum     Link
0.0.0.0           192.168.2.1     1390 0x80000006 0x9fe2       1
0.0.0.0           192.168.2.2     1354 0x80000007 0x4af5       2
0.0.0.0           192.168.2.3     1308 0x80000004 0xbbc4       1
                       Network-LSA (Area 0.0.0.0)
Link State ID   ADV Router        Age  Seq#        CkSum
0.0.7.211         192.168.2.1     1390 0x80000001 0x897e
0.0.7.211         192.168.2.2     1354 0x80000001 0x9b69
                       Intra-Area-Prefix-LSA (Area 0.0.0.0)
Link State ID   ADV Router        Age  Seq#        CkSum   Prefix  Reference
0.0.0.1           192.168.2.1     1389 0x80000005 0x7e2e       1   Router-LSA
0.0.0.2           192.168.2.1     1389 0x80000001 0x22cb       1   Network-LSA
0.0.0.1           192.168.2.3     1306 0x80000002 0xd0d7       1   Router-LSA
```

Displayed information’s	Explanations
Link-LSA (Interface Vlan1)	Link LSA messages of interface Vlan1
Router-LSA (Area 0.0.0.0)	Router LSA messages in Area 0

Network-LSA (Area 0.0.0.0)	Network LSA in Area 0
Intra-Area-Prefix-LSA (Area 0.0.0.0)	Intra-domain Prefix LSA in Area 0

1.7.5.1.12 show ipv6 ospf interface

Command: show ipv6 ospf interface [ifname]|vlan <vlan-id>

Function: Display the OSPF interface messages.

Parameter: ifname is the name of the interface.

Default: Not displayed.

Command Mode: All modes.

Example:

```
Loopback is up, line protocol is up
OSPFv3 not enabled on this interface
Vlan1 is up, line protocol is up
Interface ID 2003
IPv6 Prefixes
fe80::203:fff:fe01:257c/64 (Link-Local Address)
2001:1:1::1/64
OSPFv3 Process (*null*), Area 0.0.0.0, Instance ID 0
Router ID 192.168.2.2, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 192.168.2.2
Interface Address fe80::203:fff:fe01:257c
Backup Designated Router (ID) 192.168.2.3
Interface Address fe80::203:fff:fe01:d28
Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:10
Neighbor Count is 1, Adjacent neighbor count is 1
Vlan2 is up, line protocol is up
Interface ID 2004
IPv6 Prefixes
fe80::203:fff:fe01:257c/64 (Link-Local Address)
2000:1:1::1/64
OSPFv3 Process (*null*), Area 0.0.0.0, Instance ID 0
Router ID 192.168.2.2, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State Backup, Priority 1
Designated Router (ID) 192.168.2.1
Interface Address fe80::203:fff:fe01:429e
Backup Designated Router (ID) 192.168.2.2
Interface Address fe80::203:fff:fe01:257c
```

```
Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:10
Neighbor Count is 1, Adjacent neighbor count is 1
```

Displayed information	Explanations
Vlan1 is up, line protocol is up	Let the interface up both logically and physically
IPv6 Prefixes fe80::203:fff:fe01:257c/64 (Link-Local Address) 2001:1:1::1/64	IPv6 address of the interface and the length of the prefix
OSPFv3 Process (*null*)	Ospfv3 process the interface belongs
Area 0.0.0.1	Area the interface belongs
Instance ID 0	Instance ID is 0
Router ID 192.168.2.2, Network Type BROADCAST, Cost: 10	Process ID; Router ID; Network Type; Cost
Transmit Delay is 1 sec, State DR, Priority 1	LAS transmission delay on the interface; state; electing the priority of the layer 3 switch.
Designated Router (ID) 192.168.2.2 Interface Address fe80::203:fff:fe01:257c	Specifying layer 3 switch
Backup Designated Router (ID) 192.168.2.3 Interface Address fe80::203:fff:fe01:d28	Back up designated layer 3 switch
Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5 Hello due in 00:00:10	OSPF protocol timer; including hello packet, poll interval packets, router dead, router retransmission.
Neighbor Count is 1, Adjacent neighbor count is 1	Numbers of the adjacent layer 3 switch; number of the layer 3 switches established with neighbor relation

1.7.5.1.13 show ipv6 ospf neighbor

Command: show ipv6 ospf [<*tag*>] neighbor [<*neighbor_id*> | <ifname> detail | detail]

Function: Show OSPF adjacent point messages

Parameter: <*tag*> is process tag, which is a character string

<*neighbor_id*> is the neighbor ID shown in IPv4 address format

detail: Show neighbor details

<*ifname*> name of the interface

Default: Not displayed

Command Mode: All modes

Usage Guide: OSPF neighbor state can be checked by viewing the output of this command

Example:

```
OSPFv3 Process (*null*)
Neighbor ID        Pri     State              Dead Time      Interface    Instance ID
192.168.2.3          1     Full/Backup        00:00:29       Vlan1          0
192.168.2.1          1     Full/DR            00:00:38       Vlan2          0              Vlan1
```

Displayed information	Explanation
Neighbor ID	Neighbor ID
Instance ID	Instance ID
Address	IP address of neighboring layer 3 switch
Interface	Interface the neighbor belongs
State	Neighbor relationship state
Pri	Priority

1.7.5.1.14 show ipv6 ospf route

Command: show ipv6 ospf [<*tag*>] route

Function: Show the OSPF route table messages

Parameter: ***<tag>*** is the processes tag, which is a character string

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ipv6 ospf route
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
          E1 - OSPF external type 1, E2 - OSPF external type 2
    Destination                                          Metric
       Next-hop
O   2000:1:1::/64                                             10
       directly connected, Vlan2
O   2001:1:1::/64                                             10
       directly connected, Vlan1
O   3000:1:1::/64                                             20
       via fe80::203:fff:fe01:429e, Vlan2
O   3003:1:1::/64                                             20
       via fe80::203:fff:fe01:d28, Vlan1
```

1.7.5.1.15 show ipv6 ospf redistribute

Command: show ip ospf v6 [<*process-tag*>] redistribute

Function: To display the routing message redistributed from external process of OSPF.

Parameter: IPv6 OSPF is the tag id, to display all the routing message redistributed from external process of ipv6 ospf if there is no parameter.

Default: None.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ipv6 ospf redistribute
        ospf process abc redistribute information：
          ospf process def
          bgp
        ospf process def redistribute information：
          ospf process abc

Switch#show ipv6 ospf abc redistribute
        ospf process abc redistribute information：
          ospf process def
bgp
```

1.7.5.1.16 show ipv6 ospf topology

Command: show ipv6 ospf [<*tag*>] topology [area <*area-id*>]

Function: Show messages of OSPF topology

Parameter: **<*tag*>** is the processes tag, which is a character string

<*area-id*> is an area ID which could be shown in digits ranging between 0～4294967295, or an IPv4 address

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ipv6 ospf topology
OSPFv3 Process (*null*)
OSPFv3 paths to Area (0.0.0.0) routers
Router ID           Bits   Metric      Next-Hop                Interface
192.168.2.1                 10           192.168.2.1             Vlan2
192.168.2.2                 --
192.168.2.3                 10           192.168.2.3             Vlan1
```

1.7.5.1.17 show ipv6 ospf virtual-links

Command: show ipv6 ospf [<*tag*>] virtual-links

Function: Show OSPF virtual link messages

Parameter: ***<tag>*** is the processes tag, which is a character string

Default: Not displayed

Command Mode: All modes

Example:

```
Switch#show ipv6 ospf virtual-links
Virtual Link VLINK1 to router 5.6.7.8 is up
Transit area 0.0.0.1 via interface Vlan1, instance ID 0
Local address 3ffe:1234:1::1/128
Remote address 3ffe:5678:3::1/128
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 Up
```

1.8 BGP

1.8.1 BGP Introduction

BGP stands for a Border Gateway Protocol.It's a dynamic routing protocol inter-autonomous system. Its basic function is automatically exchanging routing information without loops. By exchanging routing reachable information with autonomous number of AS sequence attributes, BGP could create autonomous topological map to eliminate routing loop and implement policies configured by users. Generally, the switches in an AS may use several IGPs (Interior Gateway Protocol) in order to exchange routing information in the AS, such as RIP and OSPF which are IGPs; and exchange information among ASes with EGP (External Gateway Protocol). For example, BGP is one kind of EGP. The AS is usually established on a single administrative department. BGP is often used on the switches among ISPs or the departments of Multi-national Corporation.

BGP has been used since1989, its earliest three versions are RFC1105（BGP-1）、RFC1163（BGP-2）and RFC1267（BGP-3）.Currently, the most popular one is RFC1771（BGP-4）. The ES4700 series switch supports BGP-4.

1．Characteristics of BGP-4

BGP-4 is suitable for the distributed structure and supports Classless InterDomain Routing (CIDR). BGP-4 is becoming the virtual external routing protocol standard used for the global Internet. The features of BGP-4 are as follows.

BGP is an external routing protocol, unlike interior routing protocol, such as OSPF and RIPng, BGP can’t discovery and calculate routes, but it can control the transmission of routes and select the best route.

By carrying AS routing information in the updating route, the problem of Routing Loops can be resolved

BGP uses TCP on port 179 as its transport protocol, this could enhance the reliability of the protocol.

BGP-4 supports CIDR (Classless InterDomain Routing), which is an important improvement to BGP-3. CIDR has a brand new way to look on IP address; it doesn’t distinguish class A , Class B and class C network. For instance, an illegal class C address 192.213.0.0 255.255.0.0 can be represented as 192.213.0.0/16 by CIDR which is a legal super network. /16 represents that the network number is formed by 16 bits from the beginning left of the address. The introduction of CIDR abbreviates the route aggregation. The route aggregation is the process of combining several different routes. So notifying several routes can be changed to notify only one route which decreases the route table.

When updating route, BGP send only incremental route. The bandwidth occupied by BGP transmission is reduced greatly and it is suitable for the mass routing information transmitted on the internet

For political and economical reasons, each AS expects to filter and control the route, BGP-4 provides abundant route policies which make BGP-4 more extendable to encourage the internet development.

2. The Overview of BGP-4 operation

Unlike RIP and OSPF protocols, BGP protocol is connection oriented. BGP switches must establish connection to exchange routing information. The operation of BGP protocol is driven by messages and the messages can be divided into four kinds:

Open message----It’s the first message which is sent after a TCP connection is established. It is used to create BGP connecting relation among BGP peers. Some parameters in Open Message are used to negotiate if a connection could be established among BGP peers.

Keepalive Message ----- it’s the message to check connection availability. It’s usually sent periodically to keep BGP connection. If this message or Update message is not received within holdtime time, BGP connection is closed.

Update Message----- it’s the most important message in the BGP system. It’s used to exchange routing information among peers. The switches exchange not only updated routing information, but also unavailable or canceled routing information. It consists of three parts: unreachable route, NLRI(Network LayerReachability Information) and Path Attributes.

Notification Message------it’s the mistake notification message. When a BGP speaker receives this message, it shutdowns the BGP connections with its neighbors

BGP-4 is connection oriented. BGP acts as higher protocol and runs on the particular equipments. When detecting a neighbor, a TCP session is established and maintained. Then the exchanging and synchronization of the route table will be carried out. By sending the whole BGP route table the routing information is exchanged only when the system initiates. After that, the routing information is exchanged only when the updated routing information is available. Only incremental update message is exchanged. BGP-4 maintains links and sessions periodically through keepalive message. That is sending and receiving keepalive message periodically to check if the connections are normal.

The switches that participate the BGP session are called BGP speaker. It continuously receives or generates new routing information and advertises it to other BGP speakers. When a BGP speaker receives a new routing notification from other AS, if this route is better than the presently known route or there is no acceptable route, it sends this route to all the other BGP speakers of the AS. A BGP speaker calls other speakers that exchange route information with it as neighbors or peers. Several relevant neighbors can constitute a peer group. BGP operates on the switches in the following two manners:

- IBGP:Internal BGP
- EBGP:External BGP

When BGP runs in the same AS, it's called IBGP. When in the different AS, it's called EBGP. Generally, the outer neighbors are connected physically and the inner neighbors can be in any place of the AS. The difference is finally shown in the dealing manner of BGP to routing information. The equipments may check the AS numbers of the Open Message from neighbors to decide treating the neighbor switches as the external neighbor or as the interior neighbor.

IBGP is used in the AS. It sends message to all the BGP neighbors in the AS. IBGP exchanges AS routing information in a big organization. Attention, the switches in the AS needn't be connected physically. Only if the switches are in the same AS, they can be neighbors each other. Because BGP can't detect route, the route tables of other inner route protocols (such as static route, direct route, OSPF and RIP) need contain neighbor IP addresses and these routes are used to exchange information among BGPs. In order to avoid routing loops, when a BGP speaker receives a route notification from inner neighbor, it would not notify this route to other inner neighbors.

EBGP is used among the AS, and it transmits routing information to the BGP neighbors of outer ASes. EBGP need physical connection and share the same medium. Because EBGP need physical connection, the boundary equipments between two AS are usually running EBGP. When a BGP speaker receives routing information from outer neighbors, it notifies these routes to other inner neighbors.

3. Route attribute

BGP-4 can share and query inner IP route table through relevant mechanisms, but it has its own route table. In the BGP route table, each route has a network number, AS listing information (also called AS path) that it passed and some routing attributes (such as origin). The routing attribute that BGP-4 used is very complex, this attribute can be used as metrics to select path.

4. Route-selecting policy of BGP

When receiving BGP notification about a same route from several neighbors, selecting the best route need to be take into account after routing filtering. This process is called BGP route selecting process. BGP route selecting process will start only when the following conditions are fulfilled:

- ➢ The switch's route must be next hop reachable. That is in the route table there is the route that can reach the next hop.
- ➢ BGP must be synchronized with IGP (unless asynchronism is configured; only restricted to IBGP)

BGP route selecting process is based on the BGP attribute. When there are several routes that indicate the same destination, BGP need select the best route to the destination. The decision-making process is as the following:

1．Select the route with the most weight first;

2. If the weights are the same, select the route with the most local preference;

3. If the local preferences are the same, select the route generated by local switch.

4. If the local preferences are the same and there is no route generated by local switch, select the route with the shortest AS path;

5. If the AS paths are the same, select the route with the lowest “origin“ type （IGP<EGP<INCOMPLETE）;

6. If the “origin” types are the same, select the route with the lowest MED attribute. Unless activating command “bgp always-compare-med”, this comparison is only available among the routes from the same neighbor AS.

7. If the MED attributes are the same, EBGP is preferable to outer confederation and outer confederation is preferable to IBGP.

8. If it's still the same by now, BGP router ID （router ID）is used to break the balance. The best route is the one from the least router ID.

1.8.2 BGP Configuration Task List

The BGP configuration tasks include basic and advanced tasks. Basic BGP configuration tasks include the following:

1．Enable BGP Routing (required)

2．Configure BGP Neighbors (required)

3．Administrate the change of routing policy
4．Configure BGP Weights
5．Configure BGP Route Filtering policy basing on Neighbors
6．Configure Next-Hop of BGP
7．Configure Multi-Hop of EGBP
8．Configure BGP Session Identifier
9．Configure BGP Version

Advanced BGP configuration tasks include the following:

1．Use Route Maps to Modify Route
2．Configure Route Aggregation
3．Configure BGP Community Filtering
4．Configure BGP Confederation
5．Configure a Route Reflector
6．Configure Peer Groups
7．Configure Neighbors and Peer Groups' Parameters
8．Adjust BGP Timers
9．Adjust BGP Announcement Interval
10．Configure the default Local preference
11．Allow to Transfer Default Route
12．Configure BGP's MED Value
13．Configure BGP Routing Redistribution
14．Configure BGP Route Dampening
15．Configure BGP capability Negotiation
16．Configure Routing Server
17．Configure Path-Selected Rule
18．Redistribution of OSPF Routing to BGP
（1）Enable Redistribution of OSPF routing to BGP
（2）Display the information about configuration of redistribution of OSPF routing to BGP

Ⅰ． Basic BGP configuration tasks

1.Enable BGP Routing

Command	Explanation
Global mode	
router bgp <*as-id*> **no router bgp <*as-id*>**	Enable BGP, the **"no router bgp <*as-id*>"**command disenable BGP process.
Router configuration mode	

network <*ip-address/M*> **no network <*ip-address/M*>**	Set the network that BGP will announce, the **no network <*ip-address/M*>** command cancels the network that will be announced.

2. Configure BGP Neighbors

Command	Explanation
Router configuration mode	
neighbor {<*ip-address*>|<*TAG*>} remote-as <*as-id*> **no neighbor {<*ip-address*>|<*TAG*>} [remote-as <*as-id*>]**	Specify a BGP neighbor, the **no neighbor {<*ip-address*>|<*TAG*>} [remote-as <*as-id*>]** command deletes the neighbor.

3．Administrate the change of routing policy

（1）Configure hard reconfiguration.

Command	Explanation
Admin Mode	
clear ip bgp {<*>|<*as-id*>|external|peer-group <*NAME*>|<*ip-address*>}	Configure hard reconfiguration.

（2） Configure outbound soft reconfiguration.

Command	Explanation
Admin Mode	
clear ip bgp {<*>|<*as-id*>|external|peer-group <NAME>|<*ip-address*>} soft out	Configure outbound soft reconfiguration.

（3）Configure inbound soft reconfiguration.

Command	Explanation
Router configuration mode	
neighbor { <*ip-address*> | <*TAG*> } soft-reconfiguration inbound **no neighbor { <*ip-address*> | <*TAG*> } soft-reconfiguration inbound**	This command can store routing information from neighbors and peers; the **no neighbor { <*ip-address*> | <*TAG*> } soft-reconfiguration inbound** command cancels the storage of routing information.
Admin Mode	

Clear ip bgp {<*>|<*as-id*>| external|peer-group *<NAME>*|<*ip-address*>} soft in	Configure BGP inbound soft reconfiguration.

4．Configure BGP Weights

Command	Explanation
Router configuration mode	
neighbor { *<ip-address>* | *<TAG>* } weight *<weight>* no neighbor { <ip-address> | <TAG> }	Configure BGP neighbor weights; the **no neighbor { <ip-address> | <TAG> }** command recovers default weights.

5．Configure BGP Route Filtering policy based on neighbor

Command	Explanation
Router configuration mode	

**neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}
no neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}**
	Filter neighbor routing updating information. The **no neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}** command cancels routing filter.

6．Configure Next-Hop

1）Set Next-Hop as the switch’s address

Command	Explanation
BGP configuration mode	
neighbor { *<ip-address>* | *<TAG>* } next-hop-self no neighbor { *<ip-address>* | *<TAG>* } next-hop-self	While sending route Next-Hop set Next-Hop as the switch’s address; the **no neighbor { *<ip-address>* | *<TAG>* } next-hop-self** command cancels the setting.

2）Cancel default Next-Hop through route map

Command	Explanation
Route mapped configuration command	

set ip next-hop <*ip-address*> **no set ip next-hop**	Set the Next-Hop attribute of outbound route. The **no set ip next-hop** command cancels this setting.

7．Configure EGBP Multi-Hop

If the connections with outer neighbors are not direct, the following command can configure neighbor Multi-Hop.

Command	Explanation
BGP configuration mode	
neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>] **no neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>]**	Configure the allowance of EBGP connection with other networks that are not connected directly; the **no neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>]** command cancels the setting.

8．Configure BGP session identifier

Command	Explanation
BGP configuration mode	
bgp router-id <*ip-address*> **no bgp router-id**	Configure the router-id value; the **no bgp router-id** command recovers the default value.

9． Configure the BGP Version

Command	Explanation
BGP configuration mode	
neighbor {<*ip-address*> | <*TAG*>} version <value> **no neighbor {<*ip-address*> | <*TAG*>} version**	Set the version used by BGP neighbors; the **no neighbor {<*ip-address*> | <*TAG*>} version** command recovers default setting. Presently only supporting version 4th .

Ⅱ．Advanced BGP configuration tasks

1．Use Route Maps to Modify Route

Command	Explanation
BGP configuration mode	

neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name >* {in | out}
no neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name >* {in | out}
	Apply a route map to incoming or outgoing routes; the **no neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name >* {in | out}** command cancels the settings of routing maps.

2. **Configure Route Aggregation**

Command	Explanation
BGP configuration mode	
aggregate-address *<ip-address/M>* [summary-only] [as-set] **no aggregate-address *<ip-address/M>* [summary-only] [as-set]**	Create an aggregate entry in the BGP routing table; the **no aggregate-address *<ip-address/M>* [summary-only] [as-set]** command cancels the aggregate entry.

3. **Configure BGP Community Filtering**

Command	Explanation
BGP configuration mode	
neighbor {*<ip-address>* | *<TAG>*} send-community **no neighbor {*<ip-address>* | *<TAG>*} send-community**	Allow the routing updates with community attributes sending to BGP neighbors; the **no neighbor {*<ip-address>* | *<TAG>*} send-community** command enables the route without community attributes.

4. **Configure BGP Confederation**

Command	Explanation
BGP configuration mode	
bgp confederation identifier *<as-id>* **no bgp confederation identifier *<as-id>***	Configure a BGP AS confederation identifier; the **no bgp confederation identifier *<as-id>*** command deletes the BGP AS confederation identifier

bgp confederation peers <*as-id*> [<*as-id*>..] **no bgp confederation peers <*as-id*> [<*as-id*>..]**	Configure the AS affiliated to the AS confederation; the **no bgp confederation peers <*as-id*> [<*as-id*>..]**command deletes the AS from the AS confederation.

5. Configure a Route Reflector

（1） The following commands can be used to configure route reflector and its clients.

Command	Explanation
BGP configuration mode	
neighbor <*ip-address*> route-reflector-client **no neighbor <*ip-address*> route-reflector-client**	Configure the current switch as route reflector and specify a client. the **no neighbor <*ip-address*> route-reflector-client** command format deletes a client.

（2） If there are more than one route reflectors in the cluster, the following commands can configure cluster-id

Command	Explanation
BGP configuration mode	
bgp cluster-id <*cluster-id*> **no bgp cluster-id**	Configure cluster id; format “no” of the **no bgp cluster-id** command cancels the cluster id configuration.

（3） If the route reflector from clients to clients is needed, the following commands can be used.

Command	Explanation
BGP configuration mode	
bgp client-to-client reflection **no bgp client-to-client reflection**	Configure the allowance of the route reflector from clients to clients; the no bgp client-to-client reflection command forbids this allowance.

t

6. Configure Peer Groups

（1） Create peer groups

Command	Explanation
BGP configuration mode	
neighbor <*TAG*> peer-group **no neighbor <*TAG*> peer-group**	Create peer groups; the **no neighbor <*TAG*> peer-group** command deletes peer groups.

（2） Add neighbors to peers groups

Command	Explanation
BGP configuration mode	
neighbor <*ip-address*> peer-group <*TAG*> **no neighbor <*ip-address*> peer-group <*TAG*>**	Make a neighbor a member of the peer group. the **no neighbor <*ip-address*> peer-group <*TAG*>** command cancels the specified member.

7．Configure neighbors and peer Groups' parameters

Command	Explanation
BGP configuration mode	
neighbor {<*ip-address*> | <*TAG*>} remote-as <*as-id*> **no neighbor {<*ip-address*> | <*TAG*>} remote-as <*as-id*>**	Specify a BGP neighbor; format "no" of the **no neighbor {<*ip-address*> | <*TAG*>} remote-as <*as-id*>** command deletes the neighbor.
neighbor { <*ip-address*> | <*TAG*> } description <*.LINE*> **no neighbor { <*ip-address*> | <*TAG*> } description**	Associate a description with a neighbor; the **no neighbor { <*ip-address*> | <*TAG*> } description** command deletes this description.
neighbor { <*ip-address*> | <*TAG*> } default-originate [route-map <*NAME*>] **no neighbor { <*ip-address*> | <*TAG*> } default-originate [route-map <*NAME*>]**	Permit to send the default route 0.0.0.0; the **no neighbor { <*ip-address*> | <*TAG*> } default-originate [route-map <*NAME*>]** command cancels sending default route.
neighbor { <*ip-address*> | <*TAG*> } send-community **no neighbor { <*ip-address*> | <*TAG*> } send-community**	Configure the community attributes sent to the neighbor .

neighbor { *<ip-address>* | *<TAG>* } timers *<keepalive> <holdtime>* **no neighbor { *<ip-address>* | *<TAG>* } timers**	Configure a particular neighbor's keep-alive and hold-time timer; the **no neighbor { *<ip-address>* | *<TAG>* } timers** command recovers the default value.
neighbor {*<ip-address>* | *<TAG>*} advertisement-interval *<seconds>* **no neighbor {*<ip-address>* | *<TAG>*} advertisement-interval**	Configure the min interval of sending BGP routing information; the **no neighbor {*<ip-address>* | *<TAG>*} advertisement-interval** command recovers the default value.
neighbor {*<ip-address>* | *<TAG>*} ebgp-multihop [<*1-255*>] **no neighbor {*<ip-address>* | *<TAG>*} ebgp-multihop**	Configure the allowance of EBGP connections with networks connected indirectly; the **no neighbor {*<ip-address>* | *<TAG>*} ebgp-multihop** command cancels this setting.
neighbor { *<ip-address>* | *<TAG>* } weight <weight> **no neighbor { *<ip-address>* | *<TAG>* } weight**	Configure BGP neighbor weights; the **no neighbor { *<ip-address>* | *<TAG>* } weight** command recovers the default weights.

neighbor { *<ip-address>* | *<TAG>* } distribute-list { *<access-list-number>* | *<name>* } { in | out }
no neighbor { *<ip-address>* | *<TAG>* } distribute-list { *<access-list-number>* | *<name>* } { *in* | *out* }
	Filter neighbor route update; format "no" of the **no neighbor { *<ip-address>* | *<TAG>* } distribute-list { *<access-list-number>* | *<name>* } { *in* | *out* }** command cancels route filtering.

neighbor { *<ip-address>* | *<TAG>* } route-reflector-client **no neighbor { *<ip-address>* | *<TAG>* } route-reflector-client**	Configure the current switch as route reflector and specify a client; the **no neighbor { *<ip-address>* | *<TAG>* } route-reflector-client** command deletes a client.
neighbor { *<ip-address>* | *<TAG>* } next-hop-self **no neighbor { *<ip-address>* | *<TAG>* } next-hop-self**	When sending route, configure Next-Hop as its address; the **no neighbor { *<ip-address>* | *<TAG>* } next-hop-self** command cancels the setting.

neighbor { *<ip-address>* | *<TAG>* } version *<value>* **no neighbor { *<ip-address>* | *<TAG>* } version**	Specify the BGP version communicating with BGP neighbors; the **no neighbor { *<ip-address>* | *<TAG>* } version** command recovers default setting.

neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name>* {in | out}
no neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name>* {in | out}
	Apply a route map to incoming or outgoing routes; the **no neighbor { *<ip-address>* | *<TAG>* } route-map *<map-name>* {in | out}** command cancels the setting of route reflector.

neighbor { *<ip-address>* | *<TAG>* } soft-reconfiguration inbound **no neighbor { *<ip-address>* | *<TAG>* } soft-reconfiguration inbound**	Store the route information from neighbor or peers; the **no neighbor { *<ip-address>* | *<TAG>* } soft-reconfiguration inbound** command cancels the storage.
neighbor { *<ip-address>* | *<TAG>* } shutdown **no neighbor { *<ip-address>* | *<TAG>* } shutdown**	Shutdown BGP neighbor or peers; the **no neighbor { *<ip-address>* | *<TAG>* } shutdown** command activates the closed BGP neighbor or peers.

8．Adjust BGP Timers

（1） Configure the BGP timer of all the neighbors

Command	Explanation
BGP configuration mode	
timers bgp *<keepalive>* *<holdtime>* **no timers bgp**	Configure the BGP timers of all the neighbors; the no timers bgp command recovers the default value.

（2） Configure the timer value of a particular neighbor

Command	Explanation
BGP configuration mode	

neighbor { *<ip-address>* | *<TAG>* } timers *<keepalive> <holdtime>* **no neighbor { *<ip-address>* | *<TAG>* } timers**	Configure the keepalive and holdtime timer of a particular neighbor; the **no neighbor { *<ip-address>* | *<TAG>* } timers** command recovers the default value.

9. Adjust BGP announcement Interval

Command	Explanation
BGP configuration mode	
neighbor {*<ip-address>* | *<TAG>*} advertisement-interval <seconds> **no neighbor {*<ip-address>* | *<TAG>*} advertisement-interval**	Configure the minimum interval among BGP routes update information; the **no neighbor {*<ip-address>* | *<TAG>*} advertisement-interval** command recovers the default setting.

10. Configure the Local Preference Value

Command	Explanation
BGP configuration mode	
bgp default local-preference <*value*> **no bgp default local-preference**	Change default local preference; the **no bgp default local-preference** command recovers the default value.

11. Enable sending default route

Command	Explanation
BGP configuration mode	
neighbor { *<ip-address>* | *<TAG>* } default-originate **no neighbor { *<ip-address>* | *<TAG>* } default-originate**	Permit sending default route 0.0.0.0; the **no neighbor { *<ip-address>* | *<TAG>* } default-originate** command cancels sending default route.

12. Configure BGP's MED Value

（1） Configure MED value

Command	Explanation
Route map configuration command	

set metric <*metric-value*> **no set metric**	Configure metric value; the **no set metric** command recovers the default value.

（2） Apply route selection based on MED according to the path from different AS

Command	Explanation
BGP configuration mode	
bgp always-compare-med **no bgp always-compare-med**	Permit the MED comparison from different AS; the **no bgp always-compare-med** command forbids the comparison.

13. Configure BGP routing redistribution

Command	Explanation
BGP configuration mode	
redistribute { connected | static | rip | ospf} [metric <*metric*>] [route-map <*NAME*>] **no redistribute { connected | static | rip | ospf}**	Redistribute IGP routes to BGP and may specify the redistributed metric and route reflector; the **no redistribute { connected | static | rip | ospf}** command cancels the redistribution.

14. Configure Route Dampening

Command	Explanation
BGP configuration mode	
bgp dampening [<*1-45*>] [<*1-20000*> <*1-20000*> <*1-255*>] [<*1-45*>] **no bgp dampening**	Enable BGP route dampening and apply the specified parameters; the **no bgp dampening** command stops route dampening

15. Configure BGP capability Negotiation

Command	Explanation
BGP configuration mode	

neighbor {<*ip-address*>|<*TAG*>} capability {dynamic | route-refresh} **no neighbor {<*ip-address*>|<*TAG*>} capability {dynamic | route-refresh}** **neighbor {<*ip-address*>|<*TAG*>} capability orf prefix-list {<*both*>|<*send*>|<*receive*>}** **no neighbor {<*ip-address*>|<*TAG*>} capability orf prefix-list {<*both*>|<*send*>|<*receive*>}** **neighbor {<*ip-address*>|<*TAG*>} dont-capability-negotiate** **no neighbor {<*ip-address*>|<*TAG*>} dont-capability-negotiate** **neighbor {<*ip-address*>|<*TAG*>} override-capability** **no neighbor {<*ip-address*>|<*TAG*>} override-capability** **neighbor {<*ip-address*>|<*TAG*>} strict-capability-match** **no neighbor {<*ip-address*>|<*TAG*>} strict-capability-match**	BGP provides capability negotiation regulation and carry out this capability match while establishing connection. The currently supported capabilities include route update, dynamic capability, outgoing route filtering capability and the address family's capability of supporting the negotiation. Use these command to enable these capabilities, its format "no" close these capabilities .It can also be configured by commands to not do capability negotiation, do strict capability negotiation or not care about the negotiation results

16. Configure Routing Server

Command	Explanation
BGP configuration mode	
neighbor {<*ip-address*>|<*TAG*>} route-server-client **no neighbor {<*ip-address*>|<*TAG*>} route-server-client**	Route server may configure BGP neighbors under EBGP environment to reduce the number of peers that every client has configured; format "no" of the command configures this router as route server and specify the clients it serves, the **no neighbor {<*ip-address*>|<*TAG*>} route-server-client** command can delete clients.

17. Configure Path-selected rules

Command	Explanation

BGP configuration mode	
bgp always-compare-med **no bgp always-compare-med** **bgp bestpath as-path ignore** **no bgp bestpath as-path ignore** **bgp bestpath compare-confed-aspath** **no bgp bestpath compare-confed-aspath** **bgp bestpath compare-routerid** **no bgp bestpath compare-routerid** **bgp bestpath med {[confed] [missing-is-worst]}** **no bgp bestpath med {[confed] [missing-is-worst]}**	BGP may change some path-select rules by configuration to change the best selection and compare MED under EBGP environment through these command, ignore the AS-PATH length, compare the confederation as-path length, compare the route identifier and compare the confederation MED etc. Its format "no" recovers the default route path-selected rules.

18. Redistribution of OSPF Routing to BGP

（1）To enable redistribution of OSPFv2 routing to BGP

Command	Notes
Router bgp configuration mode	
redistribute ospf [<*process-id*>] [route-map<*word*>] **no redistribute ospf [<*process-id*>]**	To enable or disable redistribution of OSPFv2 routing to BGP.

（2）Display the information about configuration of redistribution of OSPF routing to BGP

Command	Notes
Admin mode and configuration mode	
show ip bgp redistribute	To display BGP routing which is redistributed from other routing protocols.
Admin mode	
debug bgp redistribute message send **no debug bgp redistribute message send** **debug bgp redistribute route receive** **no debug bgp redistribute route receive**	To enable or disable debugging messages sent by BGP for redistributing OSPF routing. To enable or disable debugging messages received from NSM for redistributing OSPF routing.

1.8.3 Command For BGP

1.8.3.1 address-family

Command: address-family <*AFI*> <*SAFI*>

Function: Enter address-family mode.

Parameter: <*AFI*> : address-family, such as IPv4、IPv6、VPNv4, etc.

<*SAFI*>: sub address-family, such as unicast 、multicast.

Default: None.

Command Mode: BGP routing mode.

Usage Guide: Since the BGP-4 supports multi-protocol, it is available to get different configuration for each address-family. Actually the configuration outside address-family mode is configuring the default address-family (normally IPv4 unicast). To configure non default mode, enter this address-family mode.

Example: Switch(config-router)# address-family ipv4 unicast

1.8.3.2 address-family ipv4

Command: address-family ipv4 {multicast | unicast | vrf<vrf-name>}

no address-family ipv4 vrf <*vrf-name*>

Function: Enter BGP VRF address-family mode. The "**no address-family ipv4 vrf <*vrf-name*>**" command deletes the configuration of the address-family.

Parameter: <*vrf-name*> specifies the name of VPN routing/forwarding instances.

Command Mode: BGP routing mode.

Usage Guide: To support VPN, VRF has to be enabled on the border routers; to realize VPN, create neighbors for BGP with the VRF address family on the private network, and with VPNv4 address-family on the public network. Configuration performed with this command to specific VRF, is independent from IPv4 unicast address-family. The VRF configuration is performed by using ip vrf <NAME> command under global mode. The address-family configuration is only available after the VRF RD is set.

Example: In the example below a VRF name test is created with RD at 100: :10, and then enter the BGP address-family for its configuration.

```
Switch(config)#ip vrf test
Switch(config-vrf)#rd 100:10
Switch(config-vrf)#exit
Switch(config)#router bgp 100
Switch(config-router)#address-family ipv4 vrf test
Switch(config-router-af)#
```

1.8.3.3 address-family vpnv4

Command: address-family vpnv4

Function: Enter the BGP VPNv4 address family mode.

Parameter: None.

Command Mode: BGP routing mode.

Usage Guide: To support VPN, VRF has to be enabled on the border routers; to realize VPN, create neighbors for BGP with the VRF address family on the private network, and with VPNv4 address-family on the public network. When configuring VPNv4 address-family with this command, IPv4 unicast address connection is available. Its neighbor configuration could be the same with IPv4 unicast only by using neighbor A.B.C.D activate on this neighbor to enable this address-family.

Example:

```
Switch(config)#router bgp 100
Switch(config-router)#address-family vpnv4
Switch(config-router-af)#
```

1.8.3.4 aggregate-address

Command: aggregate-address *<ip-address/M>* [summary-only] [as-set]

no aggregate-address *<ip-address/M>* [summary-only] [as-set]

Function: Configure the aggregate-address.The "**no aggregate-address *<ip-address/M>* [summary-only] [as-set]**" command deletes the aggregate-address.

Parameter: *<ip-address/M>*: IP address, length of mask.

[summary-only]: Send summary only ignoring specific route.

[as-set]: Show AS on the path in list, each AS is shown once.

Default: No aggregate configuration.

Command Mode: BGP routing mode.

Usage Guide: Address aggregation reduces spreading routing messages outside. Use summary-only option so to spread aggregate route to the neighbors without spreading specific route. as-set option will list AS from each route covered by the aggregation only once without repeat.

Example:

```
Switch(config-router)# aggregate-address 100.1.0.0/16 summary-only
Switch(config-router)# aggregate-address 100.2.0.0/16 summary-only as-set
Switch(config-router)# aggregate-address 100.3.0.0/16 as-set
```

1.8.3.5 bgp aggregate-nexthop-check

Command: bgp aggregate-nexthop-check

no bgp aggregate-nexthop-check

Function: Configures whether BGP checks all the route next-hop in aggregating. The "**no bgp aggregate-nexthop-check**" command cancels this configuration, namely not check the next-hop accordance of aggregate route.

Parameter: None.

Default: No nexthop checked during aggregating.

Command Mode: Global mode.

Usage Guide: When check is enabled, the aggregate will not be performed if the next-hop of the covered routes are not in accordance. When checking is disabled, all covered route will be aggregated into the aggregate route.

Example: Switch(config)#bgp aggregate-nexthop-check

1.8.3.6 bgp always-compare-med

Command: bgp always-compare-med

no bgp always-compare-med

Function: Configures If MED comparation is always performed. The "**no bgp always-compare-med**" command cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Normally the BGP compares the MED only when the AS is the same. By using this configuration, MED of routes from different AS source will also be compared.

Example: The AS (200) receives the same route prefix form the two AS (100 and 300) carrying different MED,configure the MED comparison is always performed.

Switch(config-router)#bgp always-compare-med

1.8.3.7 bgp bestpath as-path ignore

Command: bgp bestpath as-path ignore

no bgp bestpath as-path ignore

Function: Set to ignore the AS-PATH length. The "**no bgp bestpath as-path ignore**" command cancels this configuration.

Parameter: None.

Default: Not set.

Command Mode: BGP routing mode.

Usage Guide: Length of AS-PATHwill be compared in BGP pathing, and its length can be ignored by using this configuration.

Example: Set to ignore the AS-PATH length.

Switch(config)#router bgp 200

Switch(config-router)#bgp bestpath as-path ignore

1.8.3.8 bgp bestpath compare-confed-aspath

Command: bgp bestpath compare-confed-aspath
no bgp bestpath compare-confed-aspath

Function: Set to concern the confederation AS-PATH length. The "**no bgp bestpath compare-confed-aspath**" command cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Normally only the length of external AS-PATH will be compared in BGP pathing. By using this configuration, lengths of AS inner confederation AS-PATH will be compared at the same time.

Example: Switch(config-router)#bgp bestpath compare-confed-aspath

1.8.3.9 bgp bestpath compare-routerid

Command: bgp bestpath compare-routerid
no bgp bestpath compare-routerid

Function: Compare route ID; the "**no bgp bestpath compare-routerid**" command cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Normally the first arrived route from the same AS (with other conditions equal) will be chosen as the best route. By using this command, source router ID will also be compared.

Example: Device（10.1.1.66, AS200）receives the same route prefix from two devices（10.1.1.64 and 10.1.1.68）of the same AS (100), configure the device to compare route ID.

Switch(config-router)#bgp bestpath compare-routerid

1.8.3.10 bgp bestpath med

Command: bgp bestpath med {[confed] [missing-as-worst]}
no bgp bestpath med {[confed] [missing-as-worst]}

Function: Configure whether the MED attributes should be compared in the confederation path and the treatment when MED is unavailable. The "**no bgp bestpath med {[confed] [missing-as-worst]}**" command cancels this configuration.

Parameter: [confed]: Compare MED in the confederation path.
[missing-is-worst]: Consider as max MED value when missing.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Choose whether MED is compared among confederations by this command. If MED is missing, it is considered max when missing-is-worst or else 0.

Example: Switch(config-router)#bgp bestpath med confed missing-as-worst

1.8.3.11 bgp client-to-client reflection

Command: bgp client-to-client reflection

no bgp client-to-client reflection

Function: Configures whether the route reflection is performed. The "**no bgp client-to-client reflection**" cancels this configuration.

Parameter: None.

Default: Reflection defaulted when client is configured.

Command Mode: BGP routing mode.

Usage Guide: After configured reflection client with neighbor {<ip-address>|<TAG>} route-reflector-client, the router performs routing reflection in default condition. The NO form of this command cancels the route reflection among CLIENT, (reflection among Clients and non-CLIENT is not disturbed.)

Example: Switch(config-router)#no bgp client-to-client reflection

1.8.3.12 bgp cluster-id

Command: bgp cluster-id {<*ip-address*>|<*01-4294967295*>}

no bgp cluster-id {<[<*ip-address*>]|<*0-4294967295*>}

Function: Configure the route reflection ID during the route reflection. The "**no bgp cluster-id {<[<*ip-address*>]|<*0-4294967295*>}**" command cancels this configuration

Parameter: ***<ip-address>*|*<1-4294967295>***: **>**: cluster-id which is shown in dotted decimal notation or a 32 digit number.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: A cluster consists of one routing reflector and its clients in an area. However in order to increase the redundancy level, sometime more than one routing reflectors may be deployed in one area. Router-id is for identifying the router exclusively in an area, and cluster-id is required for two or more reflector identification.

Example: Switch(config-router)#bgp cluster-id 1.1.1.1

1.8.3.13 bgp confederation identifier

Command: bgp confederation identifier <*as-id*>

no bgp confederation identifier [<*as-id*>]

Function: Create a confederation configuration. The “**no bgp confederation identifier [<*as-id*>]**” command deletes a confederation.

Parameter: ID number of the confederation AS.

Default: No confederation.

Command Mode: BGP routing mode.

Usage Guide: Confederation is for divide large AS into several smaller AS, while still identified as the large AS. Create large AS number with this command.

Example: Switch(config-router)# bgp confederation identifier 600

1.8.3.14 bgp confederation peers

Command: bgp confederation peers <*as-id*> [<*as-id*>..]

no bgp confederation peers <*as-id*> [<*as-id*>..]

Function: Add/delete one or several AS to a confederation.

Parameter: ID numbers of the AS included in the confederation, which could be multiple.

Default: No members.

Command Mode: BGP routing mode.

Usage Guide: Confederation is for dividing large AS into several smaller AS, while still identified as the large AS. Use this command to add/delete confederation members.

Example:

```
Switch(config-router)# bgp confederation identifier 600
Switch(config-router)#bgp confederation peers 100 200
```

1.8.3.15 bgp dampening

Command: bgp dampening [<*1-45*>] [<*1-20000*> <*1-20000*> <*1-255*>] [<*1-45*>]

no bgp dampening

Function: Configure the route dampening. The “**no bgp dampening**” command cancels the route dampening function.

Parameter: **<*1-45*>**: Respectively the penalty half-lives of accessible and unaccessible route, namely the penalty value is reduced to half of the previous value, in minutes.

<*1-20000*>: Respectively the penalty reuse border and restrain border.

<*1-255*>: Maximum restrain route time, in minutes.

Default: Half-life of accessible route is 15 minutes, 15 minutes for unaccessible. The restrain border is 2000, reuse border is 750, and maximum restrain time is 60 minutes.

Command Mode: BGP routing mode.

Usage Guide: Abundant route update due to unstable route could be reduced with route dampening technology, of which the algorithm is lay penalty on the route when the route fluctuates, and when penalty exceeds the restrain border this route will no longer be advertised.

The penalty value will be reduced by time by the half-life index regulation if the route keeps stable and finally be advertised again when the penalty falls below the border or the restrain time exceeds the maximum restrain time. This command is for enabling/disabling the route dampening and configuring its parameters.

Example: Switch(config-router)# bgp dampening

1.8.3.16 bgp default

Command: bgp default {ipv4-unicast|local-preference <*0-4294967295*>}

no bgp default {ipv4-unicast|local-preference [<*0-4294967295*>]}

Function: Set the BGP defaults, the "**no bgp default {ipv4-unicast|local-preference [<*0-4294967295*>]}**" command cancels this configuration.

Parameter: <*0-4294967295*>: Default local priority.

Default: The IPv4 unicast is default enabled when BGP is enabled. The default priority is 100.

Command Mode: BGP routing mode.

Usage Guide: IPv4 unicast address-family is default enabled in BGP. Cancel this setting with no bgp default ipv4-unicast command so to not enable this address-family in default. Default local priority can be configured through bgp default local-preference command.

Example:

Switch(config)#router bgp 200

Switch(config-router)# bgp default local-preference 500

1.8.3.17 bgp deterministic-med

Command: bgp deterministic-med

no bgp deterministic-med

Function: Use the best MED for the same prefix in the AS to compare with other AS. The "**no bgp deterministic-med**" cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Normally if same prefix routes from several paths, each path will be compared. With this configuration, the system will only use the path with the smallest MED in the AS (when other main attributes equal) to compare with other AS. After the best one is elected, select the path among AS with no regard to MED value.

Example: Switch(config-router)#bgp deterministic-med

1.8.3.18 bgp enforce-first-as

Command: bgp enforce-first-as

no bgp enforce-first-as

Function: Enforces the first AS position of the route AS-PATH contain the neighbor AS number or else disconnect this peer when the BGP is reviving the external routes. The "**no bgp enforce-first-as**" command cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: This command is usually for avoiding unsafe or unauthenticated routes.

Example: Switch(config-router)#bgp enforce-first-as

1.8.3.19 bgp fast-external-failover

Command: bgp fast-external-failover

no bgp fast-external-failover

Function: Fast reset when the BGP neighbor connection varies at the interface other than wait for TCP timeout. The "**no bgp fast-external-failover**" command cancels this configuration.

Parameter: None.

Default: Configured.

Command Mode: BGP routing mode.

Usage Guide: This command is for immediately cutting of the neighbor connection when the interface is down.

Example: Switch(config-router)# bgp fast-external-failover

1.8.3.20 bgp inbound-route-filter

Command: bgp inbound-route-filter

no bgp inbound-route-filter

Function: The bgp do not install the RD routing message which does not exist locally. The "**no bgp inbound-route-filter**" command means the RD will be installed with no regard to the local existence of the RD.

Parameter: None.

Command Mode: BGP routing mode.

Usage Guide: Normally when the switch plays as PE, whether the route bgp acquired from VPN is saved in BGP depends on if the VRF configured in this PE has got matched information. With the "**no bgp inbound-route-filter**" command the BGP will save the routing message with no regard to the matched information.

Example:

```
Switch(config)#router bgp 100
Switch(config-router)#no bgp inbound-route-filter
```

1.8.3.21 bgp inbound-max-route-num

Command：bgp inbound-max-route-num <0-500000>

no bgp inbound-max-route-num

Function：Set the number limit of routers learnt by the bgp process from its neighbors.

Parameters：The number limit of routers, ranging from 0 to 500000.

Default：The number limit is 50000 by default.

Command Mode：BGP routing mode and address family mode.

Usage Guide：Limit the number of routers learnt by the bgp process from its neighbors with this command.

Example：The following configuration will limit max number of routers that the bgp process receives from its neighbors as 20000.

```
Switch(config-router)# bgp inbound-max-route-num 20000
```

1.8.3.22 bgp log-neighbor-changes

Command: bgp log-neighbor-changes

no bgp log-neighbor-changes

Function: Output log message when BGP neighbor changes. The "**no bgp log-neighbor-changes**" command cancels this configuration.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Can display neighbor change messages on the monitor.

Example: Switch(config-router)# bgp log-neighbor-changes

1.8.3.23 bgp network import-check

Command: bgp network import-check

no bgp network import-check

Function: Set whether check the IGP accessibility of the BGP network route or not. The "**no bgp network import-check**" command sets to not checking the IGP accessibility.

Parameter: None.

Default: Not configured.

Command Mode: BGP routing mode.

Usage Guide: Checking the IGP accessibility of the route advertised by BGP is to check the existence of next-hop and its IGP accessibility.

Example: Switch(config-router)# bgp network import-check

1.8.3.24 bgp RFC1771-path-select

Command:bgp RFC1771-path-select

no bgp RFC1771-path-select

Parameter: None.

Default: Not following.

Command Mode: Global mode.

Usage Guide: After this attribute is set, path selecting will follow the way defined in RFC 1771, namely not checking the AS internal metric when different AS exist, which should be perform without this attribute set.

Example:

Switch(config)# bgp RFC1771-path-select

1.8.3.25 bgp RFC1771-strict

Command: bgp RFC1771-strict

no bgp RFC1771-strict

Function: Set whether strictly follows the RFC1771 restrictions. The "**no bgp RFC1771-strict**" command set to not strictly following.

Parameter: None.

Default: Not following RFC 1771 restrictions.

Command Mode: Global mode.

Usage Guide: With this attribute set, generation types of routes from protocols such as RIP, OSPF, ISIS, etc will be regarded as IGP (internal generated), or else as incomplete. **Example:**

Switch(config)# bgp RFC1771-strict

Switch(config)# no bgp RFC1771-strict

1.8.3.26 bgp router-id

Command: bgp router-id <*ip-address*>

no bgp router-id [<*ip-address*>]

Function: Configure the router ID manually. The "**no bgp router-id [<*ip-address*>]**" cancels this configuration.

Parameter: <*ip-address*>: Router ID.

Default: Automatically acquire router ID.

Command Mode: BGP routing mode.

Usage Guide: Manually set the router ID with this command.

Example: Switch(config-router)# bgp router-id 1.1.1.1

1.8.3.27 bgp scan-time

Command:bgp scan-time <*0-60*>

no bgp scan-time [<*0-60*>]

Function: Set the time interval of the periodical next-hop validation; the "**no bgp scan-time [<*0-60*>]**" command restores to the default value.

Parameter: <*0-60*>: Validation time interval.

Default: Default interval is 60s.

Command Mode: BGP routing mode.

Usage Guide: Validate the next-hop of BGP route, this command is for configuring the interval of this check. Set the parameter to 0 if you don't want to check.

Example: Switch(config-router)# bgp scan-time 30

1.8.3.28 clear ip bgp

Command: clear ip bgp [view <*NAME*>] {<*>|<*as-id*>| external|peer-group <*NAME*>|<*ip-address*>} [<*ADDRESS-FAMILY*>] [in [prefix-filter] |out|soft [in|out]]

Function: Clear up BGP links or states.

Parameter: all.

<*as-id*>: AS number.

<*NAME*>: Respectively BGP instance name and peer group name.

<*ip-address*>: IP address.

<*ADDRESS-FAMILY*>: Address family, such as "ipv4 unicast".

Default: None.

Command Mode: Admin mode.

Usage Guide: Clearing up BGP state in different parameters (such as AS number, peer group name, IPv4 address, address-family, external neighbor), or the inbound or outbound messages. Also it is optional to use the saved ORF as soft reconfiguration, or use the soft in|out command for in or out soft reconfiguration if it is already set.

Example: When soft reconfiguration is set, use this command for soft reconfiguration.

Switch# clear ip bgp * soft in

Will clear up all established connections.

Switch# clear ip bgp *

1.8.3.29 clear ip bgp dampening

Command:clear ip bgp [<ADDRESS-FAMILY>] dampening [<ip-address>|<ip-address/M>]

Function: Used for resetting BGP routing dampening.

Parameter: <*ADDRESS-FAMILY*>: address-family, such as "ipv4 unicast".

<*ip-address*>: IP address.

<*ip-address/M*>: IP address and mask.

Default: None.

Command Mode: Admin mode.

Usage Guide: It is possible to clear BGP routing dampening messages and state by different parameters (such as address-family or IPv4 address).

Example: Switch# clear ip bgp ipv4 unicast dampening

1.8.3.30 clear ip bgp flap-statistics

Command:clear ip bgp [<*ADDRESS-FAMILY*>] flap-statistics [<*ip-address*>|<*ip-address/M*>]

Function: For resetting BGP routing dampening statistics messages.

Parameter: **<*ADDRESS-FAMILY*>**: address-family such as "ipv4 unicast".

<*ip-address/M*>: IP address and mask.

Default: None.

Command Mode: Admin mode.

Usage Guide: It is possible to clear BGP routing dampening statistic messages and state by different parameters (such as address-family or IPv4 address).

Example: Switch#clear ip bgp ipv4 unicast flap-statistics

1.8.3.31 distance

Command: distance <*1-255*> <*ip-address/M*> [<*WORD*>]

no distance <*1-255*> <*ip-address/M*> [<*WORD*>]

Function: Set the manage distance of the routing prefix. The "**no distance <*1-255*> <*ip-address/M*> [<*WORD*>]**" command restores to the default value.

Parameter: **<*1-255*>**: Manage distance.

<*ip-address/M*>: Routing prefix.

<*WORD*>: Access-list name.

Default: Not set.

Command Mode: BGP routing mode.

Usage Guide: Set the manage distance for specified BGP route as the path selecting basis.

Example: Switch(config-router)# distance 90 10.1.1.64/32

1.8.3.32 distance bgp

Command: distance bgp <*1-255*> <*1-255*> <*1-255*>

no distance bgp [<*1-255*> <*1-255*> <*1-255*>]

Function: Set the BGP protocol manage distance. The "**no distance bgp [<*1-255*> <*1-255*> <*1-255*>]**" command restores the manage distance to default value.

Parameter: Respectively the EBGP, IBGP and LOCAL manage distance of the BGP.

Default: Default EBGP is 20, others are 200.

Command Mode: BGP routing mode.

Usage Guide: Set the manage distance for BGP routing as the NSM path selecting basis.

Example: Switch(config-router)# distance bgp 15 150 150

1.8.3.33 exit-address-family

Command: exit-address-family

Function: Exit the BGP address-family mode.

Parameter: None.

Default: None.

Command Mode: BGP address-family mode.

Usage Guide: Use this command to exit the mode so to end the address-family configuration when configuring address-family under BGP.

Example:

```
Switch(config)#router bgp 100
Switch(config-router)#address-family ipv4 unicast
Switch(config-router-af)# exit-address-family
Switch(config-router)#
```

1.8.3.34 import map

Command: import map <*map-name*>

no import map <*map-name*>

Function: Use this command to configure the route-map regulations when introducing routes into VRF.

Parameter: <*map-name*> is the route-map name used.

Command Mode: vrf mode.

Usage Guide: Use the route map command route-map NAME permit|deny <1-65535> to create the route-map and establish the regulations. Using this command will apply regulations to the route introducing of this VRF.

Example:

```
Switch(config)#route-map map1 permit 15
Switch(config-map)#match interface Vlan1
Switch(config-map)#set weight 655
Reconfiguring VRF test with this route-map
Switch(config-map)#exit
Switch(config)#ip vrf test
Switch(config-af)#rd 100:10
Switch(config-af)#route-target both 100:10
Switch(config-af)#import map map1
Switch#show ip bgp vpn all
```

```
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:10 (Default for VRF test)
*> 11.1.1.0/24      11.1.1.64                0             0 200 ?
*>i15.1.1.0/24      10.1.1.68                0    100    655 300 ?
*> 20.1.1.0/24      11.1.1.64                0             0 200 ?
*>i100.1.1.0/24     10.1.1.68                0    100    655 300 ?
Route Distinguisher: 100:10
*>i15.1.1.0/24      10.1.1.68                0    100      0 300 ?
*>i100.1.1.0/24     10.1.1.68                0    100      0 300 ?
```

As we can see, the weight of the route from the VPN changes to 655 after introduced into VRF test.

1.8.3.35 ip as-path access-list

Command: ip as-path access-list <*.LINE*> {<*permit*>|<*deny*>} <*LINE*>

no ip as-path access-list <*.LINE*> {<*permit*>|<*deny*>} <*LINE*>

Function: Configure the AS-PATH access-list. The "**no ip as-path access-list <*.LINE*> {<*permit*>|<*deny*>} <*LINE*>**" command deletes this access-list.

Parameter: <*.LINE*>: name of access-list.

<*LINE*>: matched strings in the AS-PATH.

Default: None.

Command Mode: Global mode.

Usage Guide: Use this command to configure the access-list related to AS-PATH, so to supply the conditions for pass/filter.

Example: Switch(config)#ip as-path access-list ASPF deny ^100$

1.8.3.36 ip community-list

Command: ip community-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>] | [standard <*WORD*>]} {deny|permit} <*.COMMUNITY*>

no ip community-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>] | [standard <*WORD*>]} [{deny|permit} <*.COMMUNITY*>]

Function: Configure the community-list. The "**no ip community-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>]|[standard <*WORD*>]} [{deny|permit} <*.COMMUNITY*>]**" command deletes the community list.

Parameter: <*LISTNAME*>: name of community list.

<*1-199*>: Standard or extended community number.

<*WORD*>: Standard or extended community number.

<*.COMMUNITY* >: Members of the community list, which may be the combination of aa:nn, or internet, local-AS, no-advertise, and no-export. It can be shown in regular expressions under extended conditions.

Default: None.

Command Mode: Global mode.

Usage Guide: With this command we can configure the community-list so to supply terms for the pass/filter/search.

Example: Switch(config)# ip community-list LN permit 100:10

1.8.3.37 ip extcommunity-list

Command: ip extcommunity-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>]|[standard <*WORD*>]} {deny|permit} <*.COMMUNITY*>

no ip extcommunity-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>]|[standard <*WORD*>]} {deny|permit} <*.COMMUNITY*>

Function: Configure the extended community-list. The "**no ip extcommunity-list {<*LISTNAME*>|<*1-199*>|[expanded <*WORD*>]|[standard <*WORD*>]} {deny|permit} <*.COMMUNITY*>**" command is for deleting the extended community list.

Parameter: <*LISTNAME*>: name of community-list.

<*1-199*>: Standard or extended community number.

<*WORD*>: Standard or extended community number.

<*.COMMUNITY* >: Members of the community list, which may be the combination of aa:nn, or internet, local-AS, no-advertise, and no-export. It can be shown in regular expressions under extended conditions.

Default: None.

Command Mode: Global mode.

Usage Guide: With this command we can configure the community-list so to supply terms for the pass/filter/search.

Example: Switch(config)# ip extcommunity-list LN permit 100:10

1.8.3.38 neighbor activate

Command: neighbor {<*ip-address*>|<*TAG*>} activate

no neighbor {<*ip-address*>|<*TAG*>} activate

Function: Configure the address family routing which do or do not switch specific address-family with BGP neighbors. The "**no neighbor {<*ip-address*>|<*TAG*>} activate**" command is for setting the route which do not switch the specified address family.

Parameter: <*ip-address*>: IP address of the neighbor.

<*TAG*>: Name of peer group.

Default: Enable the routing switch of IP unicast address-family, and disable other address-families.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: IP unicast is configured under BGP routing mode. Configure whether specific address-family is switched under address-family mode. If this option on any side between local side and partner is not enabled, the address-family route will not be acquired by the partner even if the corresponding address family routes acquired before will be cancelled after this option is disabled.

Example:

```
Switch(config-router)#neighbor 10.1.1.64 activate
Switch(config-router)#address-family ipv4
Switch(config-router-af)#no neighbor 10.1.1.64 activate
Switch(config-router-af)#
```

1.8.3.39 neighbor advertisement-interval

Command: neighbor {<*ip-address*>|<*TAG*>} advertisement-interval <*0-600*>

no neighbor {<*ip-address*>|<*TAG*>} advertisement-interval [<*0-600*>]

Function: Configure the update interval of specific neighbor route. the "**no neighbor {<*ip-address*>|<*TAG*>} advertisement-interval [<*0-600*>]**" command restores to default.

Parameter: <*ip-address*>: IP address of the neighbor.

<*TAG*>: Name of the peer group.

<*0-600*>: Advertise interval, in seconds.

Default: Default interval for IBGP is 5s, for EBGP is 30s.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Reduce this value will improve the route updating speed while also consumes more bandwidth.

Example:

```
Switch(config-router)#neighbor 10.1.1.64 advertisement-interval 20
Switch(config-router)#no neighbor 10.1.1.64 advertisement-interval
Switch(config-router)#
```

1.8.3.40 neighbor allowas-in

Command: neighbor {<*ip-address*>|<*TAG*>} allowas-in [<*1-10*>]

no neighbor {<*ip-address*>|<*TAG*>} allowas-in

Function: Configure the counts same AS is allowed to appear in the neighbor route AS table. The "**no neighbor {<*ip-address*>|<*TAG*>} allowas-in**" restores to not allow any repeat.

Parameter: <*ip-address*>: IP address of the neighbor.

<*TAG*>: Name of the peer group.

<*1-10*>: Allowed count of same AS number.

Default: In default conditions AS is not allowed repeating in the same route, and when set the repeat count it is defaulted at 3 when <1-10> parameters not set.

Command Mode: BGP routing mode and address family mode.

Usage Guide: Normally BGP will not allow same AS number appears in the route more than one time. The system will deny a route when its AS number appears in the AS-PATH. However to support some special needs, especially the VPN support, the extended BGP allows the AS re-appear counts by configuration. This command is for configure the re-appear counts.

Example: Switch(config-router)#neighbor 10.1.1.66 allowas-in

1.8.3.41 neighbor attribute-unchanged

Command: neighbor {<*ip-address*>|<*TAG*>} attribute-unchanged [as-path][med][next-hop]

no neighbor {<*ip-address*>|<*TAG*>} attribute-unchanged [as-path] [med] [next-hop]

Function: Configure certain attributes which is kept unchanged for transmitting, namely the attribute transparent transmission. The "***no* neighbor {<*ip-address*>|<*TAG*>} attribute-unchanged [as-path] [med] [next-hop]**" command means the attribute transparent transmission is not performed.

Parameter: <*ip-address*>:IP address of the neighbor.

<*TAG*>: Name of the peer group.

Default: No attribute transparent defined.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: With this configuration specified route attributes will not change when transmitted to the specified neighbor. The BGP routing mode is the IPv4 unicast configuration. No parameter refers to above three parameter are configured together.

Example:

Switch(config-router)#neighbor 10.1.1.64 attribute-unchanged

Switch(config-router)# no neighbor 10.1.1.64 attribute-unchanged as-path med

1.8.3.42 neighbor capability

Command: neighbor {<*ip-address*>|<*TAG*>} capability {dynamic | route-refresh}

no neighbor {<*ip-address*>|<*TAG*>} capability {dynamic | route-refresh}

Function: Configure dynamic update between neighbors and the route refresh capability negotiation. The "**no neighbor {<*ip-address*>|<*TAG*>} capability {dynamic | route-refresh}**" command do not enable the specific capability negotiation.

Parameter: <ip-address>:Neighbor IP address.

<*TAG*>: Name of peer group.

Default: Not configure the dynamic update capability but the route refresh capability.

Command Mode: BGP routing mode and address family mode.

Usage Guide: This is an extended BGP capability. With this configuration supported capabilities by both side will be negotiated in the OPEN messages, and the partner will respond if this capability is supported by the partner and send NOTIFICATION if not. The originating side will then send an OPEN excluded the capability to reestablish the connection. The dynamic capability refers to when the address family negotiation changes, the connection don't have to be restarted. Route refresh refers to sending refresh request when configuring some soft reconfigurable attributes and the partner will retransmit the existing route to the originating side. With route refresh attribute, the connection will not have to be restarted but be refreshed with the clear ip bgp * soft in command.

Example:

Switch(config-router)#neighbor 10.1.1.64 capability dynamic

Switch(config-router)# no neighbor 10.1.1.64 capability route-refresh

1.8.3.43 neighbor capability orf prefix-list

Command: neighbor {<*ip-address*>|<*TAG*>} capability orf prefix-list {<*both*>|<*send*>|<*receive*>}

no neighbor {<*ip-address*>|<*TAG*>} capability orf prefix-list {<*both*>|<*send*>|<*receive*>}

Function: Configure the out route filter capability negotiation between neighbors. The "**no neighbor {<*ip-address*>|<*TAG*>} capability orf prefix-list {<*both*>|<*send*>|<*receive*>}**" command set to not perform the negotiation.

Parameter: <*ip-address*>:Neighbor IP address.

<*TAG*>:Name of peer group.

Default: ORF capability not configured.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: This is an extended BGP capability. With this configuration supported capabilities by both side will be negotiated in the OPEN messages, and the partner will respond if this capability is supported by the partner and send NOTIFICATION if not. The originating side will then send an OPEN excluded the capability to reestablish the connection. With this capability, the side configured with in prefix-list filter rules will transmit its own filter rules to the peer, the peer group will apply this rule as its own out rules, so to avoid sending route which will be denied by the partner.

Example: Switch(config-router)#neighbor 10.1.1.66 capability orf prefix-list both

1.8.3.44 neighbor collide-established

Command:neighbor {<*ip-address*>|<*TAG*>} collide-established

no neighbor {<*ip-address*>|<*TAG*>} collide-established

Function: Enable the collision check and settlement in the TCP connection collision. The "**no neighbor {<*ip-address*>|<*TAG*>} collide-established**" command disables the TCP connection collision settlement.

Parameter:*<ip-address>*: Neighbor IP address.

<TAG>: Name of the peer.

Default: Disabled and Unavailable.

Command Mode: route mode and address family mode.

Usage Guide: This command is for settling the problem that multi-connection among peers due to TCP connection collision. Connections created with this option on will always be check even at established state. And it will be checked if local side IP is larger than partner IP when collides. If yes, the original connection will be deleted, and if not the option will be configured to only checks when the connection originated from local side at open sent and open confirm state.

Example: Switch(config-router)#neighbor 10.1.1.64 collide-established

1.8.3.45 neighbor default-originate

Command: neighbor {<*ip-address*>|<*TAG*>} default-originate [route-map <*WORD*>]

no neighbor {<*ip-address*>|<*TAG*>} default-originate [route-map <*WORD*>]

Function: Configures whether enables transmitting default route to the specific neighbor. The "**no neighbor {<*ip-address*>|<*TAG*>} default-originate [route-map <*WORD*>]**" command configures not sending default route to neighbors.

Parameter:*<ip-address>*: IP address of the neighbor.

<TAG>: Name of the peer.

<WORD> :Name of route map.

Default: Not sending default route.

Command Mode:BGP routing mode and address-family mode.

Usage Guide: With this option, the default route of local side will be transmitted to partner, or else not. It supplies with options of which one to supply the default route. if several neighbors of the partner supply default route, the best one will be elected according to path selecting principles. According to route mirror, it can be chosen when to send the default route.

Example: Set to transmit the local default route to neighbor 10.1.1.64.

Switch(config-router)#neighbor 10.1.1.64 default-originate

1.8.3.46 neighbor description

Command: neighbor {<*ip-address*>|<*TAG*>} description <*.LINE*>

no neighbor {<*ip-address*>|<*TAG*>} description

Function: Configure the description string of the peer or peer group. The "**no neighbor {<*ip-address*>|<*TAG*>} description**" command deletes the configurations of this string.

Parameter:*<ip-address>*: Neighbor IP address.

<TAG>: Name of peer group.

<.LINE>: Description string consists of displayable characters less than 80.

Default: Description string is empty.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Configure the introduction of the peer or peer group.

Example:

```
Switch(config-router)#neighbor 10.1.1.64 description tester
Switch(config-router)#
```

1.8.3.47 neighbor distribute-list

Command: neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}

no neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}

Function: Configure the policy applied in partner route update transmission. The "**no neighbor {<*ip-address*>|<*TAG*>} distribute-list {<*1-199*>|<*1300-2699*>|<*WORD*>} {in|out}**" command cancels the policy configuration.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>:Name of peer group.

<*1-199*>|<*1300-2699*>|<*WORD*>: Number or name of the access-list.

Default: Policy not applied.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Configure the policies with access-list command and apply this command on route sending and receiving. It will filter the update route from partner when use in mode, and will filter the route from local side to partner with out mode.

Example:

Configure the access-list

```
Switch(config)#access-list 101 deny ip 100.1.0.0 0.0.1.255 any
Switch(config)#access-list 101 permit ip any any
Switch(config)#router bgp 100
Switch(config-router)# neighbor 10.1.1.66 distribute-list 101 out
```

1.8.3.48 neighbor dont-capability-negotiate

Command: neighbor {<*ip-address*>|<*TAG*>} dont-capability-negotiate

no neighbor {<*ip-address*>|<*TAG*>} dont-capability-negotiate

Function: Set to not perform capability negotiate in creating connections. The "**no neighbor {<*ip-address*>|<*TAG*>} dont-capability-negotiate**" command cancels this configuration.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of the peer group.

Default: Capability negotiation performed.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: As the negotiation is the default, it can be disabled with this configuration when it is known that the partner BGP version is old which don't support capability negotiation.

Example: Last addition capability negotiation will not be realized in the connection by configuring as follows.

Switch(config-router)#neighbor 10.1.1.64 dont-capability-negotiate

1.8.3.49 neighbor ebgp-multihop

Command: neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>]

no neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>]

Function: Configures the EBGP neighbors can existing in different segment as well as its hop count (TTL). The "**no neighbor {<*ip-address*>|<*TAG*>} ebgp-multihop [<*1-255*>]**" set that the EBGP neighbors must be in the same segment.

Parameter:<*ip-address*>: Neighbor IP address.

<*TAG*>: Name of the peer group.

<*1-255*>: Allowed hop count.

Default: Must be in the same segment.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Without this command, EBGP peers are required to be in the same segment and after this command is configured, peer addresses may from different segments. The allowed hop count can be configured and will be 255 if not.

Example:

Three device 10.1.1.64(AS100) and 11.1.1.120(AS300) connected respectively to the two interface 10.1.1.66 and 10.1.1.100 of another device. IGP accessibilities of 10.1.1.64 and 11.1.1.120 on both side routes are ensured through static configuration. The neighbor relationship is established only after both side are configured as follows:

on 10.1.1.64

Switch(config-router)#neighbor 11.1.1.120 ebgp-multihop

on 11.1.1.120

Switch(config-router)#neighbor 10.1.1.64 ebgp-multihop

After this, switches in different segments will be able to create BGP neighbor relationship

1.8.3.50 neighbor enforce-multihop

Command: neighbor {<*ip-address*>|<*TAG*>} enforce-multihop

no neighbor {<*ip-address*>|<*TAG*>} enforce-multihop

Function: Enforce the multihop connection to the neighbor. The "**no neighbor {<*ip-address*>|<*TAG*>} enforce-multihop**" command cancels this configuration.

Parameter: <*ip-address*>: Neighbor IP address.

<TAG>: Name of peer group.

Default: Not enforced.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: In fact the direct route can not be enforced to multihop, however will be treated as a multihop connection with this configuration, namely the check originally only performed on IBGP and EBGP of non-direct routes will be performed on all after this attribute set. The nexthop direct connected check will not be performed at exit in enforce multihop conditions.

Example: Switch(config-router)#neighbor 10.1.1.66 enforce-multihop

1.8.3.51 neighbor filter-list

Command: neighbor {<*ip-address*>|<*TAG*>} filter-list <*.LINE*> {<*in*>|<*out*>}

no neighbor {<*ip-address*>|<*TAG*>} filter-list <*.LINE*> {<*in*>|<*out*>}

Function: Access-list control for AS-PATH. The "**no neighbor {<*ip-address*>|<*TAG*>} filter-list <*.LINE*> {<*in*>|<*out*>}**" cancels the AS-PATH access-list control.

Parameter: <*ip-address*>: Neighbor IP address.

<TAG>: Name of peer group.

<.LINE>:AS-PATH access-list name configured through ip as-path access-list <*.LINE*> <*permit|deny*> <*LINE*>*.*

Default: Not configured.

Command Mode:BGP routing mode and address list mode.

Usage Guide: After first configured the IP AS-PATH access-list, apply this option to specified neighbor will be able to send/receive routes with specified AS numbers in the AS list. Accepting or denying depends on the configuration of the access-list, while sending and receiving are configured by this command.

Example:

Configure the AS-PATH access control list, "ASPF" is the name of the access-list. The route with AS number of 100 will not be able to update to the partner due to the filter table control.

```
Switch(config)#ip as-path access-list ASPF deny 100
Switch(config)#router bgp 100
Switch(config-router)# redistribute static
Switch(config-router)neighbor 10.1.1.66 filter-list aspf out
```

1.8.3.52 neighbor interface

Command: neighbor <*ip-address*> interface <*IFNAM*>

no neighbor <*ip-address*> interface <*IFNAM*>

Function: Specify the interface to the neighbor. The "**no neighbor <*ip-address*> interface <*IFNAM*>**" of the command cancels this configuration.

Parameter: <*ip-address*>: Neighbor IP address.

***<IFNAME>*:** Interface name, e.g. “Vlan 2”.

Default: Not configured.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Specifies the exit interface to the neighbor with this command. Interface destination accessibility should be ensured.

Example: Switch(config-router)# neighbor 10.1.1.64 interface Vlan2

1.8.3.53 neighbor maximum-prefix

Command: neighbor {<*ip-address*>|<*TAG*>} maximum-prefix <*1-4294967295*> [<*1-100*> <*warning-only*>]

no neighbor {<*ip-address*>|<*TAG*>} maximum-prefix <*1-4294967295*> [<*1-100*> <*warning-only*>]

Function: Control the number of route prefix from the neighbor. The “**no neighbor {<*ip-address*>|<*TAG*>} maximum-prefix <*1-4294967295*> [<*1-100*> <*warning-only*>]**” command cancels this configuration.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of the peer.

<*1-4294967295*>: Max prefix value allowed.

<*1-100*>: Percentage of the max value at which it warns.

<*warning-only*>: Warning only or not.

Default: Not limited.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Due to concerns of too much route updates from neighbors (e.g. attack), the max number of prefix acquired from a neighbor is limited, and will warns when the number hits certain rate. If the warning-only option is set, then there will be warning only, if not, the connection to the neighbor will be cut till clear the records with clear ip bgp command.

Example:

Switch(config-router)#neighbor 10.1.1.64 maximum-prefix 12 50

In above configuration, it warns when the number of route prefix reaches 6, and the connection will be cut when the number hit 13.

1.8.3.54 neighbor next-hop-self

Command: neighbor {<*ip-address*>|<*TAG*>} next-hop-self

no neighbor {<*ip-address*>|<*TAG*>} next-hop-self

Function: Ask the neighbor to point the route nexthop sent by the local side to local side. The “**no neighbor {<*ip-address*>|<*TAG*>} next-hop-self**” command cancels this configuration.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of peer group.

Default: Not configured by default.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: In the EBGP environment, the nexthop will automatically point to the source neighbor. However in IBGP environment, the nexthop remains the same for route in the same segment. If it is not broadcast network, errors will be encountered. This command is for force self as the nexthop of the neighbor under IBGP.

Example: Switch(config-router)#neighbor 10.1.1.66 next-hop-self

1.8.3.55 neighbor override-capability

Command: neighbor {<*ip-address*>|<*TAG*>} override-capability

no neighbor {<*ip-address*>|<*TAG*>} override-capability

Function: Whether enable overriding capability negotiation. The "**no neighbor {<*ip-address*>|<*TAG*>} override-capability**" command restores the capability negotiation.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of the peer group.

Default: Disabled.

Command Mode: BGP route mode.

Usage Guide: With this attribute, error notify due to unsupported capability negotiation the neighbors required will not be sent.

Example: Switch(config-router)#neighbor 10.1.1.64 override-capability

1.8.3.56 neighbor passive

Command: neighbor {<*ip-address*>|<*TAG*>} passive

no neighbor {<*ip-address*>|<*TAG*>} passive

Function: Configure whether the connecting request is positively sent in the connection with specified neighbor; the "**no neighbor {<*ip-address*>|<*TAG*>} passive**" command restores to positively send the connecting request.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of peer group.

Default: Positively send the connecting request.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: With this attribute set, the local side will not positively send the TCP connecting request after the neighbors are configured, but stays in listening mode waiting for the connecting request from partners.

Example:

Switch(config-router)#neighbor 10.1.1.64 passive

After configured with this attribute and reestablishing the connection , the local side do not attempt to create connection but stays in ACTIVE state waiting for the TCP connection request from the partner.

1.8.3.57 neighbor peer-group （Creating）

Command: neighbor < *TAG*> peer-group

no neighbor < *TAG*> peer-group

Function: Create/delete a peer group. The "**no neighbor < *TAG*> peer-group**" command deletes a peer group.

Parameter:<*TAG*>: Name of the peer group of which the largest length contains 256 characters.

Default: No peer group.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: By configuring the peer group, a group of peers with the same attributes will be configured at the same time so to reduce the configuration staff labor. Assign members to the peer group with neighbor <ip-address> peer-group <TAG> command.

Example:

```
Switch(config-router)#neighbor pg peer-group
Switch(config-router)#neighbor 10.1.1.64 peer-group pg
Switch(config-router)#neighbor pg remote-as 100
```

1.8.3.58 neighbor peer-group （Configuring group members）

Command: neighbor <*ip-address*> peer-group <*TAG*>

no neighbor <*ip-address*> peer-group <*TAG*>

Function: Assign/delete peers in the group. The "**no neighbor <*ip-address*> peer-group <*TAG*>**" command deletes the peers from the peer group.

Parameter: <*ip-address*>: Neighbor IP address.

<*TAG*>: Name of peer group.

Default: No peer group.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: By configuring the peer group, a group of peers with the same attributes will be configured at the same time so to reduce the configuration staff labor. Create peer group with above command and assign members into the group with this command.

Example: Refer to above examples.

1.8.3.59 neighbor port

Command: neighbor <*ip-address*> port <*0-65535*>

no neighbor <*ip-address*> port [<*0-65535*>]

Function: Specify the TCP port number of the partner through which the communication carries.

The "**no neighbor *<ip-address>* port [*<0-65535>*]**" command restore the port number to default value.

Parameter: *<ip-address>*: Neighbor IP address.

<TAG>: Name of the peer group.

<0-65535>: TCP port number.

Default: Default port number is 179.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: This is a configuration when the partner may connect through ports not specified by BGP.

Example: Switch(config-router)#neighbor 10.1.1.64 port 1023

1.8.3.60 neighbor prefix-list

Command: neighbor {*<ip-address>*|*<TAG>*} prefix-list *<LISTNAME|number>* {*<in|out>*}

no neighbor {*<ip-address>*|*<TAG>*} prefix-list *<LISTNAME|number>* {*<in>*|*<out>*}

Function: Configure the prefix restrictions applied in sending or receiving routes from specified neighbors. The "**no neighbor {*<ip-address>*|*<TAG>*} prefix-list *<LISTNAME|number>* {*<in>*|*<out>*}**" command cancels this configuration.

Parameter: *<ip-address>*:Neighbor IP address.

<TAG>:Name of the peer group.

<LISTNAME|number>:Name or sequence number of the prefix-list.

<in|out>:Direction on which the restrictions applied.

Default: No prefix restrictions applied.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Specify the prefix and its scope by configuring ip prefix-list and determines whether this scope is permitted or denied. Only the route with permitted prefix will be sent or received.

Example:

```
Switch(config)#ip prefix-list prw permit 100.1.0.0/22 ge 23 le 25
Switch(config)#router bgp 200
Switch(config-router)#redistribute static
Switch(config-router)neighbor 10.1.1.66 prefix-list prw out
```

1.8.3.61 neighbor remote-as

Command: neighbor {*<ip-address>*|*<TAG>*} remote-as *<as-id>*

no neighbor {*<ip-address>*|*<TAG>*} [remote-as *<as-id>*]

Function: Configure the BGP neighbor. The "**no neighbor {*<ip-address>*|*<TAG>*} [remote-as *<as-id>*]**" command is used for deleting BGP neighbors.

Parameter:*<ip-address>*: Neighbor IP address.

<TAG>: Name of peer group.

<as-id>: Neighbor AS number ranging between 1-65535.

Default: No neighbors.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: The BGP neighbors are completely generated through command configurations. A neighbor relationship can only be really established by mutual configuring. Partner AS number should be specified in configuration. The neighbor relationship can not be established when the AS number is incorrect. The partner AS number is the same with that of local side inside the AS.

Example:

Switch(config)#router bgp 200

Switch(config-router)# neighbor 10.1.1.64 remote-as 100

1.8.3.62 neighbor remove-private-AS

Command: neighbor {<*ip-address*>|<*TAG*>} remove-private-AS

no neighbor {<*ip-address*>|<*TAG*>} remove-private-AS

Function: Configures whether remove the private AS number when sending to the neighbor.The "**no neighbor {<*ip-address*>|<*TAG*>} remove-private-AS**" command cancels this configuration.

Parameter: *<ip-address>*: Neighbor IP address.

<TAG>: Name of peer group.

Default: Not configured.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: Configure this attribute to avoid assigning the internal AS number to the external AS sometimes. The internal AS number ranges between 64512-65535, which the AS number could not be sent to the INTERNET since it is not a valid external AS number. What removed here is private AS numbers of the totally private AS routes. Those who have private AS numbers while also have public AS numbers are not processed.

Example: Switch(config-router)#neighbor 10.1.1.64 remove-private-AS

1.8.3.63 neighbor route-map

Command: neighbor {<*ip-address*>|<*TAG*>} route-map <*NAME*> {<*in*|*out*>}

no neighbor {<*ip-address*>|<*TAG*>} route-map <*NAME*> {<*in*|*out*>}

Function: Configure the route mapping policy when sending or receiving route. the "**no neighbor {<*ip-address*>|<*TAG*>} route-map <*NAME*> {<*in*|*out*>}**" command cancels this configuration.

Parameter: *<ip-address>*:Neighbor IP address.

<TAG>: Name of peer group.

<NAME>: Name of route mapping.

<in|out>: Direction of route mapping.

Default: Not set.

Command Mode: BGP routing mode and address-family mode.

Usage Guide: First it has to configure route mapping under global mode by creating a route map with route-map command and configure the match condition and actions, then the command can be applied.

Example:

```
Switch(config)#route-map test permit 5
Switch(config-route-map)#match interface Vlan1
Switch(config-route-map)#set as-path prepend 65532
Switch(config-route-map)#exit
Switch(config)#router bgp 200
Switch(config-router)#neighbor 10.1.1.64 route-map test out
```

1.8.3.64 neighbor route-reflector-client

Command: neighbor {<*ip-address*>|<*TAG*>} route-reflector-client

no neighbor {<*ip-address*>|<*TAG*>} route-reflector-client

Function: Configure the route reflector client. The "**no neighbor {<*ip-address*>|<*TAG*>} route-reflector-client**" command cancels this configuration

Parameter: *<ip-address>*: Neighbor IP address

<TAG>: Name of peer group

Default: Not configured

Command Mode: BGP routing mode and address-family mode.

Usage Guide: The route reflection is used for reducing the peers when the internal IBGP routers inside AS are too much. The client only exchanges messages with route reflector while the reflector deals with message exchange among each client and other IBGP, EBGP routers. This command configures itself as the route reflector, while specific peer group is as its client. Note: this configuration is only available inside AS

Example:

```
Switch(config)#router bgp 100
Switch(config-router)#neighbor 10.1.1.66 remote 100
Switch(config-router)#neighbor 10.1.1.66 route-reflector-client
Switch(config-router)#neighbor 10.1.1.68 remote 100
Switch(config-router)#neighbor 10.1.1.68 route-reflector-client
Switch(config-router)#
```

1.8.3.65 neighbor route-server-client

Command: neighbor {<*ip-address*>|<*TAG*>} route-server-client

no neighbor {<*ip-address*>|<*TAG*>} route-server-client

Function: Configure the route server client. The "**no neighbor {<*ip-address*>|<*TAG*>} route-server-client**" command cancels this configuration

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of peer group

Default: Not configured

Command Mode: BGP routing mode and address-family mode.

Usage Guide: The route service is for reducing the peers when the router between AS is too much under EBGP environment. The server transparently transforms the routing messages to other clients with its client exchanges messages through route server

Example:

Three routers : 10.1.1.64（AS100） and 10.1.1.68（AS300） respectively creates neighbor relationship with the connected 10.1.1.66（AS200）, the configuration is as follows

```
Switch(config)#router bgp 200
Switch(config-router)#neighbor 10.1.1.64 remote-as 100
Switch(config-router)#neighbor 10.1.1.64 route-server-client
Switch(config-router)# neighbor 10.1.1.68 remote-as 300
Switch(config-router)# neighbor 10.1.1.68 route-server-client
```

1.8.3.66 neighbor send-community

Command: neighbor {<*ip-address*>|<*TAG*>} send-community [both|extended|standard]

no neighbor {<*ip-address*>|<*TAG*>} send-community [both|extended|standard]

Function: Configures whether sending the community attribute to the neighbors. The "**no neighbor {<*ip-address*>|<*TAG*>} send-community [both|extended|standard]**" command set to not sending.

Parameter: <*ip-address*>:IP address of the neighbor

<*TAG*>:Name of peer group

[both|extended|standard]: Standard community only, extended community or both.

Default: Sending the community attributes

Command Mode: BGP routing mode and address-family mode.

Usage Guide: The community attributes can be sent to the outside or not. By default of our company we set to sending while the default in standard protocol is not sending. By configuring this attribute community attributes will be carried when sending routing information's to the neighbors, or else not. Omission of the following choice will be equal to standard.

Example:

```
Switch(config-router)#no neighbor 10.1.1.66 send-community
```

Switch(config-router)#neighbor 10.1.1.66 send-community

1.8.3.67 neighbor shutdown

Command: neighbor {<*ip-address*>|<*TAG*>} shutdown

no neighbor {<*ip-address*>|<*TAG*>} shutdown

Function: Disconnect the neighbor connection. The "**no neighbor {<*ip-address*>|<*TAG*>} shutdown**" cancels this configuration

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of peer group

Default: Not disconnecting

Command Mode: BGP routing mode and address-family mode

Usage Guide: Directly disconnect/connect to a peer (group) without canceling the neighbor configuration

Example: Switch(config-router)#neighbor 10.1.1.64 shutdown

1.8.3.68 neighbor soft-reconfiguration inbound

Command:neighbor {<*ip-address*>|<*TAG*>} soft-reconfiguration inbound

no neighbor {<*ip-address*>|<*TAG*>} soft-reconfiguration inbound

Function: Configures whether perform inbound soft reconfiguration; the "**no neighbor {<*ip-address*>|<*TAG*>} soft-reconfiguration inbound**" command set to not perform the inbound soft reconfiguration

Parameter: <*ip-address*>:Neighbor IP address

<*TAG*>: Name of peer group

Default: Not perform inbound soft reconfiguration

Command Mode: The system saves the inbound messages in the buffer after the soft reconfiguration is set, will applies as soon as it restarts so to reduce consumptions of switching with other routers. The command is only available when the route refresh capability is not enabled

Example: Switch(config-router)#neighbor 11.1.1.120 soft-reconfiguration inbound

1.8.3.69 neighbor soo

Command: neighbor <*ip-addr*> soo <*soo-val*>

no neighbor <*ip-addr*> soo <*soo-val*>

Function: Configure the origin source from the neighbor route

Parameter: The neighbor IP address show in dotted decimal notation

<*soo-val*> is the origin source ,which the format is the same with RD

Command Mode: vrf mode

Usage Guide: If the user AS connects with several ISP devices, to avoid the user route returns to itself through P area, this attribute can be set. Once this attribute is set, it spreads with route. routes carrying SOO attributes will not be spreaded to a neighbor configured with the attribute

Example:

```
Switch(config)#ROUTER BGP 100
Switch(config-router)#address-family ipv4 vrf test
Switch(config-router-af)# neighbor 11.1.1.64 remote 200
Switch(config-router-af)# neighbor 11.1.1.64 soo 100:10
```

After this attribute set, the switch will no longer spreads the route with 100:10 rt attribute to 11.1.1.64. (what have to be mentioned here is that the soo attribute will be judged together with other rt attributes, which means if the rt is configured with the same attribute, it will be regarded as the origin neighbor even if it's not the real origin source. As a matter of fact, the normal configured soo are a single configuration which is different from rt/rd and unique within the accessible scope. In this way can only the origin concept be exactly expressed)

1.8.3.70 neighbor strict-capability-match

Command:neighbor {<*ip-address*>|<*TAG*>} strict-capability-match

no neighbor {<*ip-address*>|<*TAG*>} strict-capability-match

Function: Configure whether strict capability match is required when establishing connections.The "**no neighbor {<*ip-address*>|<*TAG*>} strict-capability-match**" command set to not requiring strict match.

Parameter: ***<ip-address>***: Neighbor IP address

<TAG>: Name of peer group

Default: No strict capability match configured

Command Mode: BGP routing mode and address-family mode.

Usage Guide: With this command, the connection can only be established when the both side are perfectly matched on capabilities.

Example: Switch(config-router)#neighbor 10.1.1.64 strict-capability-match

1.8.3.71 neighbor timers

Command: neighbor {<*ip-address*>|<*TAG*>} timers <*0-65535*> <*0-65535*>

no neighbor {<*ip-address*>|<*TAG*>} timers <*0-65535*> <*0-65535*>

Function: Configure the KEEPALIVE interval and hold time; the "**no neighbor {<*ip-address*>|<*TAG*>} timers <*0-65535*> <*0-65535*>**" command restores the defaults

Parameter: Neighbor IP address

<TAG>: Name of peer group

<0-65535>:Respectively the KEEPALIVE and HOLD TIME

Default: Default KEEPALIVE time is 60s, while HOLD TIME is 240s

Command Mode: BGP routing mode and address-family mode

Usage Guide: Send KEEPALIVE interval and HOLD TIME intervals sent in the peer connection. The hold time is the time period for maintain the connection when no message is received from the partner (such as KEEPALIVE). And the connection will be closed after this hold time.

Example: Switch(config-router)#neighbor 10.1.1.64 timers 50 200

1.8.3.72 neighbor timers connect

Command: neighbor {<*ip-address*>|<*TAG*>} timers connect <*0-65535*>

no neighbor {<*ip-address*>|<*TAG*>} timers connect [<*0-65535*>]

Function: Configure the connecting retry time interval. The "**no neighbor {<*ip-address*>|<*TAG*>} timers connect [<*0-65535*>]**" command restores the default value

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of peer group

<*0-65535*>: Retry interval

Default: 120s.

Command Mode: BGP routing mode and address-family mode

Usage Guide: Configure the connecting time interval when connecting a peer. The NO form restores the default value.

Example: Switch(config-router)#neighbor 10.1.1.64 timers connect 100

1.8.3.73 neighbor unsuppress-map

Command: neighbor {<*ip-address*>|<*TAG*>} unsuppress-map <*WORD*>

no neighbor {<*ip-address*>|<*TAG*>} unsuppress-map <*WORD*>

Function: Configure or cancel the unsurprising to conditions meet the specified route map. The "**no neighbor {<*ip-address*>|<*TAG*>} unsuppress-map <*WORD*>**" command cancels this configuration

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of peer group

<*WORD*>: Name of route-map

Default: Not set

Command Mode: BGP routing mode.

Usage Guide: This command is generally for route suppressed by the aggregated and summary-only conditions. Routes meet the route map conditions will still be send separately other than suppressed

Example:

```
Switch(config-router)#neighbor 10.1.1.66 unsuppress-map rmp
Switch(config)#access-list 10 permit 10.1.1.100 0.0.0.255
Switch(config)#route-map rmp permit 5
```

Switch(config-route-map)#match ip next-hop 10

Route with nexthop as 10.1.1.100 will not be restrained

1.8.3.74 neighbor update-source

Command: neighbor {<*ip-address*>|<*TAG*>} update-source <*IFNAME*>

no neighbor {<*ip-address*>|<*TAG*>} update-source <*IFNAME*>

Function: Configure the update source. The "**no neighbor {<*ip-address*>|<*TAG*>} update-source <*IFNAME*>**" cancels this configuration

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of peer group

<*IFNAME*>: Name or IP of the interface

Default: Not configured, namely use nearest interface

Command Mode: BGP routing mode.

Usage Guide: Specified update source is allowed to connect with any available interface which normally is the loop back interface. The NO forms restores to the nearest interface update source. Improper update source use may lead to neighbor connection unavailable, while the invalid interface causes problem which is also the reasons we use loop back interfaces. Note: the loop back interface should be maintained with its address accessibility to be able to establish connections when as the update source.

Example: Switch(config-router)#neighbor 10.1.1.66 update-source 192.168.0.1

1.8.3.75 neighbor version 4

Command: neighbor {<*ip-address*>|<*TAG*>} version 4

Function: Configure the BGP version of the partner

Parameter: <*ip-address*>: Neighbor IP address

<*TAG*>: Name of the peer group

4: Allowed BGP version, 4 only

Default: 4.

Command Mode: BGP routing mode.

Usage Guide: Only version 4 is supported so far, so whatever the configuration is the version remains at 4.

Example:

Switch(config-router)#neighbor 10.1.1.66 version 4

Switch(config-router)#

1.8.3.76 neighbor weight

Command: neighbor {<*ip-address*>|<*TAG*>} weight <*0-65535*>

no neighbor {<*ip-address*>|<*TAG*>} weight [<*0-65535*>]

Function: Configure the route weight sent from the partner. The "**no neighbor {<*ip-address*>|<*TAG*>} weight [<*0-65535*>]**" command restores the default value

Parameter: **<*ip-address*>**:Neighbor IP address

<*TAG*>:Name of IP address

<*0-65535*>: Weight

Default: The default weight acquired from other routers is 0. The default weight on the local static configuration is 32768

Command Mode: BGP routing mode.

Default: The default weight acquired from other routers is 0. The default weight on the local static configuration is 32768

Usage Guide: The path selecting can be affected through the configuration of the weight. The weight is only relevant to the router which is not an attribute transmittable to outside.

Example: Switch(config-router)#neighbor 10.1.1.66 weight 500

1.8.3.77 network (BGP)

Command: network <*ip-address/M*> [route-map <*WORD*>] [backdoor]

no network <*ip-address/M*> [route-map <*WORD*>] [backdoor]

Function: Configure the BGP managed network, the route map specified in network application, or set the "back door" for the network. the "**no network <*ip-address/M*> [route-map <*WORD*>] [backdoor]**" command cancels this configuration

Parameter: **<*ip-address/M*>**: Network prefix identifier

<*WORD*>: Name of route-map

Default: None

Command Mode: BGP routing mode.

Usage Guide: As for BGP routes, specify the route through which the BGP advertisements go. With the network defined by this command, the peer will be spreaded into the route map of the neighbor even if there is no route locally. Using the attribute specified in the network application through route map, when specifying the route comes from EBGP or inside the network through back door parameters, the inside route will be the optimized route even if the external route is of shorter distance.

Example: Switch(config-router)# network 172.16.0.0/16

1.8.3.78 redistribute (BGP)

Command: redistribute <*ROUTES*> [route-map <*WORD*>]

no redistribute <*ROUTES*> [route-map <*WORD*>]

Function: Set the BGP to redistribute route from other modes into BGP. The "**no redistribute <*ROUTES*> [route-map <*WORD*>]**" command cancels this configuration

Parameter: ***<ROUTES>***: Route source or protocol, including: connected, isis, kernel, ospf, rip, static, etc.

<WORD>: Name of route map

Default: None

Command Mode: BGP routing mode.

Usage Guide: Route from other ways will be distributed into the BGP route table with this command and transmitted to the neighbors

Example: The static route is introduced into BGP with this configuration and advertised to the neighbors

Switch(config-router)# redistribute static

1.8.3.79 redistribute ospf

Command: redistribute ospf [<*process-id*>] [route-map<*word*>]

no redistribute ospf [<*process-id*>]

Function: To redistribute routing information form OSPFv2 to BGP. The no form of this command will remove the configuration.

Parameters: process-id is the process id of the OSPFv2 process, limited between 1 and 65535. If no process id is specified, the default process id will be used.

route-map <word> is the pointer to the introduced routing map.

Default: Not redistributed by default.

Command Mode: bgp configuration mode.

Usage Guide: None.

Example: To redistribute ospfv2 to bgp (as number is 1).

Switch(config)#router bgp 1

Switch(config-router)#redistribute ospf 2

1.8.3.80 redistribute ospf (vrf)

Command: redistribute ospf [<*process-id*>] [route-map<*word*>]

no redistribute ospf [<*process-id*>]

Function: To redistribute the routing information from OSPFv2 processes to BGP within the local vrf. The no form of this command will disable the redistribution.

Parameters: process-id is the process id for the OSPFv2 process, limited between 1 and 65535. If no parameter is appended, the default process will be used.

route-map <word> is the pointer to the routing-map of the redistributed routing entries.

Default: Not redistributed by default.

Command Mode: BGP vrf configuration mode.

Usage Guide: None.

Example: To redistribute routing information from OSPF process with process id as 2 to BGP with AS number as 1 in vrf aaa.

```
Switch(config)#router bgp1
Switch(config-router)#address-family ipv4 vrf aaa
Switch(config-router-af)#redistribute ospf 2
```

1.8.3.81 rd

Command: rd <*rd-val*>

Function: Configure the VRF route identification label.

Parameter: <*rd-val*> is the route identification label, which normally should be（AS number or IP address）: digits, such as: 100:10

Command Mode: vrf mode

Usage Guide: Under VRF mode the configured RD is for identifying different VRF each of which shall have a unique RD; The BGP distinct routes with different VRF with this identification label. But attention should be paid on that once RD is configured, it will not be changed. So there is no form command to cancel this configuration and you have to reconfigure VRF

Example:

```
Switch(config)#ip vrf test
Switch(config-vrf)#rd 100:10
Switch(config-vrf)#
```

Above example creates a VRF named test with RD value at 100:10

1.8.3.82 router bgp

Command: router bgp <*as-id*>

no router bgp <*as-id*>

Function: Enable BGP instance. The "**no router bgp <*as-id*>**" command deletes BGP instance.

Parameter:<*as-id*>:1-65535 is AS number.

Default: BGP Not enabled

Command Mode: Global mode

Usage Guide: Enable BGP by specified AS, and then enter the config-router state, the protocol can be configured at this prompt.

Example:

```
Switch(config)#router bgp 200
Switch(config-router)#exit
```

1.8.3.83 route-target

Command: route-target {import|export|both} <*rt-val*>

no route-target {import|export|both} <*rt-val*>

Function: Configure the route extended community attributes, so to determine whether the route be spreaded to specific VRF.

Parameter: ***<rt-val>*** is the same as RD form, standing for the extended community attributes of the routes.

Command Mode: vrf mode

Usage Guide: Under VRF mode, the configured RT attributes decides which VRF will accept the route. There are 3 RT configurations: the import RT stands for the RT value acceptable by this VRF, the export RT represents the RT value carried with this VRF when routing spreading, both refers to above two option both enabled. If the export RT carried with the received route ever matches with the import RT of this VRF, then this VRF will accept this route or else not (except for the no bgp inbound-route-filter is configured which enanbles RD match). Several RT can be configured on the same VRF. Normally we set one RT with the both mode so to equal the RD and RT_VALUE.

Example:

```
Switch(config)#ip vrf test
Switch(config-vrf)#rd 100:10
Switch(config-vrf)#route-target both 100:10
Switch(config-vrf)#
```

In above example is created a VRF named test with RD value 100:10. the RT is configured bilateral. The RT-VALUE is equal to RD.

1.8.3.84 set vpnv4 next-hop

Command: set vpnv4 next-hop <*ip-addr*>

no set vpnv4 next-hop <*ip-addr*>

Function: Configure the nexthop of the VPNv4 route.

Parameter: ***<ip-addr>***is nexthop of vpnv4 route

Command Mode:vrf mode

Usage Guide: Configure VPNv4 route nexthop with this command. As normal nexthop settings are only for IPv4 route, this command specially configures the VPNv4 address-family.

Example:

Configure the address-family as follows:

```
Switch(config)#route-map map1 permit 15
Switch(config-map)#match interface Vlan1
Switch(config-map)#set weight 655
Switch(config-map)#set vpnv4 next-hop 10.1.1.250
Switch(config-map)#exit
Switch(config)#router bgp 100
Switch(config-router)#neighbor 10.1.1.68 remote-as 100
Switch(config-router)#neighbor 10.1.1.68 route-map map1 in
Switch(config-router)#address-family vpnv4 unicast
Switch(config-router-af)#neighbor 10.1.1.68 activate
Switch(config-router-af)#exit-address-family
```

View the routing message after refresh

```
Switch#show ip bgp vpn all
    Network            Next Hop              Metric LocPrf Weight Path
Route Distinguisher: 100:10 (Default for VRF test)
*> 11.1.1.0/24       11.1.1.64                 0              0 200 ?
*>i15.1.1.0/24       10.1.1.250                0     100    655 200 ?
*> 20.1.1.0/24       11.1.1.64                 0              0 200 ?
*>i100.1.1.0/24      10.1.1.250                0     100    655 200 ?
Route Distinguisher: 100:10
*>i15.1.1.0/24       10.1.1.68                 0     100      0 200 ?
*>i100.1.1.0/24      10.1.1.68                 0     100      0 200 ?
```

We can see that the nexthop 10.1.1.68 of the VPN route is changed to 10.1.1.250 after applied with route-map

1.8.3.85 timers bgp

Command: timers bgp <*0-65535*> <*0-65535*>

no timers bgp [<*0-65535*> <*0-65535*>]

Function: Configure all neighbor time in BGP. The “**no timers bgp [<*0-65535*> <*0-65535*>]**” command restores these times to default value

Parameter: Respectively the KEEPALIVE interval and the hold time

Default: KEEPALIVE is 60s, HOLD TIME is 240s.

Command Mode: BGP routing mode.

Usage Guide: Similar to neighbor time configuration which just performed on all neighbors

Example: Switch(config-router)# timers bgp 50 200

1.8.4 Configuration Examples of BGP

1.8.4.1 Examples 1: configure BGP neighbor

SwitchB, SwitchC and SwitchD are in AS200, SwitchA is in AS100. SwitchA and SwitchB share the same network segment. SwitchB and SwitchD are not connected physically.

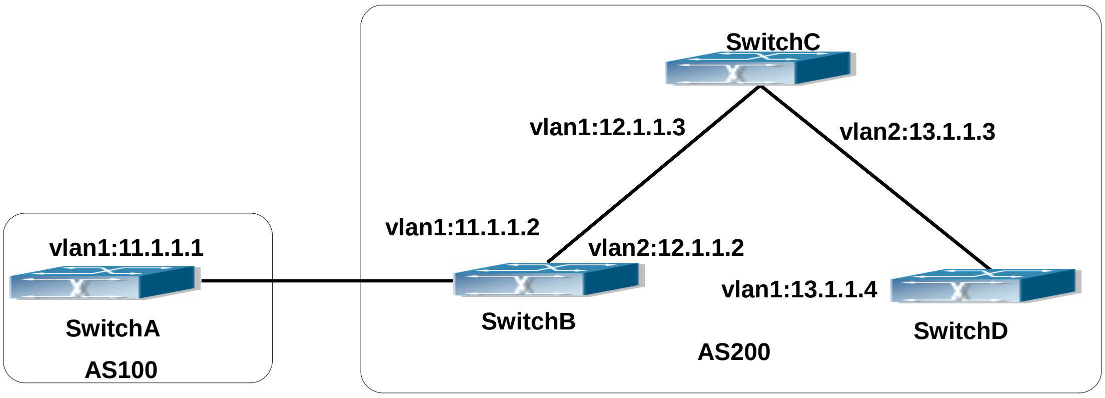

Fig 1-12 BGP Network Topological Map

The configurations of SwitchA are as following:

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
SwitchA(config-router-bgp)#exit
```

The configurations of SwitchB are as following:

```
SwitchB(config)#router bgp 200
SwitchB(config-router-bgp)#network 11.0.0.0
SwitchB(config-router-bgp)#network 12.0.0.0
SwitchB(config-router-bgp)#network 13.0.0.0
SwitchB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
SwitchB(config-router-bgp)#neighbor 12.1.1.3 remote-as 200
SwitchB(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchB(config-router-bgp)#exit
```

The configurations of SwitchC are as following:

```
SwitchC(config)#router bgp 200
SwitchC(config-router-bgp)#network 12.0.0.0
SwitchC(config-router-bgp)#network 13.0.0.0
SwitchC(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
SwitchC(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchC(config-router-bgp)#exit
```

The configurations of SwitchD are as following:

```
SwitchD(config)#router bgp 200
SwitchD(config-router-bgp)#network 13.0.0.0
SwitchD(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
SwitchD(config-router-bgp)#neighbor 13.1.1.3 remote-as 200
SwitchD(config-router-bgp)#exit
```

Presently, the connection between SwitchB and SwitchA is EBGP, and other connections with SwitchC and SwitchD are IBGP. SwitchB and SwitchD may have BGP connection without physical connection. But there is a precondition that these two switches must have reachable route to each other. This route can be attained through static route or IGP.

1.8.4.2 Examples 2: configure BGP aggregation

In this sample, configure route aggregation. Firstly, enable command redistribute to redistribute static route to BGP route table:

```
SwitchB(config)#ip route 193.0.0.0/24 11.1.12
SwitchB(config)#router bgp 100
SwitchB(config-router-bgp)#redistribute static
```

When there is at least one route affiliated to the specified range, the following configuration will create an aggregation route in the BGP route table. The aggregation route will be regarded as the AS from itself. More detailed route information about 193.0.0.0 will be announced.

```
SwitchB(config#router bgp 100
SwitchB(config-router-bgp)#aggregate 193.0.0.0/24
```

At the same time, the aggregation command above can be modified as following, then this switch only announce aggregation route 193.0.0.0 and forbid to announce more specified route to all the neighbors.

```
SwitchB(config-router-bgp)#aggregate 193.0.0.0/24 summary-only
```

1.8.4.3 Examples 3: configure BGP community attributes

In the following sample, “route map set-community” is used for the outgoing update to neighbor 16.1.1.6. By accessing to route in table 1 to configure special community value to “1111”, other can be announced normally.

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map set-community out
Switch(config-router-bgp)#exit
Switch(config)#route-map set-community permit 10
Switch(config-route-map)#match address 1
Switch(config-route-map)#set community 1111
Switch(config-route-map)#exit
Switch(config)#route-map set-community permit 20
Switch(config-route-map)#match address 2
Switch(config-route-map)#exit
Switch(config)#access-list 1 permit 11.1.0.0 0.0.255.255
Switch(config)#access-list 2 permit 0.0.0.0 255.255.255.255
Switch(config)#exit
Switch#clear ip bgp 16.1.1.6 soft out
```

In the following sample, configure the MED local preference of the routes from neighbor 16.1.1.6 selectively according to the route community value. All the routes that match the community list will set MED as 2000, community list com1 permits the route with community value “100 200 300”or”900 901” to pass. This route may have other community attributes. All the routes that pass community list com2 will set the local preference as 500. But the route that can’t pass both com1 and com2 will be rejected.

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map match-community in
Switch(config-router-bgp)#exit
Switch(config)#route-map match-community permit 10
Switch(config-route-map)#match community com1
Switch(config-route-map)#set metric 2000
Switch(config-route-map)#exit
Switch(config)#route-map match-community permit 20
Switch(config-route-map)#match community com2
Switch(config-route-map)#set local-preference 500
Switch(config-route-map)#exit
Switch(config)#ip community-list com1 permit 100 200 300
Switch(config)#ip community-list com1 permit 900 901
Switch(config)#ip community-list com2 permit 88
Switch(config)#ip community-list com2 permit 90
Switch(config)#exit
Switch#clear ip bgp 16.1.1.6 soft out
```

1.8.4.4 Examples 4: configure BGP confederation

The following is the configuration of an AS. As the picture illustrated, SwitchB and SwitchC establish IBGP connection. SwitchD is affiliated to AS 20.SwitchB and SwitchC establish EBGP of inner AS confederation. AS10 and AS20 form AS confederation with the AS number AS200; SwitchA belongs to AS100, SwitchB may create EBGP connection by AS200.

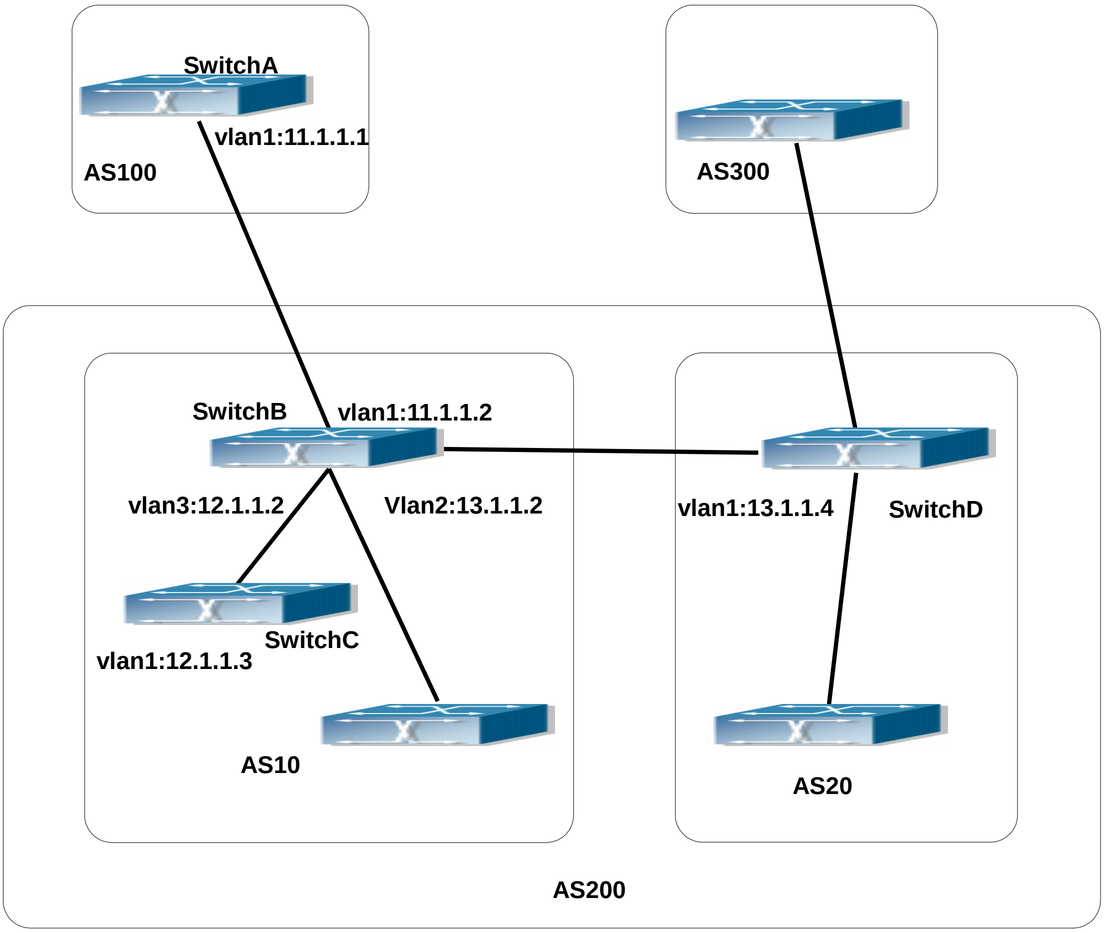

Fig 1-13 Confederation configuring topology

The configurations are as following:

SwitchA:

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
```

SwitchB:

```
SwitchB(config)#router bgp 10
SwitchB(config-router-bgp)#bgp confederation identifier 200
SwitchB(config-router-bgp)#bgp confederation peers 20
SwitchB(config-router-bgp)#neighbor 12.1.1.3 remote-as 10
SwitchB(config-router-bgp)#neighbor 13.1.1.4 remote-as 20
SwitchB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
```

SwitchC:

```
SwitchC(config)#router bgp 10
SwitchC(config-router-bgp)#bgp confederation identifier 200
SwitchC(config-router-bgp)#bgp confederation peers 20
SwitchC(config-router-bgp)#neighbor 12.1.1.2 remote-as 10
```

SwitchD:

```
SwitchD(config)#router bgp 20
SwitchD(config-router-bgp)#bgp confederation identifier 200
SwitchD(config-router-bgp)#bgp confederation peers 10
SwitchD(config-router-bgp)#neighbor 13.1.1.2 remote-as 10
```

1.8.4.5 Examples 5: configure BGP route reflector

The following is the configuration of a route reflector. As the picture illustrated, SwitchA, SwitchB, SwitchC, SwitchD, SWE, SWF and SWG establish IBGP connection which is affiliated to AS100. SwitchC creates EBGP connection with AS200. SwitchA creates EBGP connection with AS300. SwitchC, SwitchD and SWG make route reflectors.

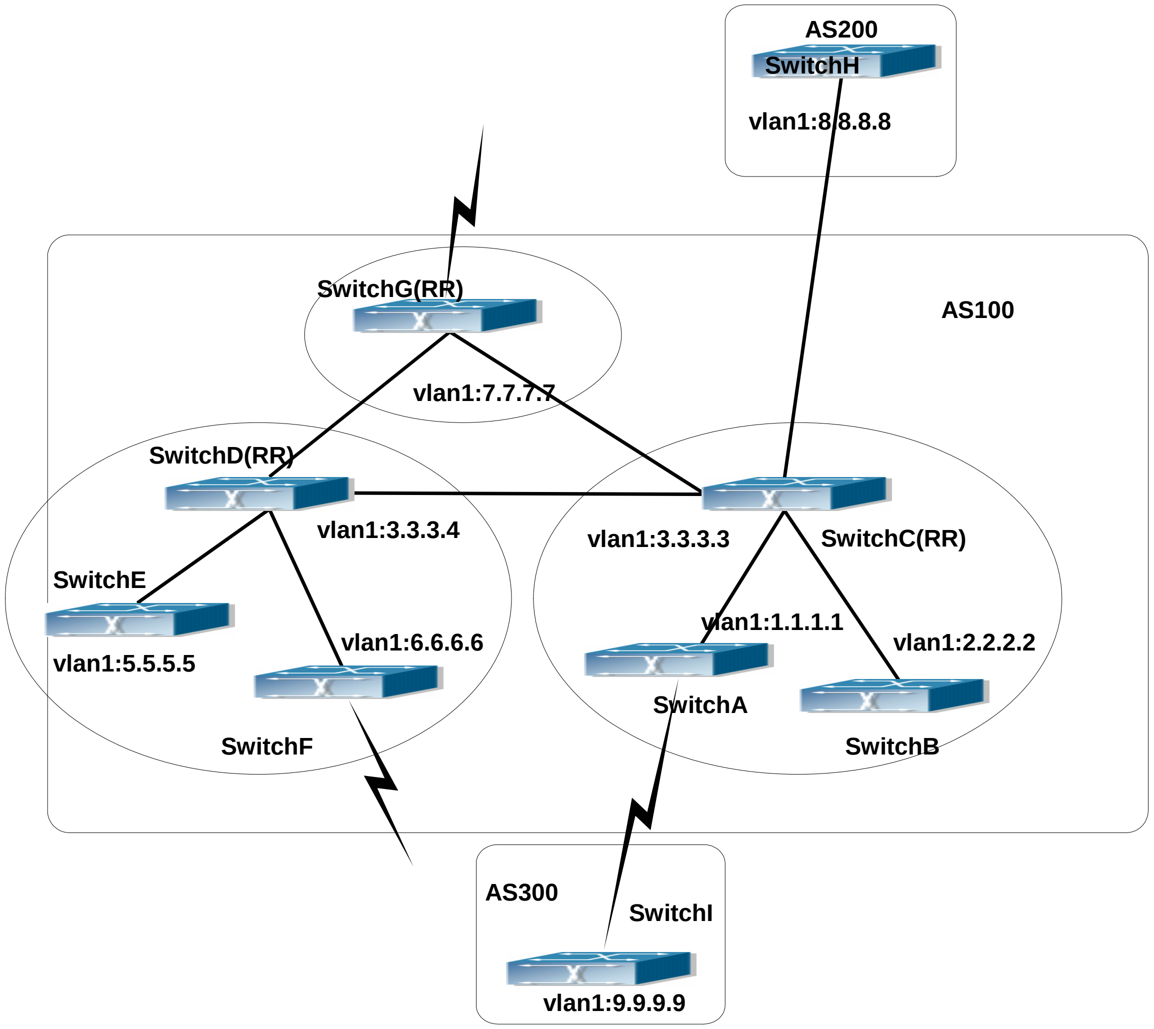

Fig 1-14 the Topological Map of Route Reflector

The configurations are as following:

The configurations of SwitchC:

```
SwitchC(config)#router bgp 100
SwitchC(config-router-bgp)#neighbor 1.1.1.1 remote-as 100
SwitchC(config-router-bgp)#neighbor 1.1.1.1 route-reflector-client
SwitchC(config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SwitchC(config-router-bgp)#neighbor 2.2.2.2 route-reflector-client
SwitchC(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
SwitchC(config-router-bgp)#neighbor 3.3.3.4 remote-as 100
SwitchC(config-router-bgp)#neighbor 8.8.8.8 remote-as 200
```

The configurations of SwitchD:

```
SwitchD(config)#router bgp 100
SwitchD(config-router-bgp)#neighbor 5.5.5.5 remote-as 100
SwitchD(config-router-bgp)#neighbor 5.5.5.5 route-reflector-client
SwitchD(config-router-bgp)#neighbor 6.6.6.6 remote-as 100
SwitchD(config-router-bgp)#neighbor 6.6.6.6 route-reflector-client
SwitchD(config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SwitchD(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
```

The configurations of SwitchA:

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 1.1.1.2 remote-as 100
SwitchA(config-router-bgp)#neighbor 9.9.9.9 remote-as 300
```

The SwitchA at this time needn’t to create IBGP connection with all the switches in the AS100 and could receive BGP route from other switches in the AS.

1.8.4.6 Examples 6: configure MED of BGP

The following is the configuration of a MED. As illustrated, SwitchA is affiliated to AS100, SwitchB is affiliated to AS400, SwitchC and SwitchD belong to AS300.

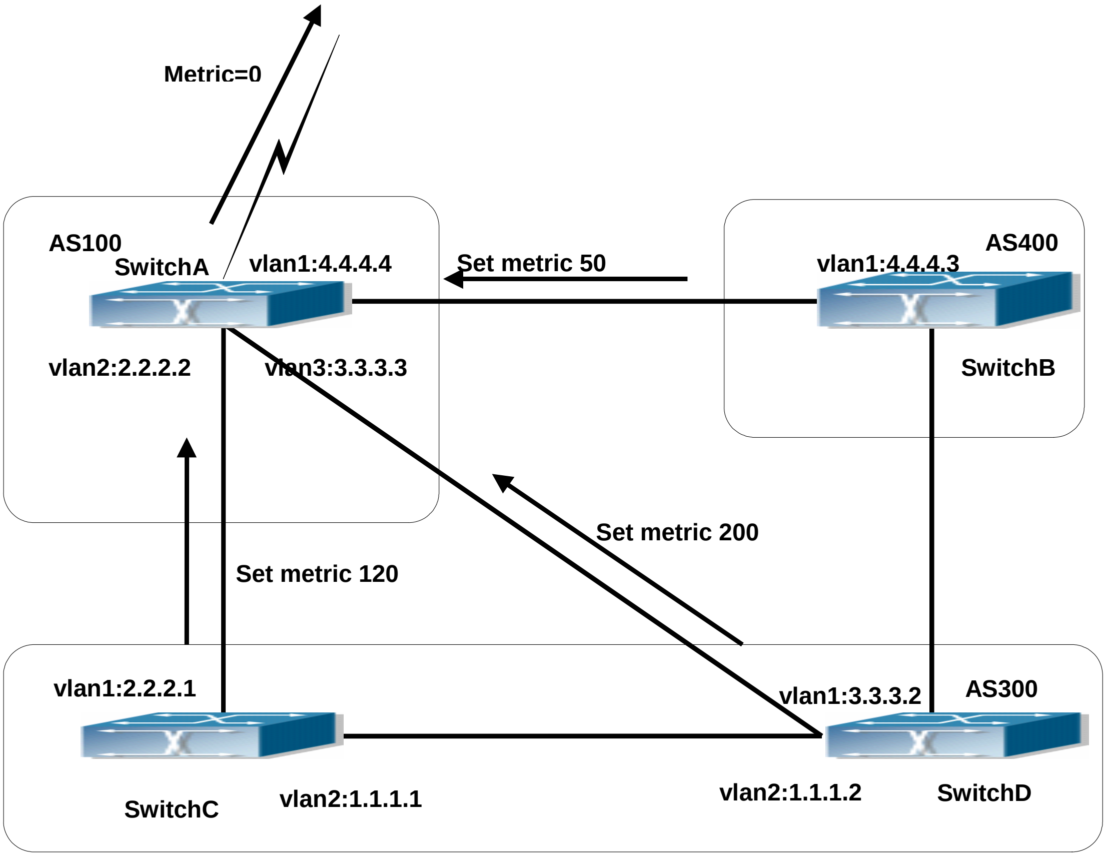

Fig 1-15 MED Configuring Topological Map

The configurations of SwitchA:

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 2.2.2.1 remote-as 300
SwitchA(config-router-bgp)#neighbor 3.3.3.2 remote-as 300
SwitchA(config-router-bgp)#neighbor 4.4.4.3 remote-as 400
```

The configurations of SwitchC:

```
SwitchC(config)#router bgp 300
SwitchC (config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SwitchC (config-router-bgp)#neighbor 2.2.2.2 route-map set-metric out
SwitchC (config-router-bgp)#neighbor 1.1.1.2 remote-as 300
SwitchC (config-router-bgp)#exit
SwitchC (config)#route-map set-metric permit 10
SwitchC (Config-Router-RouteMap)#set metric 120
```

The configurations of SwitchD

```
SwitchD (config)#router bgp 300
SwitchD (config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SwitchD (config-router-bgp)#neighbor 3.3.3.3 route-map set-metric out
SwitchD (config-router-bgp)#neighbor 1.1.1.1 remote-as 300
SwitchD (config-router-bgp)#exit
SwitchD (config)#route-map set-metric permit 10
SwitchD (Config-Router-RouteMap)#set metric 200
```

The configurations of SwitchB

```
SwitchB (config)#router bgp 400
SwitchB (config-router-bgp)#neighbor 4.4.4.4 remote-as 100
SwitchB (config-router-bgp)#neighbor 4.4.4.4 route-map set-metric out
SwitchB (config-router-bgp)#exit
SwitchB (config)#route-map set-metric permit 10
SwitchB (Config-Router-RouteMap)#set metric 50
SwitchA(config-router-bgp)# bgp always-compare-med
```

After the configuration above, SwitchB, SwitchC and SwitchD are assumed to send a route 12.0.0.0 to SwitchA. According to the comparison of BGP route strategy; there is an assumption that the routes sent by the three switches above have the same attribute value before the comparison of metric attribute. At this time, the route with lower value is the better route. But the comparison of metric attribute will only be done with the routes from the same AS. For SwitchA, the routes passed SwitchC are preferable to the one passed SwitchD. Because SwitchC and SwitchB are not located in the same AS, the SwitchA will not do metric comparison between the two switches. If the metric comparison between different AS is needed, the command" bgp always-compare-med" will be used. If this command is configured, the routes passed SwitchB are the best to SwitchA. At this time, the following command may be added on SwitchA: "SwitchA (config-router-bgp)# bgp always-compare-med"

1.8.5 BGP Troubleshooting Help

In the process of configuring and implementing BGP protocol, physical connection, configuration false probably leads to BGP protocol doesn't work. Therefore, the customers should give their attention to points as follow:

First of all, to ensure correct physical connection;

Secondly, to ensure interface and link protocol are UP (execute show interface instruction);

And startup BGP protocol (use router bgp command), configure affiliated IBGP and EBGP neighbors (use neighbor remote-as command).

Notice BGP protocol itself can't detect route, needs to import other routes to create BGP route. Only it enables these routes to announce IBGP and EBGP neighbors by importing routes.

Direct-link routes, static route, and IGP route (RIP and OSPF) are included in these imported routes. Network and redistribute (BGP) command are the ways of imported routes.

For BGP, pay attention to the difference between the behaviors of IBGP and EBGP.

After configuration finishes, the command of show ip bgp summary can be used to observe neighbor’s connections, so that all of the neighbors keep BGP connection situation. And use show ip bgp command to observe BGP routing table.

If BGP routing problem still can’t be solved by debugging, please use debug instructions like debug ip bgp packet/events etc, and copy DEBUG information in 3 minutes, then send them to ourTechnology Service Center.

1.8.5.1 Monitor And Debug Command

1.8.5.1.1 show ip bgp

Command: show ip bgp [<*ADDRESS-FAMILY*>] [<*ip-address*>|<*ip-address/M*> [longer-prefixes]| cidr-only]

Function: For displaying the routing messages permitted by BGP

Parameter: <*ADDRESS-FAMILY*>: address-family such as “ipv4 unicast”

<*ip-address*>: IP address

<*ip-address/M*>: IP address and the mask

Default: None

Command Mode: All mode

Usage Guide: We can display BGP routing messages by different parameters (such as address-family or IPv4 address), or a route covered by a prefix, or only the routing message don’t match the earliest IP address-family (namely the route is not A or B or C type address.)

Example:

```
Switch#show ip bgp
BGP table version is 147, local router ID is 10.1.1.64
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
*> 12.0.0.0         10.1.1.121               0         32768 ?
*> 100.1.1.0/24     10.1.1.200               0         32768 ?
*> 100.1.2.0/24     10.1.1.200               0         32768 ?
*> 172.0.0.0/8      0.0.0.0                            32768 i
Total number of prefixes 4
```

1.8.5.1.2 show ip bgp attribute-info

Command: show ip bgp attribute-info

Function: Display the BGP attributes messages

Parameter: None

Default: None

Command Mode: All modes.

Usage Guide: For displaying the attribute messages permitted by BGP

Example:

```
Switch#sh ip bgp attribute-info
attr[1] nexthop 0.0.0.0
attr[1] nexthop 10.1.1.64
attr[3] nexthop 10.1.1.64
attr[1] nexthop 10.1.1.121
attr[2] nexthop 10.1.1.200
```

1.8.5.1.3 show ip bgp community

Command: show ip bgp [<*ADDRESS-FAMILY*>] community <*TYPE*> [exact-match]

Function: For displaying route permitted by BGP with community information

Parameter: <*ADDRESS-FAMILY*>: Address-family, such as “ipv4 unicast”

<*TYPE*>: Community attributes number show in AA:NN form or combination of local-AS, no-advertise, and no-export.

Default: None

Command Mode: All mode

Usage Guide: We can choose several communities at a time, exact-match shows only the perfect match entries will be displayed.

Example:

```
Switch#show ip bgp community
BGP table version is 10, local router ID is 10.1.1.64
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
100.1.1.0/24       0.0.0.0                  32768 700 800 i
172.0.0.0/8        0.0.0.0                  32768 700 800 i
Total number of prefixes 2
```

1.8.5.1.4 show ip bgp community-info

Command: show ip bgp community-info

Function: For displaying the community messages permitted by BGP

Parameter: None

Default: None

Command Mode: All modes

Usage Guide: Messages in the same community multiply closable at the same time

Example:

```
Switch#show ip bgp community-info
Address Refcnt Community
[0x3312558] (3) 100:50
```

1.8.5.1.5 show ip bgp community-list

Command:show ip bgp [<*ADDRESS-FAMILY*>] community-list <*NAME*> [exact-match]

Function: For displaying the routes containing the community list messages and permitted by BGP

Parameter: **<*ADDRESS-FAMILY*>**: Address-family such as “ipv4 unicast”

<*NAME*>: Community list

Default: None

Command Mode: All mode

Usage Guide: Configure the community list with ip community-list command and the contained community as well. When displayed with its name, communities included in all the lists are contained

Example:

```
Switch(config)#ip community-list commu per 100:50
Switch#sh ip bgp community-list commu
BGP table version is 25, local router ID is 10.1.1.64
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
100.1.1.0/24       0.0.0.0                32768 700 800 i
172.0.0.0/8        0.0.0.0                32768 700 800 i
```

1.8.5.1.6 show ip bgp dampening

Command: show ip bgp [<ADDRESS-FAMILY>] dampening {<dampened-paths>|<flap-statistics>|<parameters>}

Function: Display the routes permitted by BGP and relevant to the route dampening.

Parameter: **<*ADDRESS-FAMILY*>**: Address-family, such as “ipv4 unicast”

Default: None

Command Mode: All mode

Usage Guide: Only the surged routes will be displayed. The Parameters shows the display configuration other than specific routes. The other two options will respectively show the restrained route and the dampening (recently recovered from invalid) routing messages.

Example:

```
Switch#sh ip bgp dampening dampened-paths
BGP table version is 12, local router ID is 10.1.1.66
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network                 From                    Reuse     Path
d 100.1.3.0/24        10.1.1.64                00:27:40 100
Total number of prefixes 1
Switch#sh ip bgp dampening flap-statistics
BGP table version is 13, local router ID is 10.1.1.66
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network                From                    Flaps   Duration   Reuse     Path
d 100.1.3.0/24        10.1.1.64                3          00:06:05   00:27:00   100 ?
Switch#sh ip bgp dampening parameters
dampening 15 750 2000 60 15 (route-map rmp)
   Reach ability Half-Life time      : 15 min
   Reuse penalty                          : 750
   Suppress penalty                      : 2000
   Max suppress time                     : 60 min
   Un-reach ability Half-Life time : 15 min
   Max penalty (ceil)                   : 11999
   Min penalty (floor)                 : 375
Total number of prefixes 1
```

1.8.5.1.7 show ip bgp filter-list

Command: show ip bgp [<*ADDRESS-FAMILY*>]filter-list [<*WORD* >]

Function: For displaying the routes in BGP meeting the specific AS filter list

Parameter: ***<ADDRESS-FAMILY>***: address-family such as “ipv4 unicast”

< *WORD* >: AS-PATH access-list name

Default: None

Command Mode: All modes

Usage Guide: Configure AS access-list with ip as-path access-list command. This command can show the routes passed the access-list.

Example:

```
Switch#SH IP BGP filter-list FL
BGP table version is 2, local router ID is 11.1.1.100
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
> 100.1.1.0/24       10.1.1.64                      0                  0 100 ?
Total number of prefixes 1
```

1.8.5.1.8 show ip bgp inconsistent-as

Command: show ip bgp [<*ADDRESS-FAMILY*>]inconsistent-as

Function: For displaying routes with inconsistent BGP AS

Parameter: <*ADDRESS-FAMILY*>: address family such as "ipv4 unicast"

Default: None

Command Mode: All modes

Usage Guide: If same prefix comes from different origin AS, the AS will be regarded as inconsistent. This command is for displaying this kind of routes.

Example:

```
Switch#sh ip bgp inconsistent-as
BGP table version is 2, local router ID is 11.1.1.100
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
  100.1.1.0/24       10.1.1.68                      0                  0 300
                     10.1.1.64                      0                  0 100
Total number of prefixes 1
```

1.8.5.1.9 show ip bgp neighbors

Command:show ip bgp [<*ADDRESS-FAMILY*>] neighbors [IP-ADDRESS] [advertised-routes|received {prefix-filter|routes}|routes]

Function: For displaying the BGP neighbor related messages

Parameter: <*ADDRESS-FAMILY*>: Address-family, such as "ipv4 unicast"

<*ip-address*>: Neighbor IP address

Default: None

Command Mode: All mode

Usage Guide: Display detailed messages of all neighbors by this command without parameters. Specifying IP address will show the detailed information of the neighbors with specified IP address. The advertised-routes、received prefix-filter、received routes、routes parameters will respectively displays the routes broadcast on local side, the received prefix filter, received routes (soft reconfiguration enabled) and the routing message from specific neighbor

Example:

```
Switch#sh ip bgp neighbor
BGP neighbor is 10.1.1.66, remote AS 200, local AS 100, external link
BGP version 4, remote router ID 11.1.1.100
BGP state = Established, up for 00:13:43
Last read 00:13:43, hold time is 240, keep alive interval is 60 seconds
Neighbor capabilities:
Route refresh: advertised and received (old and new)
Address family IPv4 Unicast: advertised and received
Received 17 messages, 0 notifications, 0 in queue
Sent 17 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
Minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 2, neighbor version 2
Index 1, Offset 0, Mask 0x2
Community attribute sent to this neighbor (both)
0 accepted prefixes
1 announced prefixes
 Connections established 7; dropped 6
```

1.8.5.1.10 show ip bgp paths

Command: show ip bgp [<*ADDRESS-FAMILY*>] paths

Function: Display the path message permitted by BGP

Parameter: <*ADDRESS-FAMILY*>: Address-family such as “ipv4 unicast”

Default: None

Command Mode: All modes

Usage Guide: Display the BGP path message includes the utilization state.

Example:

```
Switch#sh ip bgp paths
Address          Refcnt Path
[0x331dad0:0] (1)
[0x331d850:93] (1) 600
[0x331d8d8:249] (2) 200 300
```

1.8.5.1.11 show ip bgp prefix-list

Command: show ip bgp [<*ADDRESS-FAMILY*>] prefix-list [<*NAME*>]

Function: For displaying the route meet the specific prefix-list in BGP.

Parameter: <*ADDRESS-FAMILY*>: Address family such as “ipv4 unicast”

<*NAME*>: Name of prefix-list

Default: None

Command Mode: All mode

Usage Guide: We can select the required BGP route by regular expression

Example:

```
Switch(config)#ip prefix-list PL permit any
Switch(config)#
Switch#sh ip bgp prefix-list PL
BGP table version is 1, local router ID is 10.1.1.64
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
  100.1.1.0/24        10.1.1.66                               0 200 300 ?
                      10.1.1.100                  0           32768 ?
Total number of prefixes 1
```

1.8.5.1.12 show ip bgp quote-regexp

Command: show ip bgp [<*ADDRESS-FAMILY*>] quote-regexp [<WORD>]

Function: For displaying the BGP route meets the specific AS related regular expression.

Parameter: <ADDRESS-FAMILY>: **>**: address-family such as “ipv4 unicast”

<WORD>: Regular expression

Default: None

Command Mode: All modes

Usage Guide: Selecting the required route through regular expressions.

Example:

```
Switch#sh ip bgp quote-regexp ^300$
BGP table version is 2, local router ID is 11.1.1.100
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
100.1.1.0/24       10.1.1.68                    0                0 300
Total number of prefixes 1
Switch#sh ip bgp quote-regexp 100
BGP table version is 2, local router ID is 11.1.1.100
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
  100.1.1.0/24        10.1.1.64                   0                0 500 100 600
Total number of prefixes 1
```

1.8.5.1.13 show ip bgp regexp

Command: show ip bgp [<*ADDRESS-FAMILY*>] regexp [<*LINE*>]

Function: For displaying the BGP routes meets specific AS related normal expressions

Parameter: <*ADDRESS-FAMILY*>: **>**: address-family such as “ipv4 unicast”

<*LINE*>: Regular expression

Default: None

Command Mode: all modes.

Usage Guide: We can select BGP route of the required AS with normal expression

Example:

```
Switch#sh ip bgp regexp 100
BGP table version is 2, local router ID is 11.1.1.100
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
   100.1.1.0/24        10.1.1.64                    0                  0 500 100 600 ?
Total number of prefixes 1
```

1.8.5.1.14 show ip bgp route-map

Command: show ip bgp [<*ADDRESS-FAMILY*>] route-map [<*NAME*>]

Function: For displaying the BGP routes meets the specific related route map

Parameter: <*ADDRESS-FAMILY*>: **>**:”“ipv4 unicast” “ address-family such as “ipv4 unicast”

<*NAME*>: Name of route map

Default: None

Command Mode: All modes

Usage Guide: Configure the route map with the route-map command, through which it can be displayed that process routes with route map. The command will display the routes meet specific route map

Example:

```
Switch#sh ip bgp route-map rmp
BGP table version is 2, local router ID is 11.1.1.100
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
100.1.1.0/24            10.1.1.64                    0                  0 500 100 600
                        10.1.1.68                    0                  0 300
Total number of prefixes 1
```

1.8.5.1.15 show ip bgp redistribute

Command: show ip bgp redistribute [vrf <*NAME*>]

Function: To display redistributed routing entries from external processes to BGP.

Parameters: vrf name. If no parameter is appended, all the redistributed routing entries of BGP will be displayed.

Default: Not shown by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip bgp redistribute
```

1.8.5.1.16 show ip bgp neighbors

Command: show ip bgp neighbors [vrf <NAME>]

Function: Show neighbor information of specified bgp or total bgp processes.

Parameter: vrf name, show bgp neighbor information of all vrf if there is no parameter.

Default: Not shown by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ip bgp neighbors
```

1.8.5.1.17 show ip bgp scan

Command: show ip bgp scan

Function: For displaying BGP scan messages

Parameter: None

Default: None

Command Mode: All modes.

Usage Guide: Scan regularly the nexthop messages. The command can show the current interval and related routes.

Example:

```
Switch#show ip bgp scan
BGP Instance: (Default) AS 200, router-id 11.1.1.100
BGP scan interval is 60
Current BGP nexthop cache:
```

1.8.5.1.18 show ip bgp summary

Command: show ip bgp [<*ADDRESS-FAMILY*>] summary

Function: For displaying the BGP summary information

Parameter: ***<ADDRESS-FAMILY>***: Address-family such as “ipv4 unicast”

Default: None

Command Mode: All modes

Usage Guide: Display some basic summary information of BGP

Example:

```
Switch#show ip bgp summary
BGP router identifier 10.1.1.66, local AS number 200
BGP table version is 1
1 BGP AS-PATH entries
0 BGP community entries
Neighbor          V      AS MsgRcvd MsgSent     TblVer   InQ OutQ Up/Down    State/PfxRcd
10.1.1.68              4     300          0           0             0       0       0      never Active
Total number of neighbors 1
```

Display Contents	Explanation
identifier	Local identifier
local AS number	The number of AS of local router
table version	the version number of BGP interior database
AS-PATH entries	The tabulation of the AS-PATH entries
community entries	The property of the community entries
Neighbor	Neighbor address
V	The BGP version of neighbor running
AS	The AS number of neighbor what is affiliated with
MsgRcvd	The amount of message received from neighbor
MsgSent	The amount of message sent to the neighbor
TblVer	the version of route table
Up/Down	It will display the conversation time length if the state with neighbor was established, otherwise display the present status.
State/PfxRcd	If the state is established,display the amount of the prefix received of the router.otherwise,display the state of the neighbor at present.

1.8.5.1.19 show ip bgp view

Command: show ip bgp view [<*NAME*>] [<*ip-address*>|<*ip-address/M*>|[<*ADDRESS-FAMILY*>] summary]

Function: For displaying the messages of specified BGP instance

Parameter: <*NAME*>:Name of BGP instance

<*ip-address*>:IP address

<*ip-address/M*>:IP address and mask

<*ADDRESS-FAMILY*>: Address-family such as “ipv4 unicast”

Default: None

Command Mode: All modes

Usage Guide: Display messages of specified BGP instance

Example:

Switch#show ip bgp view as300 100.1.1.0/24

1.8.5.1.20 show ip bgp view neighbors

Command: show ip bgp view [<*NAME*>] neighbors [<*ip-address*>]

Function: Display neighbor messages of specified BGP instance

Parameter: <*NAME*>:Name of BGP instance

<*ip-address*>: neighbor IP address

Default: None

Command Mode: All mode

Usage Guide: Display neighbor messages of specified BGP instance

Example: Switch#show ip bgp view as300 neighbors

1.8.5.1.21 show ip bgp vpnv4

Command: show ip bgp vpnv4 {all|rd <*rd-val*>|vrf <*vrf-name*>}

Function: Display the BGP VPN routing messages

Parameter: <*rd-val*> is the route identification label which is normally the （AS number or IP address）: digits, such as 100:10; **<*vrf-name*>** is the name of VRF, created through if vrf<*vrf-name*> command

Command Mode: All modes

Usage Guide: Available to display by specified RD or VRF.

Example:

```
Switch#sh ip bgp vpn all
    Network              Next Hop                Metric LocPrf Weight Path
Route Distinguisher: 100:10 (Default for VRF test)
11.1.1.0/24        11.1.1.64                  0                0 200
20.1.1.0/24        11.1.1.64                  0                0 200
```

1.8.5.1.22 debug bgp

Command: debug bgp [<*MODULE*>|all]

no debug bgp [<*MODULE*>|all]

Function: For BGP debugging. the “**no debug bgp [<*MODULE*>|all]**” command closes the BGP debugging messages

Parameter: <*MODULE*>:BGP module names, including dampening、events、filters、fsm、keepalives、nsm、updates, etc.

Default: None

Command Mode: Admin mode and global mode

Usage Guide: For monitoring BGP events and the encountered errors, warning messages.

Example: Switch#debug bgp all

1.8.5.1.23 debug bgp redistribute message send

Command: debug bgp redistribute message send

no debug bgp redistribute message send

Function: To enable debugging of sending messages for redistribution of routing information from external process such as OSPF and RIP to BGP.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug bgp redistribute message send
Switch#no debug bgp redistribute message send
```

1.8.5.1.24 debug bgp redistribute route receive

Command: debug bgp redistribute route receive

no debug bgp redistribute route receive

Function: To enable debugging of received messages from NSM for BGP. The no form of this command will disable debugging of received messages from NSM for BGP.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug bgp redistribute route receive
Switch#no debug bgp redistribute route receive
```

1.9 MBGP4+

1.9.1 MBGP4+ Introduction

MBGP4+ is multi-protocol BGP (Multi-protocol Border Gateway Protocol) extension to IPv6, referring to BGP protocol chapter about BGP protocol introduction in this manual. Different from RIPng and OSPFv3, BGP has no corresponging independent protocol for IPv6, instead,it takes extensions to address families on the original BGP. The extensions to BGP by MBGP4+ are mostly embodied:

a. neighbor address configured can be IPv6 address;

b. Increase IPv6 unicast address family configuration.

1.9.2 MBGP4+ Configures Mission List

1. Configure IPv6 neighbor
2. Configure and enable IPv6 address family
3. Redistribution of OSPFv3 Routing to MBGP4+

（1）Enable Redistribution of OSPFv3 Routing to MBGP4+

（2）Display the information about configuration of redistribution of OSPFv3 Routing to MBGP4+

1.Configure IPv6 neighbor

Command	Explanation
BGP Protocol Configuration Mode	
neighbor <X:X::X:X> remote-as <*as-id*>	Configure IPv6 neighbor

2.Configure and activate IPv6 address family

Command	Explanation
BGP Protocol Configuration Mode	
address-family IPv6 unicast	Enter IPv6 unicast address family
BGP protocol address family comfiguration mode	
neighbor <X:X::X:X> activate **no neighbor <X:X::X:X> activate**	Configure IPv6 neighbor to activate/inactivate the address family
exit-address-family	Exit address family configuration mode

3. Redistribution of OSPFv3 Routing to MBGP4+

（1）To enable redistribution of OSPFv3 routing to MBGP4+

Command	Notes
Router ipv6 bgp configuration mode	
redistribute ospf [<*process-tag*>] [route-map<*word*>] **no redistribute ospf [<*process-tag*>]**	To enable or disable redistribution of OSPFv3 routing to BGP4+

（2）To display configuration information

Command	Notes

Admin mode and configuration mode	
show ipv6 bgp redistribute	To display configuration information about BGP4+ routing which is redistributed from other routing protocols.

（3）Debugging

Command	Notes
Admin mode	
debug ipv6 bgp redistribute message send **no debug ipv6 bgp redistribute message send** **debug ipv6 bgp redistribute route receive** **no debug ipv6 bgp redistribute route receive**	To enable or disable debugging messages sent by BGP4+ for redistribution of OSPFv3 routing. To enable or disable debugging messages received from NSM.

1.9.3 Command For BGP4+

1.9.3.1 redistribute ospf

Command: redistribute ospf [<*process-tag*>] [route-map<*word*>]

no redistribute ospf [<*process-tag*>]

Function: To redistribute routing information form OSPFv3 to BGP4+. The no form of this command will remove the configuration.

Parameters: process-id is the process id of the OSPFv3 process, limited between 1 and 65535. If no process id is specified, the default process id will be used.

route-map <word> is the pointer to the introduced routing map.

Default: Not redistributed by default.

Command Mode: BGP IPv6 configuration mode.

Usage Guide: None.

Example: To redistribute routing information from OSPFv3 process with the tag as abc to BGP4+ with the AS number as 1.

```
Switch(config)#router bgp 1
Switch(config-router)#address-family ipv6 unicast
Switch(config-router-af)#redistribute ospf abc
```

1.9.3.2 show ipv6 bgp redistribute

Command: show ipv6 bgp redistribute

Function: Show the configuration information of redistribution other out routing to bgp4+.

Parameter: None.

Default: Not shown by default.

Command Mode: Admin Mode and Configuration Mode.

Usage Guide: None.

Example:

```
Switch#show ipv6 bgp redistribute
```

1.9.3.3 debug ipv6 bgp redistribute message send

Command: debug ipv6 bgp redistribute message send

Function: To enable debugging of sending messages for redistribution of routing information from external process such as OSPFv3 and others to BGP4+.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 bgp redistribute message send
Switch#no debug ipv6 bgp redistribute message send
```

1.9.3.4 debug ipv6 bgp redistribute route receive

Command: debug ipv6 bgp redistribute route receive
no debug ipv6 bgp redistribute route receive

Function: To enable debugging of received messages from NSM for BGP4+. The no form of this command will disable debugging of received messages from NSM for BGP4+.

Parameter: None.

Default: Close the debug by default.

Command Mode: Admin Mode.

Usage Guide: None.

Example:

```
Switch#debug ipv6 bgp redistribute route receive
Switch#no debug ipv6 bgp redistribute route receive
```

1.9.4 MBGP4+ Examples

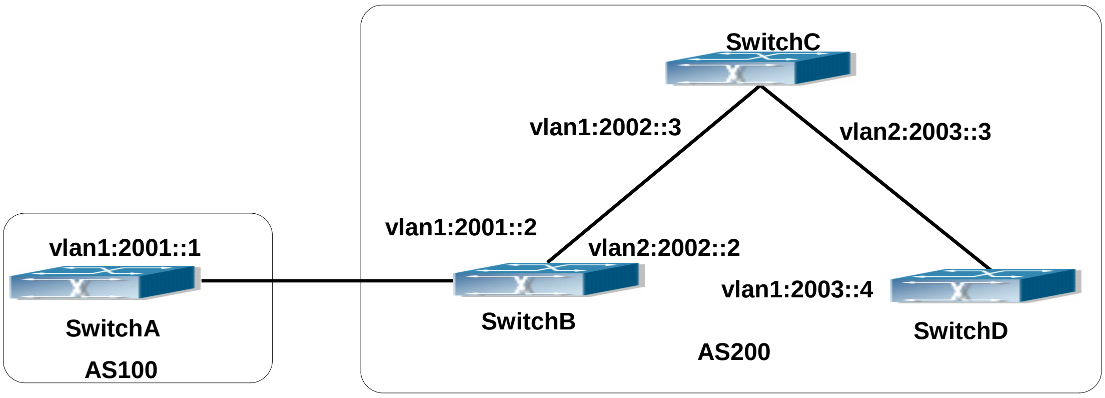

Fig 1-16 BGP Network Topological Map

Accordingly SwitchA configuration as follows:

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 2001::2 remote-as 200
SwitchA(config-router-bgp)#address-family IPv6 unicast
SwitchA(config-router-af)#neighbor 2001::2 activate
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router-bgp)#exit
SwitchA(config)#
```

SwitchB configuration as follows:

```
SwitchB(config)#router bgp 200
SwitchB(config-router-bgp)#neighbor 2001::1 remote-as 100
SwitchB(config-router-bgp)#neighbor 2002::3 remote-as 200
SwitchB(config-router-bgp)#neighbor 2003::4 remote-as 200
SwitchB(config-router-bgp)#address-family IPv6 unicast
SwitchB(config-router-af)#neighbor 2001::1 activate
SwitchB(config-router-af)#neighbor 2002::3 activate
SwitchB(config-router-af)#neighbor 2003::4 activate
SwitchB(config-router-af)#exit-address-family
SwitchB(config-router-bgp)#exit
SwitchB(config)#
```

SwitchC configuration as follows:

```
SwitchC(config)#router bgp 200
SwitchC(config-router-bgp)#neighbor 2002::2 remote-as 200
SwitchC(config-router-bgp)#neighbor 2003::4 remote-as 200
SwitchC(config-router-bgp)#address-family IPv6 unicast
SwitchC(config-router-af)#neighbor 2002::2 activate
SwitchC(config-router-af)#neighbor 2003::4 activate
SwitchC(config-router-af)#exit-address-family
SwitchC(config-router-bgp)#exit
```

SwitchD configuration as follows:

```
SwitchD(config)#router bgp 200
SwitchD(config-router-bgp)#neighbor 2003::3 remote-as 200
SwitchD(config-router-bgp)#neighbor 2002::2 remote-as 200
SwitchD(config-router-bgp)#address-family IPv6 unicast
SwitchD(config-router-af)#neighbor 2002::2 activate
SwitchD(config-router-af)#neighbor 2003::3 activate
SwitchD(config-router-af)#exit-address-family
SwitchD(config-router-bgp)#exit
```

Here the connection between SwitchB and SwitchA is EBGP, and the connection between SwitchC and SwitchD is IBGP. The BGP connection can be processed between SwitchB and SwitchD without physical link, but the premise is a route which reaches from one switch to the other switch. The route can be obtained by static routing or IGP.

1.9.5 MBGP4+ Troubleshooting Help

It is the same as corresponding section of BGP.

Chapter 2 Black Hole Routing Manual

2.1 Introduction to Black Hole Routing

Black Hole Routing is a special kind of static routing which drops all the datagrams that match the routing rule.

2.2 IPv4 Black Hole Routing Configuration Tasks

1. To configure IPv4 Black Hole Routing

1. To configure IPv4 Black Hole Routing

Command	Notes
Global Configuration Mode	
ip route {<*ip-prefix*> <*mask*> |<*ip-prefix*> | <*prefix-length*>} null0 [<*distance*>] **no ip route {<*ip-prefix*> <*mask*> | <*ip-prefix*> | <*prefix-length*>} null0**	To configure the static Black Hole Routing. The no form of this command will remove the specified black hole routing configuration.

2.3 IPv6 Black Hole Routing Configuration Task

1. Enable the IPv6 function
2. Configure the IPv6 Black Hole Routing

1. Enable the IPv6 function

Command	Notes
Global Configuration Mode	
ipv6 enable	To enable the IPv6 function on the switch.

2.Configure the IPv6 Black Hole Routing

Command	Notes

Global Configuration Mode	
ipv6 route *<ipv6-prefix/prefix-length>* null0 [*<precedence>*] **no ipv6 route *<ipv6-prefix/prefix-length>* null0**	To configure static IPv6 black hole routing. The no form of this command will remove the specified configuration.

2.4 Black Hole Routing Command

2.4.1 ip route null0

Command: ip route {*<ip-prefix> <mask>*|*<ip-prefix>*|*<prefix-length>*} null0 [*<distance>*]

no ip route {*<ip-prefix> <mask>*|*<ip-prefix>*|*<prefix-length>*} null0

Function: To configure routing destined to the specified network to the interface of null0.

Parameters: <ip-prefix> and **<mask>** are the IP address and network address mask of the destination, in dotted decimal format. **<ip-prefix>** and **<prefix-length>** are the IP address of the destination and the length of the prefix respectively. **null0** is the output interface for the black hole routing. **<distance>** is the management distance of the routing entry with limitation between 1 and 255.

Default: None.

Command Mode: Global configuration mode.

Usage Guide: null0 should be used as the output interface for IPv4 black hole routing.

Example: To configure the routing to 192.168.188.0/24 as a black hole routing.

```
Switch(config)#ip route 192.168.188.0/24 null0 20
```

2.4.2 ipv6 route null0

Command: ipv6 route *<ipv6-prefix><prefix-length>* null0 [*<precedence>*]

no ipv6 route *<ipv6-prefi><|prefix-length>* null0

Function: To configure routing destined to the specified network to the interface of null0.

Parameters: <ipv6-prefix> and **<mask>** are the IPv6 address and network address mask of the destination, in dotted decimal format. **<ipv6-prefix>** and **<prefix-length>** are the IPv6 address of the destination and the length of the prefix respectively. **null0** is the output interface for the black hole routing. **<distance>** is the management distance of the routing entry with limitation between 1 and 255.

Default: None.

Command Mode: Global configuration mode.

Usage Guide: When configuring IPv6 black hole routing, it is much like configuring normal static routing, but using null0 as the output interface.

Example: To configure a route to 2001:2:3:4::/64 as a black hole routing.

```
Switch(config)#ipv6 route 2001::/64 null0
```

2.5 Black Hole Routing Configuration Exmaple

Example 1: To configure IPv6 black hole routing.

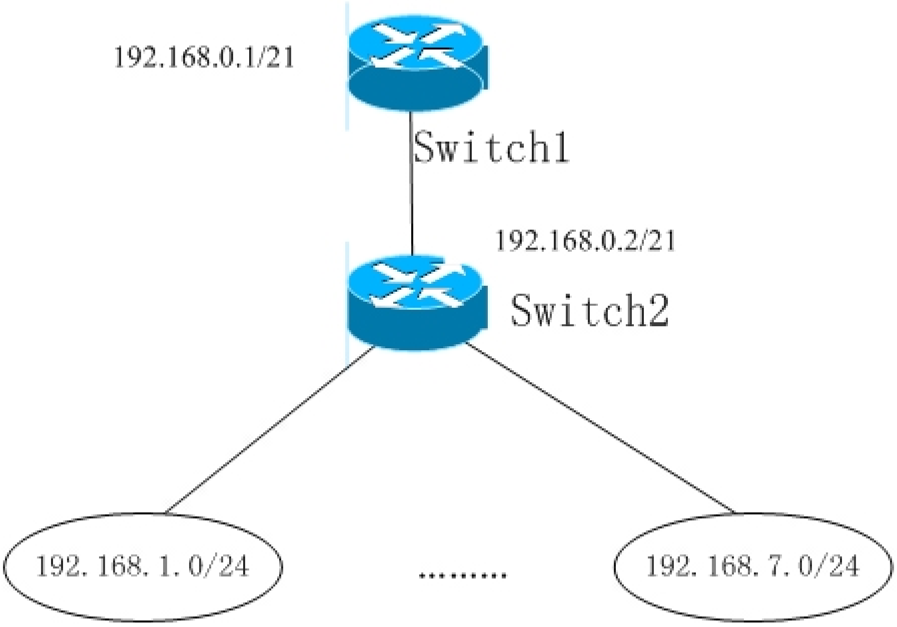

Fig 2-1 IPv4 Black Hole Routing Configuration Example

As it is shown in the figure, in Switch2, eight in all interfaces are configured as Layer 3 VLAN interfaces for access interfaces. The network addresses are 192.168.1.0/24 ~ 192.268.7.0/24. A default routing is configured on Switch2 to connect to Switch1. And a backward default routing is configured on Switch1 to Switch2, whose network address is 192.168.0.0/21. Commonly, this configuration will work well. However, if one of the Layer 3 interfaces in Switch2 goes down, for example, the interface belonged to 192.168.1.0/24. When datagrams arrives at VLAN1 in Switch2, there will be no routing rules for these datagrams. The switch then will forward these datagrams according to the default routing, back to Switch1. When Switch1 receives these datagrams, it will forward them back to Switch2. Thus, loopback exists. To solve this problem, black hole routing can be introduced on Switch2.

```
ip route 192.168.0.0/21 null0 50
```

Then Switch2 will drop the datagrams from interface vlan1 that match the black hole routing rule.

Configuration steps are listed as below:

```
Switch#config
Switch(config)#ip route 192.168.0.0/21 null0 50
```

Example 2: IPv6 Black Hole Routing Configuration.

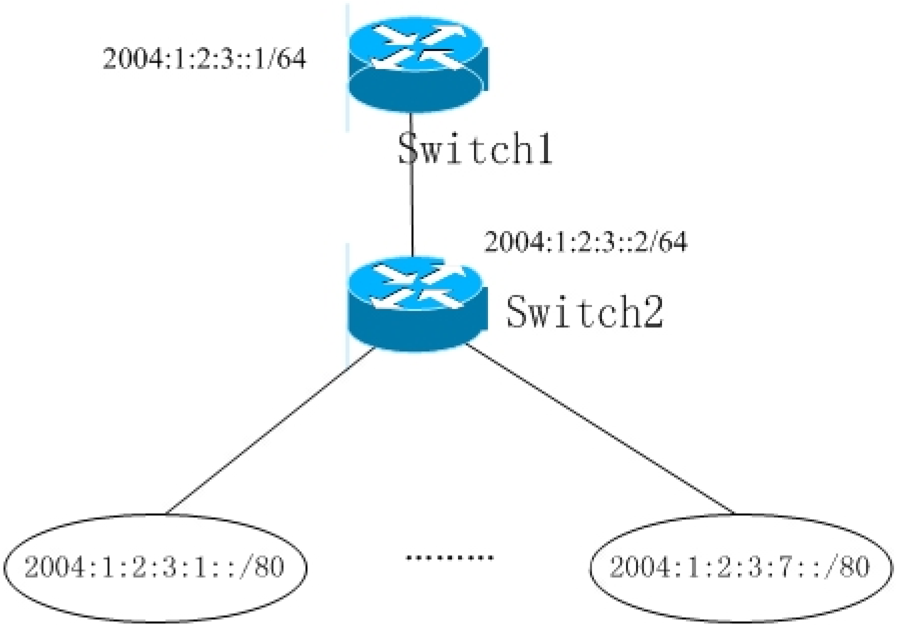

Fig 2-2 IPv6 Black Hole Routing Configuration Example

As it is shown in the figure, in Switch2, eight in all interfaces are configured as Layer 3 VLAN interfaces for access interfaces. The network addresses are 2004:1:2:3:1/80~2004:1:2:3:7/80. A default routing is configured on Switch2 to connect to Switch1. And a backward default routing is configured on Switch1 to Switch2, whose network address is 2004:1:2:3::/64. Commonly, this configuration will work well. However, if one of the Layer 3 interfaces in Switch2 goes down, for example, the interface belonged to 2004:1:2:3:1/80. When datagrams arrives at VLAN1 in Switch Switch2, there will be no routing rules for these datagrams. The switch then will forward these datagrams according to the default routing, back to Switch1. When Switch1 receives these datagrams, it will forward them back to Switch2. Thus, loopback exists. To solve this problem, black hole routing can be introduced on Switch2.

```
ipv6 route 2004:1:2:3::/77 null0 50
```

Then Switch2 will drop the datagrams from interface vlan1 that match the black hole routing rule. And loopback routing is prevented.

Configuration steps:

```
Switch#config
Switch(config)#ipv6 route 2004:1:2:3::/64 null0 50
```

2.6 Black Hole Routing Trouble Shooting

When configuring the Black Hole Routing function, the configuration may not work due to some reasons such as incorrect network address mask, and incorrect management distance. Attention should be paid to the following items:

- ✧ IPv6 should be enabled before IPv6 Black Hole Routing can work.
- ✧ It is suggested that the length of the network address mask should be longer than that of normal routing configuration, in order to prevent the black hole routing from intervening other routing configuration.
- ✧ When the network address mask of black hole routing configuration is the same with some other configuration, it is suggested that the distance of black hole routing is set lower.
- ✧ For problems that cannot be fixed through above methods, please issue the command show ip route distance and show ip route fib, and show l3. And copy and paste the output of the commands, and send to Center for support.