

# Routing Protocol Configuration Commands

# Table of Contents

Chapter 1 RIP Configuration Commands .....	1
1.1 RIP Configuration Commands .....	1
1.1.1 auto-summary .....	2
1.1.2 default-information originate .....	3
1.1.3 <b>default-metric</b> .....	4
1.1.4 <b>ip rip authentication</b> .....	5
1.1.5 <b>ip rip message-digest-key</b> .....	5
1.1.6 <b>ip rip passive</b> .....	6
1.1.7 <b>ip rip password</b> .....	7
1.1.8 ip rip receive version .....	8
1.1.9 ip rip send version .....	9
1.1.10 ip rip split-horizon .....	10
1.1.11 neighbor .....	11
1.1.12 network .....	12
1.1.13 offset .....	13
1.1.14 router rip .....	14
1.1.15 timers expire .....	15
1.1.16 timers holddown .....	16
1.1.17 timers update .....	17
1.1.18 validate-update-source .....	18
1.1.19 version .....	18
1.1.20 distance .....	19
1.1.21 filter .....	20
1.1.22 maximum-count .....	22
1.1.23 show ip rip .....	23
1.1.24 show ip rip database .....	24
1.1.25 show ip rip protocol .....	25
1.1.26 debug ip rip database .....	26
1.1.27 debug ip rip protocol .....	26
Chapter 2 BEIGRP Configuration Commands .....	29
2.1 BEIGRP Configuration Commands .....	29
2.1.1 auto-summary .....	30
2.1.2 clear ip beigrp neighbors .....	31
2.1.3 debug ip beigrp .....	32
2.1.4 debug ip beigrp fsm .....	32
2.1.5 debug ip beigrp neighbors .....	33
2.1.6 debug ip beigrp packet .....	34
2.1.7 debug ip beigrp transmit .....	35
2.1.8 default-metric .....	36
2.1.9 distance .....	37
2.1.10 filter .....	39

2.1.11 beigrp log-neighbor-changes .....	40
2.1.12 beigrp router-id .....	40
2.1.13 ip beigrp bandwidth-percent .....	41
2.1.14 ip beigrp hello-interval .....	42
2.1.15 ip beigrp hold-time .....	43
2.1.16 ip beigrp passive .....	43
2.1.17 ip beigrp split-horizon .....	44
2.1.18 ip beigrp summary-address .....	45
2.1.19 metric weights .....	46
2.1.20 network .....	47
2.1.21 offset .....	48
2.1.22 redistribute .....	49
2.1.23 router beigrp .....	50
2.1.24 show ip beigrp interface .....	51
2.1.25 show ip beigrp neighbors .....	52
2.1.26 show ip beigrp protocol .....	53
2.1.27 show ip beigrp topology .....	54
2.1.28 show ip beigrp traffic .....	56
Chapter 3 OSPF Configuration Commands .....	58
3.1 OSPF Configuration Commands .....	58
3.1.1 area authentication .....	59
3.1.2 area default-cost .....	60
3.1.3 area range .....	62
3.1.4 area stub .....	63
3.1.5 area virtual-link .....	64
3.1.6 debug ip ospf adj .....	67
3.1.7 debug ip ospf events .....	68
3.1.8 debug ip ospf flood .....	69
3.1.9 debug ip ospf lsa-generation .....	70
3.1.10 debug ip ospf packet .....	71
3.1.11 debug ip ospf retransmission .....	72
3.1.12 debug ip ospf spf .....	72
3.1.13 debug ip ospf tree .....	74
3.1.14 default-information originate (OSPF) .....	75
3.1.15 default-metric .....	76
3.1.16 distance ospf .....	77
3.1.17 filter .....	78
3.1.18 ip ospf authentication .....	79
3.1.19 ip ospf cost .....	80
3.1.20 ip ospf dead-interval .....	81
3.1.21 ip ospf hello-interval .....	82
3.1.22 ip ospf message-digest-key .....	83
3.1.23 ip ospf network .....	84
3.1.24 ip ospf passive .....	85
3.1.25 ip ospf password .....	86
3.1.26 ip ospf priority .....	87

---

3.1.27 ip ospf retransmit-interval .....	88
3.1.28 ip ospf transmit-delay .....	88
3.1.29 neighbor .....	89
3.1.30 network area .....	91
3.1.31 redistribute .....	92
3.1.32 router ospf .....	92
3.1.33 show ip ospf .....	93
3.1.34 show ip ospf border-routers .....	95
3.1.35 show ip ospf database .....	96
3.1.36 show ip ospf interface .....	97
3.1.37 show ip ospf neighbor .....	98
3.1.38 show ip ospf virtual-link .....	100
3.1.39 summary-address .....	101
3.1.40 timers delay .....	102
3.1.41 timers hold .....	103
Chapter 4 BGP Configuration Commands .....	104
4.1.1 aggregate-address .....	105
4.1.2 bgp always-compare-med .....	107
4.1.3 bgp bestpath med .....	108
4.1.4 bgp client-to-client reflection .....	109
4.1.5 bgp cluster-id .....	110
4.1.6 bgp confederation identifier .....	111
4.1.7 bgp confederation peers .....	112
4.1.8 bgp dampening .....	113
4.1.9 bgp default .....	115
4.1.10 bgp deterministic-med .....	116
4.1.11 bgp redistribute-internal .....	116
4.1.12 clear ip bgp .....	117
4.1.13 debug chat .....	119
4.1.14 debug dialer .....	120
4.1.15 debug ip bgp .....	121
4.1.16 distance .....	122
4.1.17 filter .....	123
4.1.18 neighbor default-originate .....	125
4.1.19 neighbor description .....	126
4.1.20 neighbor distribute-list .....	127
4.1.21 neighbor ebgp-multihop .....	128
4.1.22 neighbor filter-list .....	129
4.1.23 neighbor maximum-prefix .....	130
4.1.24 neighbor next-hop-self .....	131
4.1.25 neighbor password .....	133
4.1.26 neighbor prefix-list .....	134
4.1.27 neighbor remote-as .....	135
4.1.28 neighbor route-map .....	136
4.1.29 neighbor route-reflector-client .....	138
4.1.30 neighbor route-refresh .....	139

4.1.31 neighbor send-community .....	140
4.1.32 neighbor shutdown .....	141
4.1.33 neighbor soft-reconfiguration .....	142
4.1.34 neighbor timers .....	143
4.1.35 neighbor update-source .....	144
4.1.36 neighbor weight .....	145
4.1.37 network (BGP) .....	146
4.1.38 redistribute (BGP) .....	147
4.1.39 router bgp .....	149
4.1.40 show ip bgp .....	150
4.1.41 show ip bgp community .....	151
4.1.42 show ip bgp neighbors .....	152
4.1.43 show ip bgp paths .....	153
4.1.44 show ip bgp prefix-list .....	153
4.1.45 show ip bgp regexp .....	154
4.1.46 show ip bgp summary .....	155
4.1.47 synchronization .....	156
4.1.48 table-map .....	157
4.1.49 timers .....	158
Chapter 5 Public Routing Configuration Commands .....	160
5.1 Ip aspath-list Configuration Commands .....	160
5.1.1 ip as-path access-list .....	160
5.1.2 show ip aspath-list .....	162
5.2 ip community-list Configuration Commands .....	163
5.2.1 ip community-list .....	163
5.2.2 show ip community-list .....	164
5.3 ip prefix-list commands .....	165
5.3.1 clear ip prefix-list .....	165
5.3.2 ip prefix-list .....	166
5.3.3 ip prefix-list description .....	168
5.3.4 ip prefix-list sequence-number .....	169
5.3.5 show ip prefix-list .....	170
5.4 route-map Commands .....	172
5.4.1 route-map .....	172
5.4.2 match as-path .....	175
5.4.3 match community .....	176
5.4.4 match ip address .....	178
5.4.5 match ip next-hop .....	179
5.4.6 match ip address prefix-list .....	181
5.4.7 match length .....	182
5.4.8 match metric .....	183
5.4.9 match tag .....	185
5.4.10 on-match .....	186
5.4.11 set aggregator .....	188
5.4.12 set as-path .....	189
5.4.13 set atomic-aggregate .....	191

5.4.14 set community .....	192
5.4.15 set community-additive.....	194
5.4.16 set dampening.....	196
5.4.17 set default.....	197
5.4.18 set interface.....	198
5.4.19 set ip default.....	198
5.4.20 set ip precedence .....	199
5.4.21 set ip tos .....	200
5.4.22 set ip next-hop.....	201
5.4.23 set local-preference.....	203
5.4.24 set metric.....	204
5.4.25 set metric-type.....	206
5.4.26 set origin.....	207
5.4.27 set tag.....	209
5.4.28 set weight .....	210
5.4.29 show route-map.....	212
Chapter 6 RSVP Configuration Command .....	214
6.1.1 debug ip rsvp local .....	214
6.1.2 debug ip rsvp packet .....	215
6.1.3 ip rsvp bandwidth .....	217
6.1.4 ip rsvp local reservation.....	218
6.1.5 ip rsvp local sender .....	219
6.1.6 ip rsvp local session .....	220
6.1.7 ip rsvp neighbor.....	221
6.1.8 ip rsvp precedence.....	222
6.1.9 ip rsvp tos .....	223
6.1.10 show ip rsvp installed .....	223
6.1.11 <b>show ip rsvp interface</b> .....	225
6.1.12 <b>show ip rsvp local</b> .....	226
6.1.13 <b>show ip rsvp neighbor</b> .....	227
6.1.14 <b>show ip rsvp precedence</b> .....	228
6.1.15 show ip rsvp <b>reservation</b> .....	230
6.1.16 show ip rsvp sender .....	231
6.1.17 show ip rsvp tos.....	232
Chapter 7 PBR Configuration Commands.....	234
7.1 PBR Configuration Commands .....	234
7.1.1 debug ip policy .....	234
7.1.2 ip policy route-map.....	235
7.1.3 match ip address.....	236
7.1.4 match length.....	237
7.1.5 set default interface .....	238
7.1.6 set interface.....	239
7.1.7 set ip default next-hop.....	240
7.1.8 set ip next-hop.....	241
7.1.9 route-map .....	242
7.1.10 debug ip policy .....	243

7.1.11 ip local policy .....	244
7.1.12 ip policy .....	246
7.1.13 ip route-weight.....	247
7.1.14 show ip local policy.....	248
7.1.15 show ip policy .....	249
Chapter 8 DNSR Configuration Commands.....	251
8.1 DNSR Configuration Commands.....	251
8.1.1 ip domain lookup .....	251
8.1.2 ip domain name-server.....	252
8.1.3 ip domain name.....	253
8.1.4 ip domain list .....	253
8.1.5 ip host.....	254
8.1.6 ip domain retry.....	255
8.1.7 ip domain timeout.....	256
8.1.8 clear ip host.....	256
8.1.9 ip domain primary-server.....	257
8.1.10 ip domain dynamic enable.....	258
8.1.11 ip domain dynamic period .....	258
8.1.12 ip domain bind .....	259
8.1.13 show ip host .....	260
8.1.14 debug ip domain.....	260
Chapter 9 PeanutHull Configuration Commands.....	262
9.1.1 ip peanuthull .....	262
9.1.2 enable.....	263
9.1.3 server .....	263
9.1.4 domain.....	264
9.1.5 port .....	265
9.1.6 username .....	265
9.1.7 password.....	266
9.1.8 bind.....	267
9.1.9 show ip peanuthull.....	267
9.1.10 debug ip domain.....	268
Chapter 10 Network Management Configuration Commands.....	269
10.1 Network Management Configuration Commands.....	269
10.1.1 distance .....	269
10.1.2 filter in.....	271
10.1.3 filter out.....	272
10.1.4 redistribute.....	274

# Chapter 1 RIP Configuration Commands

## 1.1 RIP Configuration Commands

RIP configuration commands include:

- auto-summary
- default-information originate
- default-metric
- ip rip authentication
- ip rip message-digest-key
- ip rip passive
- ip rip password
- ip rip receive version
- ip rip send version
- ip rip split-horizon
- neighbor
- network
- offset
- router rip
- timers expire
- timers holddown
- timers update
- validate-update-source
- version
- distance
- filter
- maximum-count
- show ip rip



- show ip rip database
- show ip rip protocol
- debug ip rip database
- debug ip rip protocol

### 1.1.1 auto-summary

To activate the automatic routing summary of X path(s), run **auto-summary**; to disable the automatic routing summary of X path(s), run **no auto-summary**.

**auto-summary**

**no auto-summary**

#### Parameter

The command has no parameters or keywords.

#### Default

The automatic routing summary is used by default.

#### Command mode

Routing configuration mode

#### Explanation

The routing summary function decreases the routing information volume in the routing table and the information exchange volume. RIP-1 does not support the subnet mask. If the subnet route is forwarded, misunderstanding may occur. Hence, RIP-1 always starts the routing summary function. If RIP-2 is used, you can disable the routing summary function by running the **no auto-summary** function. If you want to broadcast the subnet route, disable the routing summary function.

#### Example

The following example shows that the RIP version on interface Serial1/0 is set to RIP-2 and the routing summary function is disabled.

```
router rip
version 2
no auto-summary
```

**Related command**

**version**

**1.1.2 default-information originate**

To generate a default route, run **default-information originate**. To disable the function, run **no default-information originate**.

**default-information originate**

**no default-information originate**

**Parameter**

None

**Default**

The function to generate a default route is disabled.

**Command mode**

Routing configuration mode

**Explanation**

After **default-information originate** is activated, the **0.0.0.0/0** routing information will be carried when the routing update is transmitted.

If a default route with a management distance of less than 120 exists in the main routing table, the command validates.

**Example**

When the routing update is transmitted, a default route (0.0.0.0/0) will be carried.

```
!  
router rip  
version 2  
network 172.68.16.0  
default-information originate  
!  
ip route default f0/0  
!
```

### 1.1.3 default-metric

To set the default route cost which is guided into the route, run **default-metric number**.  
To resume the default settings, run **no default-metric**.

**default-metric number**

**no default-metric**

#### Parameter

Parameter	Description
number	To-be-set route weight, ranging between 1 and 16

#### Default

The corresponding routing cost will be applied for the automatic transfer of each routing protocol.

#### Command mode

Routing configuration mode

#### Explanation

The **default-metric** command is used to set the default routing cost when the route of other routing protocol is guided into the RIP packet. When the **redistribute** command is used to guide the route of other routing protocol, the specific routing cost will not be specified but the default routing cost designated by the **default-metric** command will be guided.

#### Example

In the following example, the router self-managed system 119 adopts the RIP router or the OSPF router, the RIP declaration comes from the OSPF route and the RIP route weight is endowed with 8.

```
router rip
default-metric 8
redistribute ospf 119
```

#### Related command

**redistribute**

**default-information originate**

### 1.1.4 ip rip authentication

To designate the authentication type of the RIP-2 packet, run **ip rip authenticate**. To disable the authentication of the packet, run **no ip rip authentication**.

**ip rip authentication {simple | message-digest}**

**no ip rip authentication**

#### Parameter

Parameter	Description
simple	Plain text authentication
message-digest	MD5 encryption authentication

#### Default

The packet is not authenticated.

#### Command mode

Interface configuration mode

#### Explanation

RIP-1 does not support the authentication.

#### Example

The following example shows that the MD5 encryption authentication mode is adopted on an interface.

```
ip rip authentication message-digest
```

#### Related command

**ip rip password**

**ip rip message-digest-key**

### 1.1.5 ip rip message-digest-key

To activate the authentication of the RIP-2 packet and specify the MD5 authentication mode on an interface, run **ip rip message-digest-key**. To stop the authentication, run **no ip rip message-digest-key**.

**ip rip message-digest-key *key-id* md5 password**

**no ip rip message-digest-key** [*key-id*]

### Parameter

Parameter	Description
key-id	An identifier character
password	Designated encryption key

### Default

The MD5 authentication is invalid.

### Command mode

Interface configuration mode

### Explanation

If no encryption key is configured through the **ip rip message-digest-key key-id md5 password** command, there is no any authentication on the interface.

### Example

The following example shows how to enable the interface to accept and transmit the **mykey** MD5-authenticated packets.

```
ip rip message-digest-key 4 md5 mykey
```

### Related command

ip rip authentication

## 1.1.6 ip rip passive

To disable the transmission of the route update on the interface, run **ip rip passive**. To re-activate the route update, run **no ip rip passive**.

**ip rip passive**

**no ip rip passive**

### Parameter

None

**Default**

The route update is transmitted on the interface.

**Command mode**

Interface configuration mode

**Explanation**

If you cancel the transmission of the route update on an interface, some specific subnet will continue declaring to other interfaces that the route update from other routers to the interface can still be accepted and handled.

**Example**

The following example shows that the RIP packet update will be transmitted to all interfaces belonging to network 172.16.0.0.

```
interface ethernet 1/0
ip address 172.15.0.1 255.255.0.0
ip rip passive
router rip
network 172.16.0.0
```

**Related command**

None

**1.1.7 ip rip password**

To activate the authentication of the RIP-2 packet and specify the plain-text authentication mode on an interface, run **ip rip password**. To cancel the authentication, run **no ip rip password**.

**ip rip password** *password*

**no ip rip password** *password*

**Parameter**

Parameter	Description
password	Designates the encryption key.

**Default**

There is no authentication.

**Command mode**

Interface configuration mode

**Explanation**

If no encryption key is configured through the **ip rip password** command, there is no any authentication on the interface.

**Example**

The following example shows how to enable the interface to accept and transmit the **mykey** text-authenticated packets.

```
ip rip password mykey
```

**Related command**

**ip rip authentication**

**1.1.8 ip rip receive version**

To specify which version of RIP packets are accepted by the interface, run **ip rip receive version**. To follow the global version's regulations, run **no ip rip receive version**.

**ip rip receive version [1] [2]**

**no ip rip receive version**

**Parameter**

Parameter	Description
1	An optional parameter, allowing the interface to accept the RIP packets of version 1
2	An optional parameter, allowing the interface to accept the RIP packets of version 2

**Default**

The RIP-1 packets and the RIP-2 packets are accepted.

**Command mode**

Interface configuration mode

## Explanation

You can use the command to replace the version-designated default function of RIP. The command is only applied to the being-configured interface. The interface can be configured to accept the RIP-1 and RIP-2 packets.

## Example

The following example shows how to enable the interface to accept the RIP-1 and RIP-2 packets:

```
ip rip receive version 1 2
```

The following example shows how to enable the interface to accept the RIP-1 packets:

```
ip rip receive version 1
```

## Related command

**ip rip send version**

**version**

### 1.1.9 ip rip send version

To specify which version of RIP packets are transmitted by the interface, run **ip rip send version**. To follow the global version's regulations, run **no ip rip send version**.

**ip rip send version [ 1 | 2 | compatibility ]**

**no ip rip send version**

## Parameter

Parameter	Description
1	An optional parameter, allowing the interface to transmit the RIP-1 packets
2	An optional parameter, allowing the interface to transmit the RIP-2 packets
compatibility	An optional parameter, allowing the interface to broadcast the RIP-2 packets

## Default

Only the RIP-1 packets are transmitted.

## Command mode

Interface configuration mode



**Explanation**

You can use the command to replace the default actions of RIP. The command is only applied to the being-configured interface. The interface can be configured to accept the RIP-1 and RIP-2 packets.

**Example**

The following example shows how to enable the interface to send the RIP-1 packets:

```
ip rip send version 1
```

The following example shows how to enable the interface to send the RIP-2 packets:

```
ip rip send version 2
```

**Related command**

**ip rip receive version**

**version**

**1.1.10 ip rip split-horizon**

To set whether the horizontal split is used at the transmission of the RIP packets, run **ip rip split-horizon**.

**ip rip split-horizon**

**no ip rip split-horizon**

**Parameter**

None

**Default**

The horizontal media varies with the media.

**Command mode**

Interface configuration mode

**Explanation**

For any interface where the frame relay or SMDS is used, the horizontal split is activated by default; if the interface is configured using the **encapsulation frame-relay** command, the horizontal split is not activated by default.

**Note:** For networks having the X.25 PSN link, the **neighbor** command can make the horizontal split invalid or you can use the **no ip rip split-horizon** command during the configuration. If you make the previous configuration, you have to configure the **no ip rip split-horizon** command on all routers in the multicast group of the network.

If there is horizontal split on the interface, run **ip rip split-horizon** to activate the horizontal split.

**Note:**

In general, you need not modify the default state of the **ip rip split-horizon** command unless you ensure that the change can make the application program declare the route. If the horizontal split is not activated on a serial interface or the interface of the packet-connected switching network, you must disable the horizontal split on all routers and access servers of the network.

## Example

The following example shows how to disable the horizontal split on the serial link which connects the X.25 network.

```
interface serial 1/0
encapsulation x25
no ip rip split-horizon
```

## Related command

**neighbor**

### 1.1.11 neighbor

To define the neighbor router, run **neighbor ip-address**. To disable the neighbor router, run **no neighbor ip-address**.

**neighbor** *ip-address*

**no neighbor** *ip-address*

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor router which exchanges the route information

## Default

No neighbor router is defined.

## Command mode

Routing configuration mode

## Explanation

The **neighbor** command is used to designate the address of the point-fixed transmission, which can meet special needs of the specific non-broadcast network where the transmission cannot be conducted in the broadcast address.

## Example

The following example shows that the RIP update can be transmitted to the designated neighbor:

```
router rip
neighbor 131.108.20.4
```

## Related command

**network**

### 1.1.12 network

To designate the number of the RIP-designated network, run **network**. To cancel the network number, run **no network**.

**network** *network-number* [*<network-mask>*]

**no network** *network-number* [*<network-mask>*]

## Parameter

Parameter	Description
<i>Network-number</i>	IP address of the direct-through network
<i>Network-mask</i>	Mask of the IP address for the direct-through network, which is optional

## Default

No network is designated.

## Command mode

Routing configuration mode

## Explanation

The designated network number cannot contain any information about the subnet. You can designate multiple **network** commands. The RIP update can be transmitted and accepted only on the interfaces of the designated network.

The RIP update can be transmitted to the interfaces of the designated network. If a network connected by an interface is not designated, it will not be declared in any RIP update.

### Example

The following example shows that the RIP is defined as a routing protocol used to connect network 128.99.0.0 and network 192.31.7.0.

```
router rip
network 128.99.0.0
network 192.31.7.0
```

### Related command

**router rip**

#### 1.1.13 offset

To add an offset value for the RIP-learning route weight, run **offset**. To cancel the newly-added offset, run **no offset**.

**offset** {*type number* | \*} {*in* | *out*} *access-list-name* **offset**

**no offset** {*type number* | \*} {*in* | *out*}

### Parameter

Parameter	Description
<b>In</b>	Applies the access control list to the incoming route weight.
<b>Out</b>	Applies the access control list to the outgoing route weight.
<i>access-list-name</i>	Name of the applied standard access control list The number 0 means all access control lists. If the offset is 0, no action will be taken.
<b>offset</b>	Positive offset which is used to match up the route weight of the accessed network
<b>type</b>	Type of the interface (optional)
<i>number</i>	Interface number which is applied to the offset list (optional)

### Default

Invalid state

### Command mode

Routing configuration mode

## Explanation

对路由权值增加一个偏移量。The offset list with interface type and interface number is the expanded one, which has higher priority than the offset list without the interface type and the interface number. Hence, if the expanded offset list and the unexpanded offset list are applied at the same time, the expanded offset will be added to the route weight.

## Example

The following example shows that the router adds 10 to the route offset gained from Ethernet interface 1/0.

```
offset ethernet 1/0 in 21 10
```

### 1.1.14 router rip

To configure the RIP routing process, run **router rip**. To shut down the RIP routing process, run **no router rip**.

```
router rip
```

```
no router rip
```

## Parameter

None

## Default

RIP is not run by default.

## Command mode

Global configuration mode

## Explanation

Only when RIP is started can the routing configuration mode be entered and all global RIP parameter be configured. However, the interface-related parameters can be configured no matter whether the RIP is started or not.

## Example

Start RIP and enter the routing configuration mode.

**Related command****network (RIP)****1.1.15 timers expire**

To regulate the timer of the RIP network, run **timers expire**. To restore the default timer, run **no timers expire**.

**timers expire interval****no timers expire****Parameter**

Parameter	Description
expire	Interval for the route being declared as invalid, which is at least three times of the value of the <b>update</b> parameter. If the update for refreshing the route has not arrived, the route will become the invalid route; the route will enter the <b>stopped</b> state and be identified as <b>inaccessible</b> and <b>unreachable</b> . However, the route can be used to forward the packet. The default value is 180 seconds.

**Default**

The default value of the **expire** parameter is 180 seconds.

**Command mode**

Routing configuration mode

**Explanation**

The basic time-calculated parameter of RIP can be regulated. Because the distributive asynchronous routing algorithm is used on RIP, it is very important to set these time-calculated parameters of all routers and access servers to the same value.

**Note:**

You can check the parameter of the current/default timer using the `show ip rip` command.

**Example**

The following example shows that the route will be declared as unavailable if no information is received from the router within 30 seconds.

**router rip**

**timers expire 30**

#### 1.1.16 timers holddown

To regulate the timer of the RIP network, run **timers holddown second**. To restore the default timer, run **no timers holddown**.

**timers holddown second**

**no timers holddown**

#### Parameter

Parameter	Description
<i>second</i>	Interval for the routing information being constrained When the update packet indicating the route is unreachable is accepted, the route enters the <b>holddown</b> state and then is declared as unreachable. However, the route can still be used to forward the packet. When the <b>holddown</b> time arrives, the route from other sources is accepted and the previous route will be removed from the routing table. The default value is 120 seconds.

#### Default

The default value of the **holddown** parameter is 120 seconds.

#### Command mode

Routing configuration mode

#### Explanation

The basic time-calculated parameter of RIP can be regulated. Because the distributive asynchronous routing algorithm is used on RIP, it is very important to set these time-calculated parameters of all routers and access servers to the same value.

#### Note:

You can check the parameter of the current/default timer using the **show ip rip** command.

#### Example

The following example shows that if no information about the router is received in 30 seconds after a route is declared as unavailable, the route will be deleted from the routing table.

**router rip**

**timers holddown 30****1.1.17 timers update**

To regulate the timer of the RIP network, run **timers update**. To restore the default timer, run **no timers update**.

**timers update update****no timers update****Parameter**

Parameter	Description
update	Basic time-calculated parameter of the router, which is used to specify the transmission interval of route update The default value is 30 seconds.

**Default**

The default value of the **update** parameter is 30 seconds.

**Command mode**

Routing configuration mode

**Explanation**

The basic time-calculated parameter of RIP can be regulated. Because the distributive asynchronous routing algorithm is used on RIP, it is very important to set these time-calculated parameters of all routers and access servers to the same value.

**Note:**

You can check the parameter of the current/default timer using the **show ip rip protocol** command.

**Example**

The following example shows that the RIP update is broadcast every five seconds.

```
router rip
timers update 5
```

**Note:**

If the lifetime of the update is set too small, the low-speed serial link may be congested. However, for the serial link on fast Ethernet and T-1 serial link, the worry does not exist. Meanwhile, if the update includes many routes, the router may take lots of time to handle the update.



### 1.1.18 validate-update-source

To verify the IP address of the router which transmits the RIP update, run **validate-update-source**. To cancel the function, run **no validate-update-source**.

**validate-update-source**

**no validate-update-source**

#### Parameter

There is no parameters or keywords.

#### Default

The function is in the active state.

#### Command mode

Routing configuration mode

#### Explanation

The command is applied only to RIP and IGRP. The software guarantees that the IP address of the router which transmits the route update is the same to that of a network defined by the receiver interface.

The system will conduct the authentication even if the horizontal split is cancelled.

For the unnumbered IP interface, the authentication is not conducted.

#### Example

The following example shows that the router authenticates the source IP address of the incoming RIP update:

```
router rip
 network 128.105.0.0
 no validate-update-source
```

### 1.1.19 version

To set the version of the RIP packet on an interface, run **version**; to resume the default value, run **no version**.

**version {1 | 2}**

**no version**

**Parameter**

Parameter	Description
1	Specifies the version to RIP-1.
2	Specifies the version to RIP-2.

**Default**

According to the configuration of each port, the RIP-1 packets and the RIP-2 packets are accepted, while only the RIP-1 packets are transmitted.

**Command mode**

Routing configuration mode

**Explanation**

After the **no version** command is used, the available RIP version can be specified on the interface, and then the **ip rip receive version** command and the **ip rip send version** command can be used; otherwise, the RIP packet will be transmitted and accepted according to the globally-configured version.

**Example**

The following example shows that the software transmits and accepts the RIP-2 packets:

```
version 2
```

**Related command**

**ip rip receive version**

**ip rip send version**

**1.1.20 distance**

To set the management distance of the RIP route, run the following command.

Distance ***weight <address mask <access-list-name>>***

**Parameter**

Parameter	Description
weight	Management distance, ranging between 1 and 255. You are recommended to set the value range between 10 and 255 (values from 0 to 9 reserved). If the parameter is used alone, the router will take it as the default.

	management distance if the router does not have relative regulations about a routing information. The router whose management distance is 255 will not be added to the routing table.
address	An optional parameter, meaning the source IP address in the <b>aa.bb.cc.dd</b> form
mask	An optional parameter, meaning the mask of the IP address in the <b>aa.bb.cc.dd</b> form. If one bit is 0, the router will omit the value of the corresponding bit in the address.
access-list-name	(optional) name of the standard access control list

**Default**

120

**Command mode**

EXEC

**Explanation**

Management distance is an integer from 0 to 255. In general, the bigger the value is, the less incredible the value is. If the **access-list-name** parameter is used in the command, the access control list is applied when a route is added to the routing table. In this way, you can filter the paths of some network according to the address of the router provided by the routing information.

**Example**

The following example shows that the distance of the route received from network 192.1.1.0/24 is set to 100.

```
router rip
distance 100 192.1.1.0 255.255.255.0
```

**1.1.21 filter**

To filter the received or transmitted RIP routes, run **filter**.

**filter \* in access-list** {*access-list-name*}

**filter \* in gateway** {*access-list-name*}

**filter \* in prefix** {*prefix-list-name*}

**filter type number in access-list** {*access-list-name*}

**filter type number in gateway** {*access-list-name*}

**filter type number in prefix** {*prefix-list-name*}

**no filter \* in**

**no filter type *number* in**

**filter \* out access-list {*access-list-name*}**

**filter \* out gateway {*access-list-name*}**

**filter \* out prefix { *prefix-list-name*}**

**filter type *number* out access-list {*access-list-name*}**

**filter type *number* out gateway {*access-list-name*}**

**filter type *number* out prefix {*prefix-list-name*}**

**no filter \* out**

**no filter type *number* out**

### Parameter

Parameter	Description
<i>access-list-name</i>	Name of the standard IP access control list, which defines what type of networks will be accepted and what type of networks will be limited during the route update
<i>prefix-list-name</i>	Name of the standard IP prefix list, which defines what type of networks will be accepted and what type of networks will be limited during the route update
<b>in/out</b>	Incoming/outgoing route update access list
<b>type</b>	Type of the interface (optional)
<i>number</i>	An optional parameter, which specifies on which interface the incoming/outgoing update access list is applied. If the interface is not specified, the incoming/outgoing update access list will be applied on all incoming/outgoing interfaces.

### Default

The received or transmitted RIP routes are not filtered.

### Command mode

EXEC

### Explanation

The command is used to filter the received or transmitted RIP routes. When the filtration list is configured for the dynamic routing protocol, if the access list is used to filter the routes, the standard access list is required.

## Example

The following example shows that the **10.0.0.0/8** route transmitted from port s2/1 is filtered.

```
router rip
filter s2/1 out access-list mylist
ip access-list standard mylist
deny 10.0.0.0 255.0.0.0
```

### 1.1.22 maximum-count

To configure the maximum number of routes for the local RIP routing table, run **maximum-count *number***. To resume the default settings, run **no maximum-count *number***.

**maximum-count** *number*

**no maximum-count**

## Parameter

Parameter	Description
<i>number</i>	To-be-set maximum number of routes, ranging between 512 and 4096

## Default

1024

## Command mode

Routing configuration mode

## Explanation

The **maximum-count** command is used to configure the maximum number of routes for the local RIP routing table. When the number of routes in the local routing table exceeds the maximum value, the route will not be added to the routing table.

## Example

The following example shows that the maximum number of routes in the local routing table is set to 2000.

```
router rip
maximum-count 2000
```

**Related command**

None

**1.1.23 show ip rip**

To display the main RIP information, run the following command.

**show ip rip**

**Parameter**

None

**Default**

None

**Command mode**

EXEC

**Explanation**

According to the output information, you can check the configuration information about the current RIP.

**Example**

The following example shows that the information about RIP configuration is displayed:

```
router#show ip rip
```

```
RIP protocol: Enabled
```

```
Decided on the interface version control
```

```
AUTO-SUMMARY: Yes
```

```
Update: 30, Expire: 180, Holddown: 120
```

```
Distance: 120
```

```
default-metric: 1
```

The fields in the previous example are explained in the following table:

Field	Description
Enabled	State of the protocol
Distance	Current management distance
version	Version of the current protocol
AUTO-SUMMARY	Specifies the auto-summary function.

Update	Transmission interval of the update packet
Holddown	Time of route running
Expire	Time of route aging
RIP default-metric	Default cost when the <b>redistribute</b> command is run

### 1.1.24 show ip rip database

To display all RIP routing information, run the following command:

```
show ip rip database
```

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check all RIP routing information.

#### Example

The following example shows that all RIP routing information is displayed.

```
router#show ip rip database
1.0.0.0/8  auto-summary
1.1.1.0/24  directly connected  Loopback1
100.0.0.0/8  via 192.1.1.2 00:00:02
192.1.1.0/24  directly connected  Serial2/1
192.1.1.0/24  auto-summary
```

The fields in the previous example are explained in the following table:

Field	Description
Network-number/network-mask	RIP route
Summary/connected/via gateway	Type of the corresponding RIP route

interface	Interface which corresponds to the direct-through RIP route and the summary route
time	Refreshed time

### 1.1.25 show ip rip protocol

To display the information about RIP configuration, run the following command:

**show ip rip protocol**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the configuration information about the current RIP.

#### Example

The following example shows that the information about RIP configuration is displayed:

```
router#show ip rip protocol
RIP is Active
  Sending updates every 30 seconds, next due in 30 seconds
  Invalid after 180 seconds, holddown 120
  update filter list for all interfaces is:
  update offset list for all interfaces is:
  Redistributing:
  redistribute connect
  Default version control: send version 1, receive version 1 2
  Interface      Send      Recv
  Async0/0       1         1 2
  FastEthernet0/0 1         1 2
  Serial1/0       1         1 2
  Ethernet1/1     1         1 2
  Serial2/0       1         1 2
```



```

Serial2/1      1      1 2
Loopback1      1      1 2
Automatic network summarization is in effect
Routing for Networks:
  174.168.0.0/16
Distance: 120 (default is 120)

```

### 1.1.26 debug ip rip database

To monitor the route event of RIP, run the following command:

**debug ip rip database**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the output information, you can check the configuration information about the current RIP.

#### Example

The following example shows that the route event of RIP is monitored.

```
router# debug ip rip database
```

```
RIP-DB: Adding summary route 192.1.1.0/24 <metric 0> to RIP database
```

The fields in the previous example are explained in the following table:

Field	Description
summary	Route type which is added to the routing table
192.1.1.0/24	Route which is added to the routing table
<metric 0>	Value of the route's metric

### 1.1.27 debug ip rip protocol

To monitor the RIP packet, run the following command:

**debug ip rip protocol****Parameter**

None

**Default**

None

**Command mode**

EXEC

**Explanation**

According to the command exported by the command, you can check the content of the received and transmitted packets of the current RIP.

**Example**

The following example shows that the RIP packet is monitored:

```
router# debug ip rip protocol
RIP: send to 255.255.255.255 via Loopback1
vers 1, CMD_RESPONSE, length 24
192.1.1.0/0 via 0.0.0.0 metric 1.
```

If version 2 is run, the following information will be displayed:

```
RIP: send to 255.255.255.255 via Loopback1
vers 2, CMD_RESPONSE, length 24
192.1.1.0/24 via 0.0.0.0 metric 1
```

The fields in the previous example are explained in the following table:

Domain	Description
Send/Recv	Received or transmitted packet
to 255.255.255.255	Destination address of the IP packet
via Loopback1	Port where the packet is transmitted or received
vers 2	Version of the transmitted or received packet
CMD_RESPONSE/ CMD_REQUEST	Type of the packet
length 24	Length of the message
192.1.1.0/24	Destination address shown in the routing information
via 0.0.0.0	Address of the next hop

metric	Cost of the route
--------	-------------------

## Chapter 2 BEIGRP Configuration Commands

### 2.1 BEIGRP Configuration Commands

BEIGRP configuration commands include:

auto-summary

clear ip beigrp neighbors

debug ip beigrp

debug ip beigrp fsm

debug ip beigrp neighbours

debug ip beigrp packet

debug ip beigrp transmit

default-metric

distance

filter

beigrp log-neighbor-changes

beigrp router-id

ip beigrp bandwidth-percent

ip beigrp hello-interval

ip beigrp hold-time

ip beigrp passive

ip beigrp split-horizon

ip beigrp summary-address

metric weights

network

offset

redistribute

router beigrp

show ip beigrp interface  
show ip beigrp neighbors  
show ip beigrp protocol  
show ip beigrp topology  
show ip beigrp traffic

### 2.1.1 auto-summary

To automatically summarize the BEIGRP routes, run **auto-summary**. By default, the BEIGRP is automatically summarized. To shut down the automatic summary function and notify each route to its neighbors, run **no auto-summary**.

**auto-summary**

**no auto-summary**

#### Parameter

None

#### Default

By default, the route will be automatically summarized.

#### Command mode

Routing configuration mode

#### Explanation

In the current BEIGRP version, the route summary is close to the **network** command. The route summary must following the following rules:

When a BEIGRP process defines multiple networks, if one subnet of a network is in the BEIGRP topology table, the summary route of the defined network will be generated.

The established summary route is pointed to the NULL0 interface, which has the minimum distance of all subnets. The summary route is inserted into the main IP routing table and the management distance is 5 (cannot be configured).

When the update is transmitted to the neighbors in different main IP networks, the subnet of regulation-1 summary and regulation-2 summary will be cancelled and only the summary route is transmitted.

The subnets of any network which are not listed in the BEIGRP process definition are not summarized.

**Related command****ip beigrp summary-address****network****2.1.2 clear ip beigrp neighbors**

To cancel the neighborhood with current neighbors, run the following command in EXEC mode:

**clear ip beigrp** [*as-number*] **neighbors** [*ip-address* | *interface-type interface-number*]**Parameter**

Parameter	Description
<i>as_number</i>	An optional parameter, which means the number of the automatic system of the neighbor
<i>ip-address</i>	An optional parameter, which means the address of the BEIGRP neighbor
<i>interface</i>	An optional parameter, which means the name of the access list After the parameter is entered, the neighborhood of all neighbors on the interface will be reset.

**Default**

None

**Command mode**

EXEC mode

**Explanation**

If no parameter is designated, all BEIGRP neighbors will be reset.

The neighborhood of one or many neighbors will be reset after the command is run, and hence the route operation will be triggered. When too many routes are affected, the route turbulence may occur. In this case, a certain time is needed for re-convergence. It is recommended that the command is not used unless the network is in the debugging state.

**Example**

```
clear ip beigrp ethernet1/1
```

The example shows that all neighbors on interface Ethernet 1/1 will be cancelled and the relative routes will be re-triggered for recalculation.

### 2.1.3 debug ip beigrp

To track the information about the BEIGRP protocol, run **debug ip beigrp** in EXEC mode.

**debug ip beigrp**

**no debug ip beigrp**

#### Parameter

None

#### Default

None

#### Command mode

EXEC mode

#### Explanation

The command can help detecting the network troubles.

#### Example

```
clear ip beigrp ethernet1/1
```

The example shows that all neighbors on interface Ethernet 1/1 will be cancelled and the relative routes will be re-triggered for recalculation.

### 2.1.4 debug ip beigrp fsm

To track the state change of the BEIGRP DUAL algorithm, run **debug ip beigrp fsm** [**detail**] in EXEC mode.

**debug ip beigrp fsm** [**detail**]

#### Parameter

Parameter	Description
detail	An optional parameter to display the detailed information

#### Default

None

**Command mode**

EXEC mode

**Explanation**

The command can help detecting the network troubles.

**Related command**

**debug ip beigrp packet**

**2.1.5 debug ip beigrp neighbors**

To track the creation and deletion of the BEIGRP neighbor, run the following command in EXEC mode.

**debug ip beigrp neighbors**

**Parameter**

None

**Default**

None

**Command mode**

EXEC mode

**Explanation**

The command can help detecting the network troubles.

**Example**

```
TestC#debug ip beigrp neighbors
BEIGRP: Neighbor 192.168.20.141 went down on Ethernet1/1 for peer restarted.
BEIGRP: Neighbor(192.168.20.141) not yet found.
BEIGRP: Neighbor(192.168.20.141) not yet found.
BEIGRP: New neighbor 192.168.20.141
BEIGRP: Neighbor 202.117.80.143 went down on Ethernet2/1 for manually cleared.
BEIGRP: Neighbor 192.168.20.141 went down on Ethernet1/1 for manually cleared.
BEIGRP: New neighbor 192.168.20.204
BEIGRP: New neighbor 202.117.80.143
```



BEIGRP: New neighbor 192.168.20.141

### Related command

**debug ip beigrp fsm**

#### 2.1.6 debug ip beigrp packet

To track the information about the transmission and reception of the BEIGRP packet, run **debug ip beigrp packets [ack | hello | query | reply | retry | terse | update]** in EXEC mode.

**debug ip beigrp packets [ack | hello | query | reply | retry | terse | update]**

**no debug ip beigrp packets [ack | hello | query | reply | retry | terse | update]**

### Parameter

Parameter	Description
<b>ack</b>	An optional parameter, which is used to track the ACK packet
<b>hello</b>	An optional parameter, which is used to track the HELLO packet
<b>query</b>	An optional parameter, which is used to track the QUERY packet
<b>reply</b>	An optional parameter, which is used to track the REPLY packet
<b>retry</b>	An optional parameter, which is used to track the RESENT packet
<b>terse</b>	An optional parameter, which is used to track all packets except the HELLO packet
<b>update</b>	An optional parameter, which is used to track the UPDATE packet

### Default

None

### Command mode

EXEC mode

### Explanation

The command can help detecting the network troubles.

### Example

```
router#debug ip beigrp packet
```

```
BEIGRP: Send HELLO packet to 224.0.0.10 via Ethernet2/1 with Ack 0/0
```

```
BEIGRP: Receive ACK packet from 192.168.20.141 via Ethernet1/1 with Ack 0/54
```

BEIGRP: Receive HELLO packet from 202.117.80.143 via Ethernet2/1 with Ack 0/0  
 BEIGRP: Receive UPDATE packet from 192.168.20.204 via Ethernet1/1 with Ack 142/0  
 BEIGRP: Send HELLO packet to 192.168.20.204 via Ethernet1/1 with Ack 0/142  
 BEIGRP: Receive HELLO packet from 192.168.20.141 via Ethernet1/1 with Ack 0/0  
 BEIGRP: Receive HELLO packet from 192.168.20.204 via Ethernet1/1 with Ack 0/0  
 BEIGRP: Receive QUERY packet from 192.168.20.204 via Ethernet1/1 with Ack 143/0  
 BEIGRP: Send HELLO packet to 192.168.20.204 via Ethernet1/1 with Ack 0/143  
 BEIGRP: Send REPLY packet to 192.168.20.204 via Ethernet1/1 with Ack 55/143  
 BEIGRP: Send UPDATE packet to 224.0.0.10 via Ethernet2/1 with Ack 57/0  
 BEIGRP: Receive ACK packet from 192.168.20.204 via Ethernet1/1 with Ack 0/55  
 BEIGRP: resend UPDATE packet for neighbor 192.168.20.204 with retry num 1.  
 BEIGRP: Receive ACK packet from 202.117.80.143 via Ethernet2/1 with Ack 0/57  
 BEIGRP: Send UPDATE packet to 202.117.80.143 via Ethernet2/1 with Ack 57/77  
 BEIGRP: Send UPDATE packet to 224.0.0.10 via Ethernet1/1 with Ack 56/0  
 BEIGRP: Receive ACK packet from 192.168.20.204 via Ethernet1/1 with Ack 0/56  
 BEIGRP: Send UPDATE packet to 192.168.20.141 via Ethernet1/1 with Ack 56/88  
 BEIGRP: Send UPDATE packet to 192.168.20.204 via Ethernet1/1 with Ack 56/143  
 BEIGRP: Receive UPDATE packet from 202.117.80.143 via Ethernet2/1 with Ack 79/0  
 BEIGRP: Send HELLO packet to 202.117.80.143 via Ethernet2/1 with Ack 0/79  
 BEIGRP: Receive ACK packet from 192.168.20.204 via Ethernet1/1 with Ack 0/56  
 BEIGRP: Send QUERY packet to 224.0.0.10 via Ethernet1/1 with Ack 60/0  
 BEIGRP: Send UPDATE packet to 224.0.0.10 via Ethernet1/1 with Ack 61/0

Field	Description
Recv / Send / Enqueueing	Receives, transmits the packets or adds the packet to the transmission queue.
HELLO / UPDATE / QUERY / ACK	Type of the received /transmitted packet
192.1.1.1	IP address of the neighbor which transmits the packet
Serial1/2	Incoming/outgoing interface of the packet
Ack 56/88	Response sequence number of the packet or the sequence number of the neighbor's packet

## Related command

**debug ip beigrp fsm**

### 2.1.7 debug ip beigrp transmit

To track the events of handling the BEIGRP packet, run **debug ip beigrp transmit** in EXEC mode.

**debug ip beigrp transmit [ack | build | link | packetize | peerdn / startup]**

**no debug ip beigrp transmit [ack | build | link | packetize | peerdn / startup]**

**Parameter**

Parameter	Description
<b>ack</b>	An optional parameter which is used to track the event
<b>build</b>	An optional parameter which is used to track the BUILD event
<b>link</b>	An optional parameter which is used to track the LINK event
<b>packetize</b>	An optional parameter which is used to track the PACKETIZE event
<b>peerdown</b>	An optional parameter which is used to track the PEERDOWN event
<b>startup</b>	An optional parameter which is used to track the STARTUP event

**Default**

None

**Command mode**

EXEC mode

**Explanation**

The command can help detecting the network troubles.

**Related command****debug ip beigrp fsm****2.1.8 default-metric**

To reset the default vector distance of BEIGRP, run **default-metric**. To resume the previous default value, run **no default-metric**.

**default-metric *bandwidth delay reliability loading mtu*****no default-metric****Parameter**

Parameter	Description
<b>bandwidth</b>	Default bandwidth
<b>delay</b>	Default delay of the interface
<b>reliability</b>	Default reliability of the interface
<b>loading</b>	Default loading of the interface

<b>mtu</b>	Default maximum transmission unit of the interface
------------	--

## Default

Bandwidth: 128kpbs

delay:2000 (10ms)

reliability: 255 (255 means 100%)

loading: 255 (255 means 100%)

mtu: 1500

## Command mode

Routing configuration mode

## Explanation

The command is used together with the **redistribute** command to specify the initial distance vector of other routing protocols distributed to the BEIGRP topology table. After the command is configured, the relative routes previously redistributed to the BEIGRP topology table will be re-calculated.

When the static/direct-through/ BEIGRP route is forwarded, the default-metric command need not be configured; while the **default-metric** command has to be configured before routes of other protocols are forwarded.

## Example

```
default-metric 200 1000 100 200 1500
```

The previous example shows how to set the bandwidth in the default distance vector to 200 kbps, the delay to 1000 (10ms), the reliability to 100, the loading to 200 and the MTU to 1500.

## Related command

**redistribute**

### 2.1.9 distance

To modify the management distance of the BEIGRP route (including management distances of the internal/external BEIGRP routes), run **distance**. You can run **no distance** to resume the default value of the BEIGRP management distance.

**distance beigrp *internal-distance external-distance***

**no distance beigrp**

**distance weight** *ip-address ip-address-mask [ip-access-list]*

**no distance weight** *ip-address ip-address-mask [ip-access-list]*

### Parameter

Parameter	Description
<i>internal-distance</i>	Management distance of the internal BEIGRP route, ranging between 1 and 255
<i>external-distance</i>	Management distance of the external BEIGRP route, ranging between 1 and 255
<i>ip-address</i>	IP address of the BEIGRP neighbor
<i>ip-address-mask</i>	Mask of the IP address for the BEIGRP neighbor
<i>ip-access-list</i>	Name of the BEIGRP neighbor access list

### Default

internal-distance: 90

external-distance: 170

### Command mode

Routing configuration mode

### Explanation

The management distance means the priority level of the routes of different routing protocols. Hence, you can finally affect the routing strategy of the router by modifying the BEIGRP management distance and thus different requirements of the users can be met.

You are recommended to use the standard access list when configuring the filtration list; if the expanded access list is configured, the configured access list has no function.

### Example

```
router beigrp 2
network 192.10.0.0 255.255.0.0
distance beigrp 100 200
distance 110 192.31.7.0 255.255.255.0
distance 220 128.88.1.0 255.255.255.0
```

The previous example shows that the management distance of the BEIGRP internal route is set to 100 and that of the BEIGRP external route is set 200. Meanwhile, the management distance of the route of the gateway address which lies in network

segment 192.31.7.0/24 is set to 110 and that of the route of the gateway address which lies in network segment 128.88.1.0/24 is set to 220.

## Related command

**show ip protocol**

### 2.1.10 filter

To filter the learned routes or transmitted routes on the designated port or filter the routes accurately using the access list or the prefix list, run **filter {*interface-type interface-number* | \*} {in | out} {access-list *access-list-name* | gateway *access-list-name* / prefix-list *prefix-list-name*}**. To cancel filtering the routes, run **no filter**.

**filter** {*interface-type interface-number* | \*} {in | out} {**access-list** *access-list-name* | **gateway** *access-list-name* / **prefix-list** *prefix-list-name*}

**no filter** {*interface-type interface-number* | \*} {in | out} {**access-list** *access-list-name* | **gateway** *access-list-name* / **prefix-list** *prefix-list-name*}

## Parameter

Parameter	Description
interface-type interface-number	Type of the interface and the number of the interface
*	All interfaces
in	Applies the access list to the incoming route update.
out	Applies the access control list to the outgoing route update.
access-list	Filters the route using the standard access control list and defines what type of the network is transmitted and what type of the network is limited in the route update.
gateway	Filters the gateway of the route using the standard access control list.
access-list-name	Name or number of the standard IP access control list
prefix-list	Filters the routes through the prefix list.
prefix-list-name:	Name of the standard IP prefix list, which defines what type of networks will be accepted and what type of networks will be limited during the route update

## Default

None

**Command mode**

Routing configuration mode

**Explanation**

You are recommended to use the standard access list when configuring the filtration list; if the expanded access list is configured, the configured access list has no function.

**Example**

The following example shows that only network 131.108.0.0 can be declared by the BEIGRP routing process:

```
access-list 1 permit 131.108.0.0
access-list 1 deny 0.0.0.0 255.255.255.255
router beigrp 64
network 131.108.0.0
filter * out 1
```

**2.1.11 beigrp log-neighbor-changes**

To write the neighbor's change to the log, run **beigrp log-neighbor-changes**. To cancel the log's record, run **no beigrp log-neighbor-changes**.

**beigrp log-neighbor-changes**

**no beigrp log-neighbor-changes**

**Parameter**

None

**Default**

disabled

**Command mode**

Routing configuration mode

**2.1.12 beigrp router-id**

To designate the identifier of the router, run **beigrp router-id *ip-address***. To cancel the identifier settings of the router, run **no beigrp router-id *ip-address***.

**beigrp router-id *ip-address***

**no beigrp router-id****Parameter**

Parameter	Description
<i>ip-address</i>	Identifier of the BEIGRP router in the form of the IP address

**Default**

BEIGRP automatically chooses the identifier of the router. If the loopback port exists, the maximum address of the loopback port will be used as the identifier of the router, or the maximum address of the direct-through port will be used as the identifier of the router.

**Command mode**

Routing configuration mode

**2.1.13 ip beigrp bandwidth-percent**

To specify the bandwidth rate of the path suitable for the BEIGRP packet interaction, run **ip beigrp bandwidth-percent percent**. You can run **no ip beigrp bandwidth-percent percent** to resume the default value.

**ip beigrp bandwidth-percent** *percent*

**no ip beigrp bandwidth-percent** *percent*

**Parameter**

Parameter	Description
<i>percent</i>	Bandwidth occupancy percent

**Default**

50%

**Command mode**

Interface configuration mode

**Explanation**

For the low-speed line, you can regulate the configuration of the command to limit the bandwidth rate used by BEIGRP, preventing BEIGRP from affecting the data transmission.



**Example**

```
interface Ethernet1/1
ip beigrp bandwidth-percent 100
```

The previous commands are used to allow BEIGRP to use all bandwidth of the interface.

**Related command**

bandwidth

**2.1.14 ip beigrp hello-interval**

To configure the transmission interval of the HELLO packet on the BEIGRP interface, run **ip beigrp hello-interval *seconds***.

**ip beigrp hello-interval *seconds***

**no ip beigrp hello-interval *seconds***

**Parameter**

Parameter	Description
<i>second</i>	Transmission interval of the HELLO packet, whose unit is second

**Default**

5 seconds

**Command mode**

Interface configuration mode

**Explanation****Example**

The following example shows that the interval for transmit the HELLO packet on interface Ethernet 1/1 to 20 seconds.

```
interface Ethernet1/1
ip beigrp hello-interval 20
```

**Related command**

**ip beigrp hold-time**

### 2.1.15 ip beigrp hold-time

To configure the timeout of the neighbor on the BEIGRP interface, run **ip beigrp hold-time *seconds***. To resume the default value, run **no ip beigrp hold-time *seconds***.

**ip beigrp hold-time *seconds***

**no ip beigrp hold-time *seconds***

#### Parameter

Parameter	Description
<i>second</i>	Neighbor's timeout termination time if the neighbor does not receive any BEIGRP packet, whose unit is second

#### Default

15 seconds

#### Command mode

Interface configuration mode

#### Explanation

#### Example

The following example shows that the hold-time of the neighbor on interface Ethernet 1/1 is set to 60 seconds.

```
interface Ethernet1/1
ip beigrp hold-time 60
```

#### Related command

**ip beigrp hello-interval**

### 2.1.16 ip beigrp passive

To enable an interface not to exchange the information about the BEIGRP route update, run **ip beigrp passive**. To resume its default settings, run **no ip beigrp passive**.

**ip beigrp passive**

**no ip beigrp passive**

**Parameter**

None

**Default**

The interface is not in **passive** mode.

**Command mode**

Interface configuration mode

**Explanation**

If an interface is set to the **passive** state, the interface then does not receive any route update and no neighborhood between the interface and other reachable neighbors will be established. However, a direct-through route generated by the interface will be broadcasted through other BEIGRP interfaces.

**Example**

The following example shows that interface Ethernet 1/1 is set to a passive interface.

```
interface ethernet1/1
ip beigrp passive
```

**2.1.17 ip beigrp split-horizon**

To activate the horizontal split of the BEIGRP process on an interface, run **ip beigrp split-horizon**. To disable the horizontal split of the BEIGRP process, run **no ip beigrp split-horizon**.

```
ip beigrp split-horizon
no ip beigrp split-horizon
```

**Parameter**

None

**Default**

The horizontal split of the BEIGRP process is in the active state.

**Command mode**

Interface configuration mode

## Explanation

The command is used to prevent the route loopback; hence, the horizontal split must be done after you ensure that no bad results will be generated.

## Example

```
interface Ethernet1/1
no ip beigrp split-horizon
```

The following example shows that the horizontal split on interface ethernet1/1 of the router is disabled.

### 2.1.18 ip beigrp summary-address

To summarize the routes transmitted from an interface, run **ip beigrp summary-address *as\_number* *address* *mask***. To disable the route summary on the interface, run **no ip beigrp summary-address *as\_number* *address* *mask***.

**ip beigrp summary-address *as\_number* *address* *mask***

**no ip beigrp summary-address *as\_number* *address* *mask***

## Parameter

Parameter	Description
<i>as_number</i>	Automatic BEIGRP system number in the route summary configuration
<i>address</i>	Destination network segment of the summary route
<i>mask</i>	Network mask of the summary route

## Default

None

## Command mode

Interface configuration mode

## Explanation

When the route summary is conducted on an interface, follow the following regulations:

- When the route summary is configured on an interface, if one subnet of a network is in the BEIGRP topology table, the summary route of the defined network will be generated.

- The summary route is pointed to the Null0 interface, having the minimum distance of each detailed route. The summary route is inserted into the main IP routing table and the management distance is 5.
- When the route update is transmitted on the interface where the route summary range is configured, the detailed route belonging to the summary network segment is cancelled. The updates transmitted to other interfaces are not affected.

### Example

```
interface Ethernet1/1
ip beigrp summary-address 100 12.1.0.0 255.255.0.0
```

In the previous example, all detailed routes belonging to network segment 12.1.0.0/16 will not be broadcasted from interface ethernet1/1; a summary route will replace all detailed routes and the destination network segment is 12.1.0.0/16.

### Related command

**auto-summary**

#### 2.1.19 metric weights

To change the index for BEIGRP to calculate the routing composite distance, run **metric weights *k1 k2 k3 k4 k5***. You can run **no metric weights** to resume the default value.

**metric weights *k1 k2 k3 k4 k5***

**no metric weights**

### Parameter

Parameter	Description
<b>k1,k2,k3,k4,k5</b>	Five constant coefficients which can transfer the vector distance of a route to a scalar value

### Default

*k1*: 1

*k2*: 0

*k3*: 1

*k4*: 0

*k5*: 0

**Command mode**

Routing configuration mode

**Explanation**

The following two steps are taken to transfer the vector distance to the scalar distance:

**Step 1**

Composite metric =  $K1 \cdot BW \cdot 256 + K2 \cdot BW / (256 - \text{load}) + K3 \cdot DLY \cdot 256$ ,

BW            10Gbps/bandwidth

DLY           delay, 10ms

**Step 2 (used only when K5 is not zero)**

Composite metric = Composite metric \*  $K5 / (\text{reliability} + K4)$

K2, K4 and K5 are legacies of IGRP which are used for being compatible with the EIGRP protocol of Cisco. In general, the parameters **load** and **reliability** are not used to calculate the composite distance. Hence, you are not recommended to modify the default values of K2, K4 and K5, avoiding unexpected results when the routing strategy is conducted.

**Example**

```
router beigrp 2
network 131.108.0.0 255.255.0.0
metric weights 2 0 2 0 0
```

**Related command**

**bandwidth**

**delay**

**2.1.20 network**

To designate a network segment to run the BEIGRP protocol, run **network network-number [netmask]**. To disable the dynamic BEIGRP route running on the network, run **no network network-number [netmask]**.

**network** *network-number* [*netmask*]

**no network** *network-number* [*netmask*]

**Parameter**

Parameter	Description
<i>network-number</i>	Address of the network segment

<i>netmask</i>	Mask of the network
----------------	---------------------

**Default**

None

**Command mode**

Routing configuration mode

**Explanation**

You can configure multiple **network** commands at the same time to enable the dynamic BEIGRP protocol to run on multiple network segments simultaneously. If the net mask is not configured, the natural mask will be used.

**Example**

```
router beigrp 2
network 131.108.0.0 255.255.0.0
network 122.11.2.0
```

**Related command****router beigrp****2.1.21 offset**

To add an offset value for the incoming/outgoing BEIGRP route weight, run **offset**. To cancel the newly-added offset, run **no offset**.

**offset** {*type number* | \*} {*in* | *out*} *access-list-name* **offset**

**no offset** {*type number* | \*} {*in* | *out*}

**Parameter**

Parameter	Description
In	Applies the access control list to the incoming route weight.
Out	Applies the access control list to the outgoing route weight.
access-list-name	Name of the applied standard access control list
Offset	Positive offset which is used to match up the route weight of the accessed network
Type	Type
Number	Interface number which is applied to the offset list (optional)

**Default**

None

**Command mode**

Routing configuration mode

**Explanation**

The offset list with interface type and interface number is the expanded one, which has higher priority than the offset list without the interface type and the interface number. Hence, if the expanded offset list and the unexpanded offset list are applied at the same time, the expanded offset will be added to the route weight.

Because BEIGRP is a vector distance, the offset is added to the delay of the interface.

You are recommended to use the standard access list when configuring the filtration list; if the expanded access list is configured, the configured access list has no function.

**Example**

In the following example, the router adds an offset value, 10, to the BEIGRP route which is suitable to ACL21.

```
offset * out 21 10
```

The following example shows that the router adds an offset value, 10, to the BEIGRP route which gained from Ethernet interface 0.

```
offset e0/0 in 21 10
```

**Related command**

**ip access-list**

**2.1.22 redistribute**

To forward the routes of other routing protocols or the routes of other BEIGRP processes to the routing table of the local BEIGRP process, run **redistribute *protocol* [*process*] route-map *name***.

**redistribute *protocol* [*process*] route-map *name***

**redistribute *protocol* [*process*]**

**Parameter**

Parameter	Description
-----------	-------------



protocol	Source protocol for redistributing the routes, which can be BGP, OSPF, static, connected or RIP protocol
<i>process</i>	An optional parameter which is a 16-bit self-managed system number for BGP or BIGP For OSPF, the parameter is an ID of the OSPF process whose routing key is redistributed. In this way, the routing process is identified. It is a non-zero decimal number. For RIP, the process identifier <b>process</b> is not required.
route-map	An optional parameter which enables the route mapping to filter the routes imported to the current routing protocol from the source protocol. If the parameter is not given, all routes will be redistributed. If the parameter is given but the identifier of the route mapping is not, no route will be imported.
name	A character string stands for the name of the route mapping

**Default**

None

**Command mode**

BEIGRP routing configuration mode

**Explanation**

You must configure the **default-metric** parameter for the routes except the straight-through routes, static routes and routes of other BEIGRP processes.

**Example**

```
default-metric 64 250 255 255 1500
redistribute ospf 1
```

**2.1.23 router beigrp**

To add a BEIGRP process, run **router beigrp**. To cancel the process, run **no router beigrp**.

**router beigrp** *autonomous-system-number*

**no router beigrp** *autonomous-system-number*

**Parameter**

Parameter	Description
autonomous-system-n	Number of the autonomous system, which is used to distinguish the

umber	BEIGRP processes
-------	------------------

**Default**

None

**Command mode**

Global configuration mode

**Explanation**

You can run the command to run multiple BEIGRP processes simultaneously.

**Example**

The following example shows how to add a BEIGRP process whose autonomous system's number is 30:

```
router beigrp 30
```

**Related command**

**network**

**2.1.24 show ip beigrp interface**

To display the states of all BEIGRP neighbors, run the following command:

**show ip beigrp interfaces** [**interface-type** *interface-number*] [*as-number*]

**Parameter**

Parameter	Description
as-number	Number of the autonomous system If the parameter is designated, only the neighbors of the BEIGRP process will be displayed.
interface	Name of the interface If the parameter is designated, only the neighbors on the interface will be displayed.

**Default**

None

**Command mode**

EXEC or global configuration mode

**Explanation**

You can run the command to check the state of the interface for the dynamic BEIGRP route.

**Related command**

**show ip beigrp topology**

**2.1.25 show ip beigrp neighbors**

To display the states of all BEIGRP neighbors, run the following command:

**show ip beigrp neighbors** [*interface-type interface-number*] [*as-number*] [**detail**]

**Parameter**

Parameter	Description
<i>as-number</i>	Number of the autonomous system  If the parameter is designated, only the neighbor of the BEIGRP process will be displayed.
<i>interface</i>	Name of the interface  If the parameter is designated, only the neighbors on the interface will be displayed.
<b>detail</b>	It is used to display the detailed information about the neighbor.

**Default**

None

**Command mode**

EXEC or global configuration mode

**Explanation**

Through the command, you can check the information about neighbors, which neighbors are newly added, which neighbors are cancelled and the state of each neighbor, helping you to detect the network faults.

## Example

Router# show ip beigrp neighbors

Information of BEIGRP neighbors with AS 1024

```
Address      interface  hold  uptime  Q_cnt  Seq
192.168.20.204 Ethernet1/1  15   00:08:06  0    159
202.117.80.143 Ethernet2/1  10   00:08:05  0    100
192.168.20.141 Ethernet1/1  12   00:07:38  0    254
```

Domain	Description
AS 64	Number of the autonomous system
Address	IP address of the neighbor
Interface	Local interface of the neighbor
Hold	Time in which the local terminal has not received the BEIGRP packets from the neighbor
Uptime	Lasting time of neighborhood relation from its establishment to the present
Q Count	Number of the queued packets which will be transmitted to the neighbor
Seq	Latest sequence number received from the neighbor

## Related command

**show ip beigrp topology**

### 2.1.26 show ip beigrp protocol

To display the parameters and the statistics information about the BEIGRP process, run the following command:

**show ip beigrp protocols** [*as-number*]

## Parameter

Parameter	Description
<i>as-number</i>	Number of the autonomous system. If the parameter is designated, only parameters and the statistics information about the BEIGRP process will be displayed.

## Command mode

EXEC or global configuration mode

## Explanation

You can run the command to check the BEIGRP topology at any time.

## Example

```
R142#show ip bei pro
Protocol Information of BEIGRP with AS 1024:
Metric Weight: K1=1, K2=0, K3=1, K4=0, K5=0.
Filter * in access-list in12
Filter * out access-list ou12
Offset * in in23 12
Offset * out ou23 12
Redistributing: connect, ospf 1, ospf 2
Automatic network summarization is enable.
Active-time: 3(minutes)
Routing for Networks:
192.168.20.0/24
10.0.0.0/8
167.20.0.0/16
202.117.80.0/24
Distance: internal 90, external 170
Active Route:
```

## Related command

**show ip beigrp topology**

### 2.1.27 show ip beigrp topology

To display the BEIGRP topology, run the following command:

**show ip beigrp topology** [*as-number*] [*network-number subnet-mask*] [**active** | **all-links** | **pending** | **summary** | **zero-successors**]

## Parameter

Parameter	Description
<i>as-number</i>	Number of the autonomous system  If the parameter is designated, only the topology of the BEIGRP process will be displayed.
<i>network-number</i>	Displays the detailed information about the specific network.
<i>subnet-mask</i>	Mask of the subnet
<b>active</b>	Displays the routes in the active state.
<b>all-link</b>	Displays all content of the topology, including the infeasible successors. Or only the successors or the feasible successors are displayed.
<b>pending</b>	Displays the items which do not receive the response.
<b>summary</b>	Displays only the summary routes

zero-successors	No successors are displayed.
-----------------	------------------------------

## Default

None

## Command mode

EXEC or global configuration mode

## Explanation

You can run the command to check the BEIGRP topology at any time.

## Example

```
Router# show ip beigrp topology
P 10.10.10.0/24 successors: 1 FD: 13056
    via connect(Loopback1) Metric: 13056/0
P 167.20.0.0/16 successors: 1 FD: 261132
    via 202.117.80.143(Ethernet2/1) Metric: 261132/258560
P 192.166.100.0/24 successors: 1 FD: 281856
    via redistribute Metric: 281856/0
P 192.168.20.0/24 successors: 1 FD: 258560
    via connect(Ethernet1/1) Metric: 258560/0
P 202.1.1.0/24 successors: 1 FD: 297246988
    via 192.168.20.204(Ethernet1/1) Metric: 297246988/297244416
P 202.117.80.0/24 successors: 1 FD: 258560
    via connect(Ethernet2/1) Metric: 258560/0
A 202.117.93.0/24 successors: 1 FD: unreachable, R serno: 32
    via 192.168.20.141(Ethernet1/1) Metric: 271372/13056
SIA-Info: (active: 00:02:20 query-origin: Local origin)
    Unreplied Neighbors:
        via 202.117.80.143, Ethernet2/1
P 202.192.168.0/24 successors: 1 FD: 284172
    via 192.168.20.204(Ethernet1/1) Metric: 284172/281600
```

Domain	Description
160.89.90.0 等	Number of the destination network
255.255.255.0	Mask of the destination network
successors	Number of the successors
FD	Feasibility distance
Via	Gateway's address
Ethernet1/1	Local interface which receives the route

SIA-Info	Information about the active route
active	Lasting time after the active state is entered
query-origin	Reason for querying the state
Unreplied Neighbors	Neighbor list which does not receive the reply

**Related command**

**show ip beigrp neighbor**

2.1.28 **show ip beigrp traffic**

To display the statistics information about the BEIGRP flow, run the following command:

**show ip beigrp traffic** [*as-number*]

**Parameter**

Parameter	Description
<i>as-number</i>	Number of the autonomous system  If the parameter is designated, only the statistics information about the BEIGRP flow will be displayed.

**Default**

None

**Command mode**

EXEC or global configuration mode

**Explanation**

You can run the command to check the statistics information about the BEIGRP flow at any time.

**Example**

```
R142#show ip bei tra
Traffic Statistics of BEIGRP 1024
Packet Type   Hello   Update   Query   Reply   ACK
Send/Receive  770/1021 133/44  29/7    7/9     60/147
```

**Related command**

**show ip beigrp topology**



## Chapter 3 OSPF Configuration Commands

### 3.1 OSPF Configuration Commands

OSPF configuration commands include:

area authenticaion

area default-cost

area range

area stub

area virtual-link

debug ip ospf adj

debug ip ospf events

debug ip ospf flood

debug ip ospf lsa-generation

debug ip ospf packet

debug ip ospf retransmission

debug ip ospf spf

debug ip ospf tree

default-information originate

default-metric

distance ospf

filter

ip ospf cost

ip ospf dead-interval

ip ospf hello-interval

ip ospf message-digest-key

ip ospf network

ip ospf passive

ip ospf password  
ip ospf priority  
ip ospf retransmit-interval  
ip ospf transmit-delay  
neighbor  
network area  
redistribute  
router ospf  
show ip ospf  
show ip ospf border-routers  
show ip ospf database  
show ip ospf interface  
show ip ospf neighbor  
show ip ospf virtual-link  
summary-address  
timers delay  
timers hold

### 3.1.1 area authentication

To authenticate an OSPF area, run **area authentication**. To cancel the authentication of an area or delete an area, run **no area area-id authentication** or **no area area-id**.

**area *area-id* authentication [*simple* / *message-digest*]**

**no area *area-id* authentication**

**no area *area-id***

#### Parameter

Parameter	Description
<i>area-id</i>	To-be-authenticated area
<i>simple</i>	An optional parameter which will take the text authentication to the information
<i>message-digest</i>	An optional parameter which will take the MD5 authentication to the

	information
--	-------------

## Default

The OSPF packets received by the interface do not require the authentication by default.

## Command mode

Routing configuration mode

## Explanation

The authentication value will be written into the OSPF packet. However, you must guarantee that the authentication types of all routers in the same area must be same. If you want all OSPF routers in a network to conduct the OSPF interconnection, the same authentication password must be saved for them.

## Example

The following example shows how to authenticate area 0 and IP address 36.0.0.0 with plain text:

```
interface ethernet 1/0
ip address 131.119.251.201 255.255.255.0
ip ospf password adcdefgh
!
interface ethernet 1/0
ip address 36.56.0.201 255.255.0.0
ip ospf password ijklmnop
!
router ospf 1
network 36.0.0.0 255.0.0.0 area 36.0.0.0
network 131.119.0.0 255.255.0.0 area 0
area 36.0.0.0 authentication simple
area 0 authentication simple
```

## Related command

**ip ospf password**

**ip ospf message-digest-key**

### 3.1.2 area default-cost

To designate the default summary route's cost of the STUB area or the NSSA area, run **area area-id default-cost cost**. To resume the default value, run **no area area-id default-cost**.

**area *area-id* default-cost *cost***

**no area *area-id* default-cost**

**no area *area-id***

### Parameter

Parameter	Description
<i>area-id</i>	ID of the STUB area
<i>cost</i>	Cost

### Default

The default value is 1.

### Command mode

Routing configuration mode

### Explanation

The command is helpful only when it is used on the boundary router connecting the NASSA area or the STUB area.

After the **area stub default-information-originate** command is configured, the cost configured by the cost will be used in LSA to set the corresponding cost.

### Note:

When you run **no area *area-id*** to cancel the previous settings, all its subcommands will be cancelled, such as **area authentication**, **area default-cost**, **area nssa**, **area range**, **area stub** and **area virtual-link**.

### Example

The following example shows how to set the default cost of stub network 36.0.0.0 to 20:

```
interface ethernet 1/0
ip address 36.56.0.201 255.255.0.0
!
router ospf 201
network 36.0.0.0 255.0.0.0 area 36.0.0.0
area 36.0.0.0 stub
area 36.0.0.0 default-cost 20
```

Related command

**area nssa**

**area stub**

### 3.1.3 area range

To summarize the routes at the field boundary, run **area area-id range address mask[ not-advertise ]**. To cancel the previous settings, run **no area range**.

**area area-id range address mask[ not-advertise ]**

**no area area-id range address mask not-advertise**

**no area area-id range address mask**

**no area area-id**

Parameter

Parameter	Description
<i>area-id</i>	Means the fields where the fields will be summarized. It can be a decimal number or an IP address.
<i>address</i>	IP address
<i>mask</i>	IP mask
<b>advertise</b>	Means that the routes are released after they are summarized.
<b>not-advertise</b>	Means that the routes are not released after they are summarized.

### Default

The command has no effect by default.

Command mode

Routing configuration mode

Explanation

The **area range** command is not run on the ABR router, enabling ABR to be broadcast to other routers through a summary route. In this way, the route of the field boundary is miniaturized. As to the outside of the area, each address range has only one summary route.

The command can be configured on the routers in multiple areas, and OSPF, hence, can summarize many address ranges.

**Note:**

When you run **no area area-id** to cancel the previous settings, all its subcommands will be cancelled, such as **area authentication**, **area default-cost**, **area nssa**, **area range**, **area stub** and **area virtual-link**.

### Example

The following example shows how to set the summary route of the ABR router for subnet 36.0.0.0 and all hosts 192.42.110.0.

```
interface ethernet 0
ip address 192.42.110.201 255.255.255.0
!
interface ethernet 1
ip address 36.56.0.201 255.255.0.0
!
router ospf 201
network 36.0.0.0 255.0.0.0 area 36.0.0.0
network 192.42.110.0 255.0.0.0 area 0
area 36.0.0.0 range 36.0.0.0 255.0.0.0
area 0 range 192.42.110.0 255.255.255.0
```

### 3.1.4 area stub

To configure a STUB area, run **area stub**. To cancel the configuration, run **no area stub**.

**area area-id stub [no-summary]**

**no area area-id stub**

**no area area-id**

### Parameter

Parameter	Description
<i>area-id</i>	Sets the ID of the STUB area. It can be a decimal number or an IP address.
<b>no-summary</b>	Forbids the ABR router to transmit the summary link to the STUB area. It is an optional parameter.

### Default

Non-stub area

### Command mode

Routing configuration mode

## Explanation

All routers and access servers in the STUB area will be configured by the **area stub** command. The ABR router adopts the **default-cost** option to set the cost from the internal router to the STUB area.

There are two commands relative with the STUB area, that is, the **stub** command and the **default-cost** command. The **stub** command must be configured on all routers and access servers connecting the STUB area. However, the **default-cost** command is only run on the boundary router that the STUB area connects. The default-cost command is used to set the cost for the aggregation route generated by the boundary router to reach the STUB area.

To decrease the number of LSA's, you can run **no summary** on the ABR router to forbid the summary LSA to enter the STUB area.

### Note:

When you run **no area area-id** to cancel the previous settings, all its subcommands will be cancelled, such as **area authentication**, **area default-cost**, **area nssa**, **area range**, **area stub** and **area virtual-link**.

## Example

The following example shows that a default cost, 20, is distributed to stub network 36.0.0.0.

```
interface ethernet 0
ip address 36.56.0.201 255.255.0.0
!
router ospf 201
network 36.0.0.0 255.0.0.0 area 36.0.0.0
area 36.0.0.0 stub
area 36.0.0.0 default-cost 20
```

## Related command

**area authentication**

**area default-cost**

### 3.1.5 area virtual-link

To configure a virtual link, run **area virtual-link**.

```
area area-id virtual-link neighbor-ID [authentication simple | message-digest]
[dead-interval dead-value][ hello-interval hello-value][ retransmit-interval
retrans-value][ transdly dly-value][ password pass-string] [ message-digest-key
key-id MD5 md5-string]
```

```
no area area-id virtual-link neighbor-ID
```

## Parameter

Parameter	Description
<i>area-id</i>	Specifies the transit-area of the virtual link.
<i>neighbor-id</i>	OSPF router ID of the peer router of virtual link
<i>simple</i>	Configures the plain text authentication for the virtual link. The types configured on the two terminals of the virtual link must be the same.
<i>message-digest</i>	Configures the MD5 authentication for the virtual link. The types configured on the two terminals of the virtual link must be the same.
<i>dead-value</i>	Interval for the local router thinks that the neighbor dies, whose unit is second. The values configured at the two terminals of the virtual link must be same.
<i>hello-value</i>	Interval for the router to transmit the HELLO packet on the virtual link, whose unit is second. The values configured at the two terminals of the virtual link must be same.
<i>retrans-value</i>	Interval for the router to transmit the <b>re-transmit</b> packet on the virtual link, whose unit is second  The values configured at the two terminals of the virtual link must be same.
<i>dly-value</i>	Delay value which is reported by the router to LSA on the virtual link, whose unit is second  The values configured at the two terminals of the virtual link must be same. The values configured at the two terminals of the virtual link must be same.
<i>pass-string</i>	If the plain text authentication is adopted on the virtual link, you have to configure a password with up to eight characters. The values configured on the two terminals of the virtual link must be same.
<i>key-id</i>	If the MD5 authentication is adopted on the virtual link, you have to use the MD5 key. The MD5 key ranges between 1 and 255 and its values configured at the two terminals of the virtual link must be same.
<i>MD5-String</i>	Means to set the MD5 password with up to 16 characters. The values configured at the two terminals of the virtual link must be same.

## Default

The virtual link is not configured.

The default values of other parameters are shown in the following:

Hello-value: 10s, Dead-value : 40s, Retrans-value : 5s, dly-value : 1s, no authentication

## Command mode

OSPF routing configuration mode



## Explanation

In order to create a virtual link, you have to perform configuration at the two terminals of the virtual link. If only one terminal need be configured, the virtual link cannot function.

The **area-id** parameter cannot be zero because the transit area of the virtual link must not be the backbone area. The area-id configured at the two terminals of the virtual link must be same.

During configuration, neighbor ID must be ospf router-id of the peer's router, or the virtual link cannot be created even if the neighbor ID is an IP address of the peer.

Parameters configured at the two terminals of the virtual link must be same.

The authentication parameter configured on the virtual link validates only after the authentication type of the virtual link is configured or the corresponding authentication method is configured in the backbone area through the **area authentication** command. Only one authentication type can be configured on the virtual link, that is to say, the MD5 authentication and the plain text authentication are mutually exclusive.

After the virtual link is created (the neighborhood is in the FULL state), the virtual link works in the Demand Circuit mode, that is, the periodical Hello packet and the LSA refresh packet are not transmitted.

You can run no **area area-id virtual-link neighbor-ID** to cancel the previous configuration of the virtual link.

You also can run **show ip ospf virtual-link** to check the state of the virtual link.

## Example

The following example shows how to create a virtual link between router A and router B.

Configuration on router A (router-id: 200.200.200.1) :

```
!  
router ospf 100  
network 192.168.20.0 255.255.255.0 area 1  
area 1 virtual-link 200.200.200.2  
!
```

Configuration on router B (router-id: 200.200.200.2) :

```
!  
router ospf 100  
network 192.168.30.0 255.255.255.0 area 1  
area 1 virtual-link 200.200.200.1  
!
```

## Related command

**show ip ospf virtual-link**

### 3.1.6 debug ip ospf adj

To monitor the creation of the OSPF neighborhood, run the following command:

**debug ip ospf adj**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the whole creation of the OSPF neighborhood.

#### Example

```
Router# debug ip ospf adj
OSPF: Interface 192.168.40.0 on Serial1/0 going down
OSPF NBR: 192.168.40.2 address 192.168.40.2 on Serial1/0 is dead, state DOWN
OSPF NBR: 192.168.40.3 address 192.168.40.3 on Serial1/0 is dead, state DOWN
Line on Interface Serial1/0, changed state to up
Line protocol on Interface Serial1/0 changed state to up
OSPF: Interface 192.168.40.0 on Serial1/0 going Up
OSPF: 2 Way Communication to 192.168.40.2 on Serial1/0, state 2WAY
OSPF: NBR 192.168.40.2 on Serial1/0 Adjacency OK, state NEXSTART.
OSPF: NBR Negotiation Done. We are the SLAVE
OSPF: NBR 192.168.40.2 on Serial1/0 Negotiation Done. We area the SLAVE
OSPF: Exchange Done with 192.168.40.2 on Serial1/0
OSPF: Loading Done with 192.168.40.2 on Serial1/0, database Synchronized (FULL)
OSPF: 2 Way Communication to 192.168.40.3 on Serial1/0, state 2WAY
OSPF: NBR 192.168.40.3 on Serial1/0 Adjacency OK, state NEXSTART.
OSPF: NBR Negotiation Done. We are the SLAVE
OSPF: NBR 192.168.40.3 on Serial1/0 Negotiation Done. We area the SLAVE
OSPF: Bad Sequence with 192.168.40.3 on Serial1/0, state NEXSTART
OSPF: NBR Negotiation Done. We are the SLAVE
OSPF: NBR 192.168.40.3 on Serial1/0 Negotiation Done. We area the SLAVE
OSPF: Exchange Done with 192.168.40.3 on Serial1/0
```

OSPF: Loading Done with 192.168.40.3 on Serial1/0, database Synchronized (FULL)

.....

### 3.1.7 debug ip ospf events

To monitor the OSPF interface and the neighborhood event, run the following command:

#### **debug ip ospf events**

##### Parameter

None

##### Default

None

##### Command mode

EXEC

##### Explanation

According to the information exported by the command, you can check the OSPF port and the neighbor trigger event.

##### Example

```
Router# debug ip ospf events
OSPF: Interface Serial1/0 going Up
OSPF: INTF(192.168.40.0) event INTF_UP
OSPF: NBR(192.168.40.2) event HELLO_RX
OSPF: NBR(192.168.40.2) event TWOWAY
OSPF: NBR(192.168.40.2) event ADJ_OK
OSPF: NBR(192.168.40.2) event NEGO_DONE
OSPF: NBR(192.168.40.2) event EXCH_DONE
OSPF: NBR(192.168.40.2) event LOAD_DONE
OSPF: NBR(192.168.40.3) event HELLO_RX
OSPF: NBR(192.168.40.3) event TWOWAY
OSPF: NBR(192.168.40.3) event ADJ_OK
OSPF: NBR(192.168.40.3) event NEGO_DONE
OSPF: NBR(192.168.40.3) event SEQ_MISMATCH
OSPF: NBR(192.168.40.3) event NEGO_DONE
OSPF: NBR(192.168.40.3) event EXCH_DONE
OSPF: NBR(192.168.40.3) event LOAD_DONE
.....
```

### 3.1.8 debug ip ospf flood

To monitor the flooding of the OSPF database, run the following command:

**debug ip ospf flood**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the flooding of the OSPF database.

#### Example

```
Router# debug ip ospf flood
OSPF: rcv UPDATE, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 2 SEQ
0x8000022B
OSPF: Send UPDATE, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ
0x80000234
OSPF: Send ACK, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 2 SEQ 0x8000022B
OSPF: rcv ACK, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ
0x80000234
OSPF: rcv ACK, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 18 SEQ
0x80000233
OSPF: Send UPDATE, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 10 SEQ
0x8000022B
OSPF: rcv UPDATE, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 5 SEQ
0x8000021C
OSPF: Send UPDATE, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 6 SEQ
0x8000021C
OSPF: Send UPDATE, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ
0x80000235
OSPF: rcv ACK, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 4 SEQ 0x8000021C
.....
```

### 3.1.9 debug ip ospf lsa-generation

To monitor the LSA generation of OSPF, run the following command:

**debug ip ospf lsa-generation**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the OSPF port and the neighbor trigger event.

#### Example

```
router# debug ip ospf lsa-generation
```

```
.....
```

```
OSPF: Send UPDATE, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 10 SEQ 0x8000022D
```

```
OSPF: rcv UPDATE, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 5 SEQ 0x8000021E
```

```
OSPF: Send UPDATE, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 6 SEQ 0x8000021E
```

```
OSPF: Send UPDATE, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ 0x80000239
```

```
OSPF: rcv ACK, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 4 SEQ 0x8000021E
```

```
OSPF: Send ACK, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 5 SEQ 0x8000021E
```

```
OSPF: rcv UPDATE, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 1 SEQ 0x8000022E
```

```
OSPF: Send UPDATE, type 1 LSID 192.168.40.2 ADV_RTR 192.168.40.2 AGE 2 SEQ 0x8000022E
```

```
OSPF: rcv ACK, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ 0x80000239
```

```
OSPF: rcv ACK, type 1 LSID 192.168.40.3 ADV_RTR 192.168.40.3 AGE 6 SEQ 0x8000021E
```

```
OSPF: rcv ACK, type 1 LSID 192.168.20.240 ADV_RTR 192.168.20.240 AGE 1 SEQ 0x80000239
```

```
.....
```

### 3.1.10 debug ip ospf packet

To monitor the OSPF packet, run the following command:

**debug ip ospf packet**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the OSPF port and the neighbor trigger event.

#### Example

```
router# debug ip ospf packet
OSPF: Recv HELLO packet from 192.168.40.3 (addr: 192.168.40.3) area 0 from Serial1/0
OSPF: End of hello processing
OSPF: Send HELLO to 224.0.0.5 on Loopback0
      HelloInt 10 Dead 40 Opt 0x2 Pri 1 len 44
OSPF: Send HELLO to 224.0.0.5 on Loopback0
      HelloInt 10 Dead 40 Opt 0x2 Pri 1 len 44
OSPF: Send HELLO to 224.0.0.5 on Loopback0
      HelloInt 10 Dead 40 Opt 0x2 Pri 1 len 44
OSPF: Recv HELLO packet from 192.168.40.2 (addr: 192.168.40.2) area 0 from Serial1/0
OSPF: End of hello processing
OSPF: Send HELLO to 224.0.0.5 on Serial1/0
      HelloInt 30 Dead 120 Opt 0x2 Pri 1 len 52
OSPF: Recv HELLO packet from 192.168.40.3 (addr: 192.168.40.3) area 0 from Serial1/0
OSPF: End of hello processing
OSPF: Send HELLO to 224.0.0.5 on Loopback0
      HelloInt 10 Dead 40 Opt 0x2 Pri 1 len 44
.....
```

### 3.1.11 debug ip ospf retransmission

To monitor the retransmission of the OSPF packet, run the following command:

**debug ip ospf retransmission**

Parameter

None

Default

None

Command mode

EXEC

Explanation

According to the information exported by the command, you can check the retransmission of the OSPF packet.

Example

```
router# debug ip ospf retransmission
OSPF: retransmit UPDATE to 192.168.40.3 (RID 192.168.40.3), state FULL
.....
```

### 3.1.12 debug ip ospf spf

To monitor the SPF calculation route of OSPF, run the following commands:

**debug ip ospf spf**

**debug ip ospf spf intra**

**debug ip ospf spf inter**

**debug ip ospf spf external**

Parameter

None

## Default

None

## Command mode

EXEC

## Explanation

According to the information exported by the command, you can check the calculation of the OSPF route.

## Example

```
router# debug ip ospf spf
OSPF: run ospf_spf_run
OSPF: start doing SPF for AREA 0.0.0.0
OSPF: RTAB_REV(ospf) 1390.
OSPF : Initializing to do SPF
OSPF: addroute LSID 192.168.20.240
OSPF: ospf_nh_find: 192.168.40.2
.....
OSPF: addroute LSID 192.168.40.3
OSPF: build a OSPF_ROUTE, dest: 192.168.40.3
OSPF: addroute LSID 192.168.40.2
.....
OSPF: SPF Area A running Network Summary
OSPF: Processing LS_SUM_NET 192.168.40.24, mask 255.255.255.248, adv 192.168.40.3,
age 599
OSPF: addroute LSID 192.168.40.24
OSPF: ospf_build_route RT 192.168.40.24
OSPF: build route 192.168.40.24(255.255.255.248).
.....
OSPF: Processing LS_SUM_NET 1.1.1.1, mask 255.255.255.255, adv 192.168.20.240, age 228
OSPF: addroute LSID 192.168.20.236
OSPF: build a OSPF_ROUTE, dest: 192.168.20.236
OSPF: start Building AS External Routes
OSPF: processing LS_ASE 192.168.42.0, mask 255.255.255.248, adv 192.168.20.236, age 258
OSPF: addroute LSID 192.168.42.0
OSPF: ospf_build_route RT 192.168.42.0
OSPF: build route 192.168.42.0(255.255.255.248).
OSPF: processing LS_ASE 192.168.43.0, mask 255.255.255.0, adv 192.168.20.236, age 258
OSPF: addroute LSID 192.168.43.0
OSPF: ospf_build_route RT 192.168.43.0
OSPF: build route 192.168.43.0(255.255.255.0).
OSPF: processing LS_ASE 192.168.44.0, mask 255.255.255.0, adv 192.168.20.236, age 258
```



```

OSPF: addroute LSID 192.168.44.0
OSPF: ospf_build_route RT 192.168.44.0
OSPF: build route 192.168.44.0(255.255.255.0).

```

```

.....

```

```

OSPF: end doing SPF for AREA 0.0.0.0

```

**Domain description:**

Domain	Description
LSA(192.168.20.23 6, LS_SUM_ASB)	ID and type of LSA

**3.1.13 debug ip ospf tree**

To monitor the SPF tree establishment of OSPF, run the following command:

**debug ip ospf tree****Parameter**

None

**Default**

None

**Command mode**

EXEC

**Explanation**

According to the information exported by the command, you can check the SPF tree establishment of OSPF.

**Example**

```

router# debug ip ospf tree
B3710_221#
OSPF: add LSA(192.168.40.0, LS_STUB) 1600 under LSA(192.168.20.240, LS_RTR)
OSPF: add LSA(192.168.40.2, LS_RTR) 1600 under LSA(192.168.20.240, LS_RTR)
OSPF: add LSA(192.168.40.3, LS_RTR) 1600 under LSA(192.168.20.240, LS_RTR)
OSPF: add LSA(192.168.40.1, LS_STUB) 0 under LSA(192.168.20.240, LS_RTR)
OSPF: add LSA(192.168.40.3, LS_STUB) 1600 under LSA(192.168.40.3, LS_RTR)
OSPF: add LSA(192.169.1.5, LS_RTR) 3200 under LSA(192.168.40.2, LS_RTR)
OSPF: add LSA(192.168.40.18, LS_STUB) 1600 under LSA(192.168.40.2, LS_RTR)
OSPF: add LSA(192.168.40.2, LS_STUB) 1600 under LSA(192.168.40.2, LS_RTR)

```

OSPF: add LSA(192.168.40.17, LS\_STUB) 3200 under LSA(192.169.1.5, LS\_RTR)  
 OSPF: add LSA(192.168.40.24, LS\_SUM\_NET) 1601 under LSA(192.168.40.3, LS\_RTR)  
 OSPF: add LSA(192.168.40.32, LS\_SUM\_NET) 3200 under LSA(192.168.40.2, LS\_RTR)  
 OSPF: add LSA(192.168.40.40, LS\_SUM\_NET) 14577 under LSA(192.169.1.5, LS\_RTR)  
 OSPF: add LSA(192.168.20.236, LS\_SUM\_ASB) 3200 under LSA(192.168.40.2, LS\_RTR)

**Relative fields are explained in the following table:**

Domain	Description
LSA(192.168.20.236, LS_SUM_ASB)	ID and type of LSA
add	Sub-LSA
under	parent LSA

### 3.1.14 default-information originate (OSPF)

To generate the default route to enter the OSPF route field, run **default-information originate [always] [route-map map-name]**.

**default-information originate [always] [route-map map-name]**

**no default-information originate [always] [route-map map-name]**

#### Parameter

Parameter	Description
originate	Transmits an external route to the OSPF routing field.
Always	An optional parameter, meaning that the system will broadcast the default route no matter whether the default route exists or not
route-map map-name	An optional parameter, meaning that a default route will be generated if the route mapping is realized

#### Default

The default route is not generated.

#### Command mode

Routing configuration mode

#### Explanation

If the router distributes the routes to the OSPF route field through the **redistribute** command and the **default-information** command, the router becomes the ASBR router. However, the ASBR router does not generate the default route and does not transmit it to the OSPF field. Or a default route must be set for generating the default route unless the **always** option is set.

When the command is used, the default network must be in the routing table and the default network must satisfy the **route-map** option. To cancel the default network in the routing table, you can run **default-information originate always route-map**.

### Example

The following example shows that the cost of the default route distributed to the OSPF routing table is set to 100 and its type is set to 1:

```
router ospf 109
 redistribute rip
 default-information originate
```

### Related command

#### Redistribute

### 3.1.15 default-metric

To set the default route weight which is guided into the route, run **default-metric value**. To resume the default settings, run **no default-metric value**.

**default-metric value**

**no default-metric**

### Parameter

Parameter	Description
<i>value</i>	To-be-set route weight, ranging between 1 and 4294967295

### Default

The default route weight is 10.

### Command mode

Routing configuration mode

### Explanation

The **default-metric** command is used to set the default routing weight when the route of other routing protocol is guided into the OSPF packet. When the **redistribute** command is used to guide the route of other routing protocol, the default routing weight designated by the **default-metric** command will be guided the specific routing weight will not be specified.

## Example

The following example shows how to set the default route weight of other routing protocols to 3:

```
router_config_ospf_100#default-metric 3
```

## Related command

**redistribute**

### 3.1.16 distance ospf

To define the management distance of the OSPF route based on OSPF type, run **distance ospf** {[intra-area *dist1*] [inter-area *dist2*] [external *dist3*]}. To cancel the configuration, run **no distance ospf**.

**distance ospf** {[intra-area *dist1*] [inter-area *dist2*] [external *dist3*]}

**no distance ospf** [intra-area] [inter-area] [external]

## Parameter

Parameter	Description
intra-area <i>dist1</i>	Sets the management distance for all routes in an area (optional). Its default value is 110.
inter-area <i>dist2</i>	Sets the management distance for all routes which are transmitted from one area to another area (optional). Its default value is 110.
external <i>dist3</i>	Sets the management distance for the routes learned from other routing areas through the <b>redistribution</b> command. Its default value is 110.

## Default

**intra-area:** 110

**inter-area:** 110

**external:** 150

## Command mode

Routing configuration mode

## Explanation

At least one parameter is in the command line.

The command has the same functions as the **distance** command. However, the **distance ospf** command can be used to configure the distance for all routing groups.

### Example

The following example shows how to set the external distance to 200:

```
Router A
router ospf 1
 redistribute ospf 2
 distance ospf external 200
!
router ospf 2
 redistribute ospf 1
 distance ospf external 200
Router B
router ospf 1
 redistribute ospf 2
 distance ospf external 200
!
router ospf 2
 redistribute ospf 1
 distance ospf external 200
```

### Related command

#### **distance**

### 3.1.17 filter

To set the routing filtration table, run **filter {*interface-type interface-number* | \*} {in | out} {*access-list access-list-name* | gateway *access-list-name* / prefix-list *prefix-list-name*}**. To resume the default settings, run **no filter {*interface-type interface-number* | \*} {in | out} {*access-list access-list-name* | gateway *access-list-name* | prefix-list *prefix-list-name*}**.

**filter {*interface-type interface-number* | \*} {in | out} {*access-list access-list-name* | gateway *access-list-name* | prefix-list *prefix-list-name*}**

**no filter {*interface-type interface-number* | \*} {in | out} {*access-list access-list-name* | gateway *access-list-name* | prefix-list *prefix-list-name*}**

### Parameter

Parameter	Description
<i>interface-type</i>	Type
<i>interface-number</i>	Port number

<b>*</b>	All interfaces
<i>In</i>	Filtrates the received OSPF routes.
<i>out</i>	Filters the transmitted routes, which is not for a specific interface but for all interfaces.
<i>access-list-name</i>	Name of the access control list
<i>prefix-list-name</i>	Name of the prefix list

**Default**

None

**Command mode**

Routing configuration mode

**Explanation**

None

**Example**

filter \* in access-list mylist

**3.1.18 ip ospf authentication**

To designate the authentication mode for receiving and transmitting the OSPF packet, run **ip ospf authentication [simple / message-digest]**. To cancel the OSPF authentication on an interface, run **no ip ospf authentication**.

**ip ospf authentication [simple / message-digest]****no ip ospf authentication****Parameter**

Parameter	Description
simple	An optional parameter which will take the text authentication to the information
message-digest	An optional parameter which will take the MD5 authentication to the information

**Default**

The OSPF packets received by the interface do not require the authentication by default.

## Command mode

Interface configuration mode

## Explanation

If the **ip ospf authentication simple** command is used to designate the text authentication mode, you need set a text password by running the **ip ospf password** command. Before the **ip ospf authentication message-digest** command is used to configure the MD5 authentication mode for a designated interface, the MD5 key must be configured through the **ip ospf message-digest-key** command. If you want all OSPF routers in a network to conduct the OSPF interconnection, the same authentication mode and password must be saved for them.

Considering the compatibility, the authentication mode for an OSPF domain is still reserved. If the OSPF authentication mode is not configured on an interface, the authentication mode of the file of the interface will be used.

## Example

The following example shows how to perform the MD5 authentication on interface e1/0.

```
interface ethernet 1/0
ip address 131.119.251.201 255.255.255.0
ip ospf authentication message-digest-key
ip ospf message-digest-key 1 md5 abcdefg
!
router ospf 1
network 131.119.0.0 255.255.0.0 area 0
```

## Related command

**ip ospf password**

**ip ospf message-digest-key**

**area authentication**

### 3.1.19 ip ospf cost

To designate the cost for the OSPF protocol running on the interface, run **ip ospf cost cost**. To resume the default settings, run **no ip ospf cost**.

**ip ospf cost** *cost*

**no ip ospf cost**

**Parameter**

Parameter	Description
<i>cost</i>	Cost for the OSPF protocol running on the interface, which is an integer between 1 and 65535

**Default**

The default cost for the OSPF protocol running on the interface is obtained based on the rate of the port.

**Command mode**

Interface configuration mode

**Example**

The following example shows how to set the cost for the OSPF protocol running on interface serial 0 to 2:

```
ip ospf cost 2
```

**3.1.20 ip ospf dead-interval**

To designate the dead interval of the neighboring router, run **ip ospf dead-interval seconds**. To resume the default value, run **ip ospf dead-interval**.

**ip ospf dead-interval** *seconds*

**ip ospf dead-interval**

**Parameter**

Parameter	Description
<i>Seconds</i>	Value of the dead interval for the neighboring router, which ranges from 1 to 65535 seconds

**Default**

The dead interval for the neighboring router is 40 seconds by default.

**Command mode**

Interface configuration mode



## Explanation

The value of the **dead-interval** parameter will be written to the HELLO packet and will be transmitted along with the HELLO packet. It must be ensured that the **dead-interval** parameter must be identical with that between the neighboring routers and the value of the **dead-interval** parameter must be four times of the value of the **hello-interval** parameter.

## Example

The following example shows how to set the dead interval of the neighboring router on interface serial0 to 60 seconds.

```
router_config_S1/0#ip ospf dead-interval 60
```

## Related command

**ip ospf hello-interval**

### 3.1.21 ip ospf hello-interval

To designate the interval for transmitting the HELLO packet on the interface, run **ip ospf hello-interval seconds**. To resume the default settings, run **no ip ospf hello-interval**.

**ip ospf hello-interval seconds**

**no ip ospf hello-interval**

## Parameter

Parameter	Description
<i>Seconds</i>	Transmission interval of the HELLO packet, ranging from 1 to 255 seconds

## Default

The default interval for transmitting the HELLO packet on the interface is 10 seconds.

## Command mode

Interface configuration mode

## Explanation

The value of the **hello-interval** parameter will be written to the Hello packet and will be transmitted along with the HELLO packet. The smaller the hello-interval is, the sooner the change of the network topology will be found. However, much more path cost will

be paid. It must be ensured that the parameter must be identical with that between the neighboring routers.

### Example

The following example shows that the interval for transmitting the HELLO packet on interface serial 1/0 is set to 20 seconds.

```
router_config_S1/0#ip ospf hello-interval 20
```

### Related command

**ip ospf dead-interval**

#### 3.1.22 ip ospf message-digest-key

To enable OSPF to use the MD5 authentication, run **ip ospf message-digest-key *keyid* md5 *key***. To cancel the configuration, run **no ip ospf message-digest-key**.

**ip ospf message-digest-key *keyid* md5 *key***

**no ip ospf message-digest-key *keyid***

### Parameter

Parameter	Description
<i>keyid</i>	Authentication ID, ranging between 1 and 255
<i>key</i>	String with 16 bits of numbers or numbers

### Default

The OSPF MD5 authentication mode is not used.

### Command mode

Interface configuration mode

### Explanation

In general, each interface generates a packet about the authentication information or the verification input through a key. The neighbor router must have the same key.

The procedure to change the key is listed as follows:

Suppose the current configuration is as follows:

```
interface ethernet 1
ip ospf message-digest-key 100 md5 OLD
```

The following configuration is what you want to change to:

```
interface ethernet 1
ip ospf message-digest-key 101 md5 NEW
```

The system supposes that the neighbor router has no new key, so the system has to transmit multiple copies of a packet, each copy having different key. In the example, the router transmits two copies of each packet, one key being 100 and the other key being 101.

The neighbor router can work when the administrator modifies the key. Once all neighbors adopt the new key, the procedure will end. If the system receives the packets containing the new key from the neighbor router, the system regards that the neighbor router has the new key.

After all neighbors adopt the new key, the old key will be deleted.

```
interface ethernet 1
no ip ospf message-digest-key 100
```

It is recommended that each interface cannot have multiple keys. After a new key is added, the old key shall be deleted to prevent the local system from communicating with the unfriendly system using the old key. At the same time, the communication load can be lessened if the old key is deleted.

### Example

The following example shows how to set a new key to **19** and its password to **8ry4222**.

```
interface ethernet 1
ip ospf message-digest-key 10 md5 xvv560qle
ip ospf message-digest-key 19 md5 8ry4222
```

### Related command

#### area authentication

### 3.1.23 ip ospf network

To set the network type for the interface, run **ip ospf network { broadcast | nonbroadcast | point\_to\_multipoint | point-to-point}**. To cancel the configuration, run **no ip ospf network**.

```
ip ospf network { broadcast | nonbroadcast | point_to_multipoint | point-to-point}
```

```
no ip ospf network { broadcast | nonbroadcast | point_to_multipoint | point-to-point}
```

### Parameter

Parameter	Description
<b>broadcast</b>	Sets the network type of the interface to <b>broadcast</b> .

<b>nonbroadcast</b>	Sets the network type of the interface to <b>NBMA</b> .
<b>point-to-point</b>	Sets the network type of the interface to <b>point-to-point</b> .
<b>point-to-multipoint</b>	Sets the network type of the interface to <b>point-to-multipoint</b> .

### Command mode

Interface configuration mode

### Explanation

The interface in the broadcast network without multi-address access should be set to NBMA. In the NBMA network, the network should be set to **point-to-multipoint** if any two routers cannot be ensured to be directly reachable.

### Example

The following example shows how to set interface Serial1/0 to NBMA.

```
router_config_S1/0#ip ospf network nonbroadcast
```

### 3.1.24 ip ospf passive

To disable the transmission of the HELLO packet on the interface, run **ip ospf passive**. To reactivate the transmission of the HELLO packet on the interface, run **no ip ospf passive**.

**ip ospf passive**

**no ip ospf passive**

### Parameter

The command has no parameters or keywords.

### Default

The HELLO packet is transmitted on the interface.

### Command mode

Interface configuration mode

### Explanation

If you cancel the transmission of the HELLO packets on an interface, some specific subnet will continue declaring to other interfaces that the route update from other

routers to the interface can still be accepted and handled. The function is used on the STUB network because no other OSPF routers exist on the network.

### Example

The following example shows that the HELLO packet will be transmitted to all interfaces (except Ethernet interface 1/0) in network 172.16.0.0.

```
interface ethernet 1/0
ip address 172.16.0.1 255.255.0.0
ip ospf passive
router ospf 110
network 172.16.0.0 255.255.0.0 area 1
```

### Related command

None

### 3.1.25 ip ospf password

To configure the password for the neighboring route, run **ip ospf password *password***. To cancel the password, run **no ip ospf password**.

**ip ospf password *password***

**no ip ospf password**

### Parameter

Parameter	Description
<i>password</i>	String of any continuous 8 bits of characters

### Default

No password

### Command mode

Interface configuration mode

### Explanation

The password generated by the command will be directly inserted into the OSPF routing packet. You can configure a password for each interface's network. Only when all neighboring routers must have the same password can the OSPF routing information be exchanged.

**Note:** The command invalidates only after the authentication is allowed through the **area authentication** command.

## Example

```
ip ospf password yourpass
```

## Related command

**area authentication**

### 3.1.26 ip ospf priority

To configure the priority for the interface to choose the router, run **ip ospf priority *priority***. To resume the default value, run **no ip ospf priority**.

**ip ospf priority *priority***

**no ip ospf priority**

## Parameter

Parameter	Description
<i>priority</i>	Priority to choose the router, ranging between 0 and 255

## Default

The default priority for the interface to choose the routers is 1.

## Command mode

Interface configuration mode

## Explanation

When two routers in the same network segment want to be the selection router, the router with higher priority will be selected. If the priority of the two routers is the same, the router with a larger ID is selected. When the priority of a router is 0, the router cannot be selected as the selection router or the standby selection router. The priority is effective only on the networks except the point-to-point network.

## Example

The following example shows how to set the priority to 8 when interface Serial1/0 selects the selection router.

```
router_config_S1/0#ip ospf priority 8
```

## Related command

**neighbor**

### 3.1.27 ip ospf retransmit-interval

To designate the retransmission interval for transmitting the link state broadcast between the interface and the neighboring router, run **ip ospf retransmit seconds**. To resume the default value, run **no ip ospf retransmit**.

**ip ospf retransmit** *seconds*

**no ip ospf retransmit**

#### Parameter

Parameter	Description
<i>seconds</i>	Transmission interval for transmitting the link state broadcast between the interface and the neighboring router, ranging between 1 and 65535 seconds

#### Default

The default interval for transmitting the link state broadcast between the interface and the neighboring router is 5 seconds.

#### Command mode

Interface configuration mode

#### Explanation

When a router transmits the link-state broadcast to its neighbor, the command will maintain the link-state broadcast until the peer receives the acknowledgement. If the link-state broadcast is not received during the transmission interval, it will be retransmitted. The value of the **seconds** parameter must be larger than the round-trip time for a packet transmitting between two routers.

#### Example

The following example shows how the default interval for transmitting the link-state broadcast between interface Serial1/0 and the neighboring router is set to 8 seconds.

```
router_config_S1/0#ip ospf retransmit 8
```

### 3.1.28 ip ospf transmit-delay

To set the delay for the link-state broadcast to be transmitted on the interface, run **ip ospf transit-delay time**. To resume the default value, run **no ip ospf transit-delay**.

**ip ospf transit-delay** *time*

**no ip ospf transit-delay**

## Parameter

Parameter	Description
<i>time</i>	Delay for transmission of the link state broadcast on an interface Its unit is second and it ranges between 1 and 65535.

## Default

The default delay for the link-state broadcast to be transmitted on the interface is 1 second.

## Command mode

Interface configuration mode

## Example

The following example shows how to set the delay for transmitting the link-state broadcast on interface Serial1/0 to 3 seconds.

```
router_config_S1/0#ip ospf transit-delay 3
```

## 3.1.29 neighbor

To configure the OSPF router connecting the non-broadcast network, run **neighbor**. To cancel the configuration, run **no neighbor**.

**neighbor** *ip-address* [*priority number*] [**poll-interval** *seconds*] [**cost** *number*]

**no neighbor** *ip-address* [*priority number*] [**poll-interval** *seconds*] [*cost number*]

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighboring router
<b>priority</b> <i>number</i>	An 8-bit priority number Its default value is 0. The option cannot be used on the point-to-multipoint interface.
<b>poll-interval</b> <i>seconds</i>	An optional parameter, standing for the query interval It must be larger than the interval of the HELLO packet. The option cannot be used on the point-to-multipoint interface.
<b>cost</b> <i>number</i>	An optional parameter, which is used to designate the cost for the neighboring router If the cost of the neighboring router is not specified, run <b>ip ospf cost</b> command to designate a cost. It functions in the point-to-multipoint network.



## Default

Default

## Command mode

Routing configuration mode

## Explanation

The OSPF can function in the broadcast mode in the X.25 network and the frame-relay network. See the detailed description about the **X25 map** command and the **frame-relay map** command.

For the non-broadcast network neighbor, you must configure it on the router. The neighbor's address must be the main address of the interface.

The HELLO packet will be transmitted to the neighboring router even if the neighboring router is not in the active state. These HELLO packets are transmitted in poll interval digression mode.

When the router is enabled, the router only transmits the HELLO packet to the router with non-zero priority. The router may become the DR router or the BDR router. After the DR router and the BDR router are selected, they will transmit the HELLO packet to the neighborhood list.

## Example

The following example shows that the address of the router, 131.108.3.4, is designated as the non-broadcast network, the priority is set to 1 and the **poll interval** parameter is set to 180 seconds.

```
router ospf
neighbor 131.108.3.4 priority 1 poll-interval 180
```

The following example shows how to configure the point-to-multipoint broadcast network.

```
interface Serial0
ip address 10.0.1.1 255.255.255.0
ip ospf network point-to-multipoint non-broadcast
encapsulation frame-relay
no keepalive
frame-relay local-dlci 200
frame-relay map ip 10.0.1.3 202
frame-relay map ip 10.0.1.4 203
frame-relay map ip 10.0.1.5 204
no shut
!
router ospf 1
network 10.0.1.0 255.255.255.0 area 0
```

```
neighbor 10.0.1.3 cost 5
neighbor 10.0.1.4 cost 10
neighbor 10.0.1.5 cost 15
```

Related command

**ip ospf priority**

### 3.1.30 network area

To define several network segments in an area as a network range, run **network network mask area area\_id [advertise | not-advertise ]**.

**network network mask area area\_id [ advertise | not-advertise ]**

**[ no ] network network mask area area\_id [ advertise | not-advertise ]**

Parameter

Parameter	Description
<i>network</i>	IP address of the network in the dotted-decimal format
<i>mask</i>	Mask in the dotted-decimal format
<i>area_id</i>	ID of the area
Advertise notadvertise	Decides whether the summary information about the network range route will be broadcasted

Default

The network range is not configured in the system by default.

Command mode

Routing configuration mode

Explanation

Once a network range is added to an area, internal routes whose IP addresses are in the network range will not be independently broadcasted to other areas but the summary information about these routes will be broadcasted. The network range and the limitation to the network range are introduced to lessen the exchange volume of the routing information among areas.

Example

The following example shows how to add range 10.0.0.0 255.0.0.0 to area 2.

```
router_config_ospf_10#network 10.0.0.0 255.0.0.0 area 2
```

### 3.1.31 redistribute

To configure the route where OSPF forwards other routing protocols, run **redistribute**. To resume the default settings, run **no redistribute**.

**redistribute** *protocol* [*as-number*] [*route-map map-tag*]

**no redistribute** *protocol* [*as-number*] [*route-map map-tag*]

#### Parameter

Parameter	Description
<b>protocol</b>	Forwards the learned original protocol which is one of the protocols: <b>beigrp</b> , <b>bgp</b> , <b>connect</b> , <b>ospf</b> , <b>rip</b> and <b>static</b> .
<i>as_number</i>	(optional) number of the autonomous system which is not for the <b>connect</b> , <b>rip</b> or <b>static</b> parameter
<i>map-tag</i>	(optional) name of the route map

#### Default

OSPF does not forward the routes of other routing protocols.

#### Command mode

Routing configuration mode

#### Explanation

None

#### Example

The following example shows how to forward the OSPF protocol of autonomous system 0.

Redistribute ospf 0

### 3.1.32 router ospf

To configure the OSPF route, run **router ospf**. To disable the OSPF route on the router, run **no router ospf**.

**router ospf** *process-id*

**no router ospf** *process-id*

## Parameter

Parameter	Description
<i>process-id</i>	Identifies the OSPF process. It is a positive integer distributed by the local router. It uniquely stands for a OSPF process.

## Default

No OSPF process is defined.

## Command mode

Global configuration mode

## Explanation

Multiple OSPF processes can be in a router.

## Example

The following example shows how to set the OSPF process to 109.

```
router ospf 109
```

## Related command

**network area**

## 3.1.33 show ip ospf

To display the main OSPF information, run the following command:

**show ip ospf** [*process-id*]

## Parameter

Parameter	Description
<i>process-id</i>	ID of the process, which is an optional parameter

## Default

None

## Command mode

EXEC

## Explanation

The information exported by the command can help checking the OSPF faults. If the **process-id** parameter follows the command, the information about the global configuration of the OSPF process is displayed.

## Example

The following example shows that the configuration information about all OSPF processes will be displayed.

```
router#show ip ospf
OSPF process: 1, Router ID is 192.168.99.81
Distance: intra-area 110 inter-area 130 external 150
Source Distance Access-list
240.240.1.1/24 1 what
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Number of areas is 3
AREA: 1
Number of interface in this area is 1(UP: 1)
Area authentication type: None
AREA: 36.0.0.1
This is a stub area.
Number of interface in this area is 0(UP: 0)
Area authentication type: None
AREA: 192.168.20.0
Number of interface in this area is 0(UP: 0)
Area authentication type: None
Net Range list:
10.0.0.0/255.0.0.0 Not-Advertise
140.140.0.0/255.255.0.0 Advertise
filter list on receiving UPDATE is Gateway: weewe
filter list on sending UPDATE is Prefix: trtwd
Summary-address list:
150.150.0.0/16 advertise
router#
```

Relative fields are explained in the following table:

Domain	Description
OSPF process: 1	ID of the process
Router ID is 192.168.99.81	ID of the router
Distance: intra-area 110 inter-area 130 external 150	Default management distance adopted when the current router generates the route
Source Distance Access-list	Management distance based on the detailed routing configuration

SPF schedule delay 5 secs, Hold time between two SPFs 10 secs	Two timer values relative with OSPF
Number of areas is 3	Number of the currently-configured fields, and parameters configured in each field
filter list on receiving	Filtration of the input routes
filter list on sending	Filtration of the output routes
Summary-address list	Route aggregation

### 3.1.34 show ip ospf border-routers

To display the data option about ABR and ASBR in the router, run the following command:

**show ip ospf border-routers**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Example

```
router#
router#sh ip os bor
OSPF process: 1
Codes: i - Intra-area route, I - Inter-area route
Destination Adv-Rtr Cost Type Area
i 192.168.20.77 192.168.20.77 11 ABR 0
router#
```

Relative fields are explained in the following table:

Domain	Description
Destination	ID of the destination router
Adv-Rtr	Next hop of the destination router
Cost	Cost of using the router
Type	Type of the destination router, which may be ABR, ASBR or both

Area	ID of the field where the learned routes belong to
------	--

### 3.1.35 show ip ospf database

To display the database information about the OSPF connection state, run the following command:

**show ip ospf database**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

The information exported by the command can help to check the database information about the OSPF connection state and to find the reason of the faults.

#### Example

```

router#
router#show ip ospf database
OSPF process: 1
(Router ID 192.168.99.81)
AREA: 0
Router Link States
Link ID ADV Router Age Seq # Checksum Link count
192.168.20.77 192.168.20.77 77 0x8000008a 0x90ed 1
192.168.99.81 192.168.99.81 66 0x80000003 0xd978 1
Net Link States
Link ID ADV Router Age Seq # Checksum
192.168.20.77 192.168.20.77 80 0x80000001 0x9625
Summary Net Link States
Link ID ADV Router Age Seq # Checksum
192.168.99.0 192.168.99.81 87 0x80000003 0xd78c
AREA: 1
Router Link States
Link ID ADV Router Age Seq # Checksum Link count

```

```
192.168.99.81 192.168.99.81 70 0x80000002 0x0817 1
```

Summary Net Link States

Link ID ADV Router Age Seq # Checksum

```
192.168.20.0 192.168.99.81 66 0x80000006 0xd1c1
```

router#

Relative fields are explained in the following table:

Domain	Description
AREA: 1	Current area
Router Link States/Net Link States/Summary Net Link States	LSA type
Link ID	LSA ID
ADV Router	Advertisement router
Age	Advertisement age
Seq #	Sequence number
Checksum	Checksum

### 3.1.36 show ip ospf interface

To display the information about the OSPF interface, run the following command:

**show ip ospf interface**

Parameter

None

Default

None

Command mode

EXEC

Explanation

According to the information displayed by the command, you can check the OSPF configuration and its running state, which helps you to detect the OSPF faults.

Example

```
router#sh ip os int
```



Ethernet 1/0 is up, line protocol is up  
 Internet Address: 192.168.20.81/24, Nettype: BROADCAST  
 OSPF process is 1, AREA 0, Router ID 202.96.135.201  
 Cost 10, Transmit Delay is 1 sec, Priority 1  
 Hello interval 10, Dead timer 40, Retransmit 5  
 OSPF INTF State is DrOther  
 Designated Router id 131.119.254.10, Interface address 131.119.254.10  
 Backup Designated router id 131.119.254.28, Interface addr 131.119.254.28  
 Neighbor Count is 8, Adjacent neighbor count is 2  
 Adjacent with neighbor 131.119.254.28 (Backup Designated Router)  
 Adjacent with neighbor 131.119.254.10 (Designated Router)  
 router#

**Relative fields are explained in the following table:**

Domain	Description
Internet Address:	IP address of the interface
Nettype	Network type of the OSPF interface
OSPF process is	ID of the OSPF process
AREA	Current area
Router ID	ID of the router where the process belongs
Cost	Cost of the OSPF interface of the router
Transmit Delay is	Transmission delay
Priority	Priority for the interface of the router
Hello interval	Hello interval
Dead timer	Dead timer
Retransmit	Retransmission interval
OSPF INTF State is	State of the OSPF interface
Designated Router id	ID and IP address of the designated router
Backup Designated router id	ID and IP address of the designated backup router
Neighbor Count is	Number of the neighboring routers
Adjacent neighbor count is	Number of the neighbors where the neighborhood is created
Adjacent with neighbor	Adjacent neighbor list

### 3.1.37 show ip ospf neighbor

To display the information about the adjacent OSPF neighbor, run the following command:

**show ip ospf neighbor**

**Parameter**

None

**Default**

None

**Command mode**

EXEC

**Explanation**

The information displayed by the command can help you to check whether the OSPF neighbor configuration is right and to detect the OSPF faults.

**Example**

```
router#show ip ospf neighbor
OSPF process: 1
AREA 1
Neighbor Pri State DeadTime Address Interface
21.0.0.32 1 FULL /DR 31 192.168.99.32 Ethernet1/0
AREA 36.0.0.1
Neighbor Pri State DeadTime Address Interface
199.199.199.137 1 EXSTART/DR 31 202.19.19.137 Ethernet2/1
AREA 192.168.20.0
Neighbor Pri State DeadTime Address Interface
140.140.0.46 1 FULL /DR 108 140.140.0.46 Serial 1/0
133.133.2.11 1 FULL /DR 110 133.133.2.11 Serial1/0
192.31.48.200 1 FULL / DROTHER 31 192.31.48.200 Ethernet1/0
```

**Relative fields are explained in the following table:**

Domain	Description
OSPF process	ID of the OSPF process
AREA	Current area
Neighbor	ID of the neighbor
Pri	Priority of the neighbor
State	State of the connection with the neighbor
DeadTime	Time when the neighbor invalidates
Address	IP address of the neighbor
Interface	Interface where the router reaches the neighbor

### 3.1.38 show ip ospf virtual-link

To display the information about the OSPF virtual link, run the following command:

**show ip ospf virtual-link**

#### Parameter

None

#### Default

None

#### Command mode

EXEC

#### Explanation

According to the information exported by the command, you can check the state of the OSPF virtual link.

You can run **show ip ospf neighbor** to check the detailed information about the adjacent neighbor.

#### Example

```
router#show ip ospf vir
Virtual Link Neighbor ID 200.200.200.2 (UP)
Run as Demand-Circuit
TransArea: 1, Cost is 185
Hello interval is 10, Dead timer is 40 Retransmit is 5
INTF Adjacency state is IPOINT_TO_POINT
```

Relative fields are explained in the following table:

Domain	Description
neighbor ID	Neighbor ID of the peer
Neighbor state	Neighborhood state for the neighbor
Demand-Circuit	Functioning the DC mode
TransArea	Transmission area
cost	Minimum cost for reaching the peer in the transmission area If the value of the cost is 0, it means that the peer is unreachable.
Hello Interval	Current Hello interval

DeadTime	Time for neighbor invalidation
Retrans	Retransmission interval
INTF Adjacency State	State of the virtual-link interface

Related command

**area virtual-link**

**show ip ospf neighbor**

### 3.1.39 summary-address

To configure the address for OSPF to create the route aggregation, run **summary-address**. To cancel the address of route aggregation, run **no summary-address**.

**summary-address** *address mask* [**not-advertise**]

**no summary-address** *address mask*

Parameter

Parameter	Description
<i>address</i>	Aggregation address with the designated address range
<i>Mask</i>	Subnet mask of the aggregation route
<b>not-advertise</b>	(optional) limits the matched route to generate the LSA.

**Default**

None

Command mode

Routing configuration mode

Explanation

Multiple groups of addresses are summarized. Routes learned from other routing protocols can also be summarized. After the aggregation, all covered networks cannot be transmitted to other routing fields. The cost of the summary route is the minimum value among the cost values of all summary routes. The command cannot be used to reduce the size of the routing table.

The command is used by OSPF to enable the ASBR to notify an external route of being an aggregation route to replace all external routes. The command is only used to

aggregate the OSPF routes of other routing protocols. You can run **area range** in OSPF to summarize the routes.

### Example

In the following example, the summary address 10.1.0.0 stands for addresses such as 10.1.1.0, 10.1.2.0 and 10.1.3.0.

```
summary-address 10.1.0.0 255.255.0.0
```

### Related command

**area range**

**ip ospf password**

**ip ospf message-digest-key**

### 3.1.40 timers delay

To designate a delay interval between OSPF receiving a topology change and starting a shortest path priority calculation, run **timer delay *spf-delay***. To resume the default settings, run **no timers delay**.

**timers delay *spf-delay***

**no timers delay**

### Parameter

Parameter	Description
<i>spf-delay</i>	Delay between the topology change and calculation start. It ranges between 0 and 65535 seconds. Its default value is 5 seconds. If the value is 0, there is no delay. That is, the calculation will be promptly started if changes occur.

### Default

spf-delay: 5 seconds

### Command mode

Routing configuration mode

### Explanation

The smaller value the delay is set to, the faster the network change is reflected. However, it will take the processor more time.

### Example

```
timers spf 10
```

#### 3.1.41 timers hold

To set the interval between two continuous SPF calculations, run **timers hold**. To resume the default settings, run **no timers hold**.

**timers hold** *spf-holdtime*

**no timers hold**

### Parameter

Parameter	Description
<i>spf-holdtime</i>	Minimum value between two continuous calculations. It ranges between 0 to 65535 seconds. Its default value is 10 seconds; when it is 0, there is no interval between the two continuous calculations.

### Default

spf-holdtime: 10 seconds

### Command mode

Routing configuration mode

### Explanation

The smaller value the delay is set to, the faster the network change is reflected. However, it will take the processor more time.

### Example

```
timers spf 20
```

## Chapter 4 BGP Configuration Commands

BGP configuration commands include:

aggregate-address

bgp always-compare-med

bgp bestpath med

bgp client-to-client reflection

bgp cluster-id

bgp confederation identifier

bgp confederation peers

bgp dampening

bgp default

bgp deterministic-med

bgp redistribute-internal

clear ip bgp

debug chat

debug dialer

debug ip bgp

distance

filter

neighbor default-originate

neighbor description

neighbor distribute-list

neighbor ebgp-multihop

neighbor filter-list

neighbor maximum-prefix

neighbor next-hop-self

neighbor password

neighbor prefix-list  
neighbor remote-as  
neighbor route-map  
neighbor route-reflector-client  
neighbor route-refresh  
neighbor send-community  
neighbor shutdown  
neighbor soft-reconfiguration  
neighbor timers  
neighbor update-source  
neighbor weight  
network (BGP)  
redistribute (BGP)  
router bgp  
show ip bgp  
show ip bgp community  
show ip bgp neighbors  
show ip bgp paths  
show ip bgp prefix-list  
show ip bgp regexp  
show ip bgp summary  
synchronization  
table-map  
timers

#### 4.1.1 aggregate-address

To create the aggregation address in the BGP routing table, run **aggregate-address**.  
To disable the function of creating the aggregation address in the BGP routing table, run **no aggregate-address**.

**aggregate-address** *A.B.C.D/n* [**summary-only**] [**route-map** *map-name*]



**no aggregate-address *A.B.C.D/n* [summary-only] [route-map *map-name*]**

### Parameter

Parameter	Description
<b>A.B.C.D/n</b>	Aggregated network
<b>summary-only</b>	Limits all detailed routes.
<b>route-map</b>	Specifies the properties of the aggregation route.
<i>map-name</i>	Name of the route map

### Default

None

### Command mode

BGP configuration mode

### Explanation

There are three methods to add the routes to the BGP routing table:

1. Dynamically add the routes through the **redistribute** command.
2. Statically add the routes through the **network** command.
3. Statically add the routes through the **aggregate** command. The routes generated through the three methods are thought to be the locally-generated routes which can be notified of other peers but cannot be added to the IP routing table.

The aggregation route is adopted to reduce the number of routes in the routing table and to improve the efficiency of route index and the stability of the routes. The BGP aggregation route functions in the BGP routing table. The aggregation route will not add the locally-generated routes to the routing table but can be seen in the BGP routing table.

The aggregation route always aggregates the existing routes according to certain rules. The existence of the aggregation route depends on the state of the source route which generates the aggregation route. The BGP aggregation route depends on the routes with the same prefix or much more precise routes. The aggregation route is effective when at least one route with the same prefix or one more precise route exists in the BGP routing table. The valid aggregation route can be displayed through the **show ip bgp** command. The aggregation route can constrain the source route. If the source route is constrained, it is marked with **s**.

If the **summary-only** option is used, the aggregation route can be created and the more detailed route can be constrained.

The **route-map** option can be used to modify the properties of the route when the aggregation route is generated.

The maximum configuration times for the **aggregate** command are up to the source of the router, such as the configured RAM.

## Example

The following example shows how to establish the aggregation address:

```
router bgp 5
aggregate-address 193.0.0.0/8
```

## Related command

**route-map**

### 4.1.2 bgp always-compare-med

To enable BGP to always compare the MED, run **bgp always-compare-med**.

**bgp always-compare-med**

**no bgp always-compare-med**

## Parameter

None

## Default

The MED of routes from different autonomous systems is not compared by default.

## Command mode

BGP configuration mode

## Explanation

In general, the MED of two routes from the same autonomous system can be compared. If the **bgp always-comapre-med** command is used, BGP always compares MED no matter whether the routes are from the same autonomous system. In this way, the process of the route selection can be modified.

## Example

The following example shows how to enable BGP to compare MED:

```
router bgp 5
bgp always-compare-med
```

## Related command

**bgp bestpath med**

**bgp deterministic-med****4.1.3 bgp bestpath med**

To modify the method for BGP to process the MED of the route, run **bgp bestpath med**. To resume the default method for BGP to process the MED of the route, run **no bgp bestpath med**.

**Parameter**

Parameter	Description
confed	Compares the MED properties of the autonomous system ally.
missing-as-worst	Views the route without the MED property is the worst when the MED property is compared.

**Default**

None

**Command mode**

BGP configuration mode

**Explanation**

By default, if the MED property need be compared when the MED property of the BGP route is not configured, the MED is always thought to be 0, that is, the MED value is the smallest and the MED has the highest priority. After the **missing-as-worst** option is configured, the MED is thought to have the biggest value and has the lowest priority if the MED property of the BGP route is not configured but the MED need be configured.

By default, the same routes released by different BGP routers in the same autonomous system have their MED compared, while the same routes released by different autonomous systems do not have their MED compared. After the **confed** option is configured, the regulation can be modified to enable the routes of the same type in an autonomous system ally to compare MED.

**Example**

Route 100 and route 200 are not from the same autonomous sub-system and they do not compare the MED by default. After **bgp bestpath med confed** is configured, route 100 and route 200 compare MED because they are respectively from autonomous sub-system 100 and autonomous sub-system 200 in the same autonomous system ally.

**Related command**

**bgp always-compare-med**

**bgp deterministic-med**

**4.1.4 bgp client-to-client reflection**

To enable the route reflection from client to client, run **bgp client-to-client reflection**. To disable the route reflection from client to client, run **no bgp client-to-client reflection**.

**bgp client-to-client reflection**

**no bgp client-to-client reflection**

**Parameter**

None

**Default**

After the route reflector is configured, it will reflect the route of one client to other clients by default.

**Command mode**

BGP configuration mode

**Explanation**

If the radiator or the autonomous ally is not configured, all IBGP's in the autonomous system must be completely connected and the neighbor will not report the route received from the IBGP neighbor. The routing loops can thus be prevented. If the route reflector is used, all IBGP's need not be fully connected. After all IBGP's in the autonomous system becomes fully-connected, the route reflection is not required.

**Example**

In the following example, the local router is a route reflector. When three neighbors are fully connected, the route reflection will be closed.

```
router bgp 5
neighbor 192..168.20.190 router-reflector-client
neighbor 192..168.20.191 router-reflector-client
neighbor 192..168.20.192 router-reflector-client
no bgp client-to-client reflection
```

**Related command**

**neighbor route-reflector-client**

**bgp cluster-id**

**4.1.5 bgp cluster-id**

**bgp cluster-id** *cluster-id*

**no bgp cluster-id** *cluster-id*

**Parameter**

Parameter	Description
<i>cluster-id</i>	ID of BGP route reflection cluster, which may be in the format of the IP address or number and is up to four bytes

**Default**

If only one route reflector exists in the BGP route reflection cluster, the ID of the router is the ID of the route reflection cluster.

**Command mode**

BGP configuration mode

**Explanation**

A BGP route reflection cluster consists of one or multiple route reflectors and client machines. In general, a BGP route reflection cluster only has one route reflector. In this case, the ID of the router which acts as the route reflector in the cluster is the ID of the route reflection cluster. To increase the redundancy and prevent the failure of a single node, a cluster may have multiple route reflectors. When you configure the route reflector in the reflection cluster, you must set it with the 4-byte cluster ID so as to the route reflector can identify the update information about other route reflectors in the same cluster.

If a cluster has multiple route reflectors, the command is used to configure the ID of the BGP route reflection cluster. All route reflectors in the same cluster must be set to the same ID.

**Example**

In the following example, the local router acts as the route reflector of the reflection cluster. The BGP reflection cluster ID is used to identify the cluster. Neighbor 198.92.70.24 is the route reflection client.

```
router bgp 5
```

```
neighbor 198.92.70.24 route-reflector-client
bgp cluster-id 50000
```

### Related command

**neighbor route-reflector-client**

**show ip bgp summary**

## 4.1.6 bgp confederation identifier

To designate a BGP autonomous system ID, run **bgp confederation identifier**. To cancel the BGP autonomous system ID, run **no bgp confederation identifier**.

**bgp confederation identifier *autonomous-system***

**no bgp confederation identifier *autonomous-system***

### Parameter

Parameter	Description
autonomous-system	ID of the autonomous system in the autonomous system ally

### Default

None

### Command mode

BGP configuration mode

### Explanation

One method to reduce the IBGP connections is to divide an autonomous system into multiple autonomous sub-systems and then combine them as a single autonomous system ally. The concept of the autonomous system ally is based on several sub-AS's in an AS. In each sub-AS, all IBGP regulations can be applied. For example, all IBGP neighbors must be in the fully-connected architecture. EBGP must run among AS's because each sub-AS has a different AS number. However, the routing choice in the ally is similar to the IBGP routing choice in a single AS. That is, the information about nexthop, MED and Localpreference will be reserved. As to the outside, the whole ally seems like a single AS.

The identifier of the autonomous system ally is a number shown to the outside. All BGP routers in the same autonomous system ally must be set to the same identifier of the autonomous system ally.

To configure the identifier of the autonomous system ally, you need recreate the BGP connection.

## Example

In the following example, the AS is divided into seven sub-AS's: 4001, 4002, 4003, 4004, 4005, 4006 and 4007; the seven sub-AS's are identified with 5, the identifier of the confederation. The local AS is 4001. Neighbor 1.2.3.4 is in the autonomous system ally, while neighbor 3.4.5.6 is outside of the autonomous system ally. As to neighbor 3.4.5.6, your AS is the one identified by the number 5.

```
router bgp 4001
  bgp confederation identifier 5
  bgp confederation peers 4002 4003 4004 4005 4006 4007
  neighbor 1.2.3.4 remote-as 4002
  neighbor 3.4.5.6 remote-as 510
```

## Related command

**bgp confederation peers**

**show ip bgp summary** 30

### 4.1.7 bgp confederation peers

To configure AS belonging to the autonomous system ally, run **bgp confederation peers**. To delete AS from the autonomous system ally, run **no bgp confederation peers**.

**bgp confederation peers** *autonomous-system* [*autonomous-system*]

**no bgp confederation peers** *autonomous-system* [*autonomous-system*]

## Parameter

Parameter	Description
<i>autonomous-system</i>	Number of the autonomous system

## Default

None

## Command mode

BGP configuration mode

## Explanation

The fact that one AS can be divided into several sub-AS's is the base of the autonomous system ally. In each AS, all IBGP regulations can be applied. For example, all IBGP neighbors must combine the fully-connected architecture. EBGP must run among AS's because each AS has a different AS number. However, the routing choice

in the ally is similar to the IBGP routing choice in a single AS. That is, the information about nexthop, MED and Localpreference will be reserved. As to the outside, the whole ally seems like a single AS.

The autonomous system specified by the command is an autonomous subsystem in the same autonomous system ally as to the local autonomous system. Each autonomous subsystem is inward fully-connected.

The **bgp confederation identifier** command is used to specify which autonomous system ally the local AS belongs to.

The command configuration requires reestablishing the BGP connection.

## Example

The following example shows how to classify AS1090, 1091, 1092 and 1093 into a single confederation.

```
router bgp 1090
bgp confederation identifier 23
bgp confederation peers 1091 1092 1093
```

## Related command

**bgp confederation identifier**

**show ip bgp summary**

### 4.1.8 bgp dampening

To configure BGP routing dampening control, run **bgp dampening [route-map name] | [half-time resuse-value suppress-value hold-time]**. To cancel the BGP routing dampening control function, run **no bgp dampening**.

**bgp dampening [route-map name] | [half-time resuse-value suppress-value hold-time]**

**no bgp dampening [route-map name] | [half-time resuse-value suppress-value hold-time]**

## Parameter

Parameter	Description
<b>route-map</b>	Specifies the route map of the BGP route wave control parameter.
<i>name</i>	Sets the name of the route map.
<i>half-time</i>	Means the half punishment time of route attenuation.
<i>reuse-value</i>	Reuses the punishment value of the route.
<i>suppress-value</i>	Suppresses the value of route punishment.
<i>hold-time</i>	Constrains the maximum hold time of the route (unit: minute).



## Default

half-time: 15 minutes  
reuse-value: 750  
suppress-value: 2000  
hold-time: 60 minutes

## Command mode

BGP configuration mode

## Explanation

Route fluctuation control has different effects on the routes in different states, that is, there are effects on whether the routes can be aggregated and whether the route can be added to the main routing table. The fluctuation procedure of the route is described as follows:

A stable route is punished because of its fluctuation. When its punishment value is smaller than the value of the **suppress** parameter, the route can continuously notify the neighbor and can be aggregated. When the punishment value of the route is larger than the value of the **suppress** parameter, the route stops to notify the neighbor and cannot be aggregated. When the route stabilized, the punishment value can decrease along with the time. When the punishment value is larger than the value of the **reuse** parameter, the route is always in the control state and it cannot notify the neighbor and cannot be aggregated. If the punishment value decreases to a value which is smaller than the value of the **reuse** parameter, the route validates and it can notify the neighbor and be aggregated.

## Example

You can enable the BGP route fluctuation control function through the **bgp dampening** command and use the default parameter configuration. You can run the following commands to set different route fluctuation control parameters for different routes:

```
Router bgp 100
bgp dampening route-map DMAP
!
route-map DMAP 10 permit
match as-path ASLIST-1
set dampening 15 750 2000 60
!
route-map DMAP 20 permit
match as-path ASLIST-2
set dampening 2 750 2000 8
!
ip as-path access-list ASLIST-1 permit ^3_
ip as-path access-list ASLIST-2 permit ^5_
```

**Related command**

**set dampening**

**4.1.9 bgp default**

To set the default parameter for the BGP process, run **bgp default**. To resume the default settings, run **no bgp default**.

**bgp default local-preference** <0-4294967295>

**no bgp default local-preference** <0-4294967295>

**Parameter**

Parameter	Description
local-preference	Sets the default parameter of the local priority.
<0-4294967295>	Means the default value of the local priority.

**Default**

The default value of the local priority is 100.

**Command mode**

BGP configuration mode

**Explanation**

BGP sets the routes from the IBGP neighbor to the local priority. The default value is 100. The priority value can be modified through the command.

**Example**

The following example shows how to set the local priority value of the route from IBGP neighbor to 200.

```
router bgp 100
bgp default local-preference 200
```

**Related command**

None

#### 4.1.10 `bgp deterministic-med`

To change the process mode for BGP to process the MED feature, run **`bgp deterministic-med`**. To resume the default value, run **`no bgp deterministic-med`**.

**`bgp deterministic-med`**

**`no bgp deterministic-med`**

##### Parameter

None

##### Default

None

##### Command mode

BGP configuration mode

##### Explanation

By default, BGP compares the MED of routes from different BGP neighbors in the same autonomous system.

##### Example

None

##### Related command

**`bgp bestpath med`**

**`bgp always-compare-med`**

#### 4.1.11 `bgp redistribute-internal`

To allow the routes obtained from IBGP to be added to IGP, such as RIP route or OSPF route, run **`bgp redistribute-internal`**.

**`bgp redistribute-internal`**

**`no bgp redistribute-internal`**

**Parameter**

None

**Default**

The routes obtained from IBGP are not inserted to IGP.

**Command mode**

BGP configuration mode

**Explanation**

When you configure the command, pay attention to the configuration among the routers, or the route loop may occur. After the command is configured, you need run **clear ip bgp \*** to reset BGP.

**Example**

The following example shows that the routes obtained from IBGP will be inserted by BGP into OSPF 3.

```
router ospf 3
 redistribute bgp 2
!
router bgp 2
 bgp redistribute-internal
!
```

**Related command**

None

**4.1.12 clear ip bgp**

To reset the BGP connection with BGP, run the following command:

**clear ip bgp** {*\** | *ip-address* | *as-number* | **peer-group** *name* | **aggregates** | **networks** | **redistribute**} [**soft** [**in** | **out**]]

**Parameter**

Parameter	Description
<i>*</i>	Resets all current BGP sessions.
<i>ip-address</i>	Resets only the designated BGP neighbor.

<b>AS</b>	Resets the neighbor with the designated autonomous system.
<i>peer-group-name</i>	Resets the designated BGP peer group.
<b>aggregates</b>	Resets all aggregation routes.
<b>Networks</b>	Resets all static network routes.
<b>redistribute</b>	Resets all forwarding routes.
<b>soft</b>	Means soft re-configuration.
<b>in   out</b>	Means soft re-configuration for the incoming or outgoing route.

## Command mode

EXEC

## Explanation

Some BGP strategy configuration cannot validate immediately because the route is transmitted only once in a BGP session. To reset the BGP session, the routing information need be retransmitted.

If the BGP soft reconfiguration designated by the **soft** keyword is passed, the session will not be reset and the router will retransmit the update information about all routes. To prevent the update information of the incoming route from being generated by BGP session resetting, the local BGP session should receive all unchanged update information through the **neighbor soft-reconfiguration** command no matter whether the update information is allowed by the incoming strategy. Because the whole process takes a large storage volume, the whole process should be avoid as much as possible. The outgoing BGP soft configuration does not require extra memory cost. You can trigger an outgoing reconfiguration at the peer of the BGP session to validate new incoming strategy.

When the **aggregates/networks/redistribute** option is used, the **soft** option cannot be used because the **aggregates/networks/redistribute** option can delete the route with designated type and regenerate routes to validate the new configuration.

## Example

The following example shows how to reset all current BGP sessions.

```
clear ip bgp *
```

## Related command

**neighbor soft-reconfiguration**

**show ip bgp**

### 4.1.13 debug chat

To trace the script's activity, run **debug chat**. You can run **no debug chat** to stop displaying the information about script operation.

**debug chat**

**no debug chat**

#### Parameter

The command has no parameters or keywords.

#### Command mode

EXEC

#### Example

```
Router#debug chat
Router#SCRIPT: start script default_dialer_script...
SCRIPT:Sending string: ATZ
SCRIPT:Expecting string: OK
SCRIPT: Receive string:
41 54 0D 0D 0A 4F 4B 0D 0A AT...OK..
SCRIPT:Completed match for expect:OK
SCRIPT:Sending string: ATDT 2
SCRIPT:Expecting string: CONNECT
SCRIPT: Receive string:
43 4F 4E 4E 45 43 54 CONNECT
SCRIPT: Completed match for expect:CONNECT
SCRIPT:Chat script finished
```

The first information line means that the **default\_dialer\_script** script is being started.

The second information line means that the ATZ character string is transmitted.

The third information line shows that the OK character string is being waited.

The fourth information line shows that the expected OK character string is received.

The fifth information line shows that the ATDT 2 character string is transmitted.

The sixth information line shows that the CONNECT character string is being waited.

The seventh information line shows that the expected CONNECT character string is received.

The eighth information line shows that the script is successfully executed.

**Related command**

**chat-script**

**4.1.14 debug dialer**

To trace the dial-up procedure and the dial-up activity, such as initializing the modem and DDR startup dial-up, run **debug dialer**. To stop displaying relative information, run **no debug dialer**.

**debug dialer**

**no debug dialer**

**Parameter**

The command has no parameters or keywords.

**Command mode**

EXEC

**Example**

```
Router#debug dialer
DIALER Serial 1/0: Dialing cause ip(PERMIT).
DIALER Serial 1/0: Dialing using Modem script: default_dialer_script & System script: none
DIALER Serial 1/0: Attempting to dial 2
DIALER Serial 1/0: process started
DIALER Serial 1/0: Chat script default_dialer_script (dialer) started.....
DIALER Serial 1/0: Connection established
DIALER Serial 1/0: Modem script finished successfully
```

The first information line shows that the dialer checks whether the packet triggers the dialing. The IP packet can trigger the dialer.

The second information line shows that the modem script used during the dial-up is the predefined default dialer script. The system script is not used.

The third information line shows that the dialer number is 2.

The fourth information line shows that the dialer process is started.

The fifth information line shows that the dialer script is started and the modem is then answered and dialed.

The sixth and seventh information lines show that the dialer script is successfully executed and the call is successful.

### 4.1.15 debug ip bgp

To enable the BGP trace function, run **debug ip bgp**. To disable the BGP trace function, run **no debug ip bgp**.

**debug ip bgp {all | fsm | keepalive | open | update}**

**no debug ip bgp {all | fsm | keepalive | open | update}**

#### Parameter

Parameter	Description
<b>All</b>	Opens all trace functions of BGP.
<b>dampening</b>	Opens the BGP routing fluctuation control trace.
<b>Event</b>	Opens the event trace of BGP.
<b>Fsm</b>	Opens the state machine trace of BGP.
<b>Keepalive</b>	Opens the Keepalive trace of BGP.
<b>notify</b>	Opens the Notify packet trace of BGP.
<b>Open</b>	Opens the trace of the <b>open</b> packet.
<b>Update</b>	Opens the trace of the <b>update</b> packet.

#### Default

All trace functions are closed.

#### Command mode

EXEC

#### Explanation

The trace functions are globally effective. After the trace functions are enabled, the trace information will be displayed at the monitor port. If other VTYs open the terminal monitor function, the trace information will also be displayed. In this case, you can run **no terminal monitor** to disable the function to forbid displaying the trace information.

The **debug ip bgp all** function is used to open all BGP trace functions, including **dampening**, **fsm**, **keepalive**, **open** and **update**. You can run **no debug ip bgp all** to close all opened BGP trace functions.

#### Example

The following example shows how an BGP connection is established. The trace information shows that the router initiates a connection to the BGP neighbor 10.1.1.3.



The connection starts with the **idle** state and finally ends with the **established** state which means that the connection is established.

The format of the trace information consists of several key parts. The initial part is the time formation if the systematic configuration allows the time information in the trace information. The real BGP information is the BGP header, the address of the BGP neighbor and relative BGP events.

```
BGP: 10.1.1.3 start connecting to peer
BGP: 10.1.1.3 went from Idle to Connect
BGP: 10.1.1.3 went from Connect to OpenSent
BGP: 10.1.1.3 send OPEN, length 41
BGP: 10.1.1.3 rcv OPEN, length 41
BGP: 10.1.1.3 went from OpenSent to OpenConfirm
BGP: 10.1.1.3 send KEEPALIVE, length 19
BGP: 10.1.1.3 rcv KEEPALIVE, length 19
BGP: 10.1.1.3 went from OpenConfirm to Established
BGP: 10.1.1.3 send KEEPALIVE, length 19
BGP: 10.1.1.3 send UPDATE, length 43
BGP: 10.1.1.3 send UPDATE, length 43
BGP: 10.1.1.3 rcv KEEPALIVE, length 19
BGP: 10.1.1.3 rcv KEEPALIVE, length 19
```

#### 4.1.16 distance

To modify the management distance of the default external/internal/local route and realize the management strategy, run **distance bgp external-distance internal-distance local-distance**. To resume the default values, run **no distance bgp**.

**distance bgp** *external-distance internal-distance local-distance*

**no distance bgp**

#### Parameter

Parameter	Description
<i>external-distance</i>	Means the management distance of the external BGP route which is the best route learned by the external route from the external AS neighbor. The default value is 20.
<i>internal-distance</i>	Means the management distance of the internal BGP route which is the route learned by the internal route from other BGP entities in the same AS. The default value is 200.
<i>local-distance</i>	Means the management distance of the external BGP route. The backdoor route configured by the network command is the local route. The default value is 200.

## Default

external-distance: 20

internal-distance: 200

local-distance: 200

## Command mode

BGP configuration mode

## Explanation

Through the **distance** command, you can modify the management distance of the route, the priority of the route and the routing choice and thus change the route strategy.

It is dangerous to modify the management distance of the route until you have the clear purpose. The routing table may become identical and the route may be damaged.

## Example

In the following example, the already known internal route is more appreciate than that learned through IGP. That's why the management distance is set.

```
router bgp 109
network 131.108.0.0
neighbor 129.140.6.6 remote-as 123
neighbor 128.125.1.1 remote-as 47
distance 20 20 200
```

## Related command

**set metric**

**set tag**

### 4.1.17 filter

To enable the port-based route filtration, run **filter**. To disable it, run **no filter**.

**filter** *interface* <in | out> **access-list** *access-list-name* **gateway** *access-list-name*  
**prefix-list** *prefix-list-name*

**no filter** *interface* <in | out> **access-list** *access-list-name* **gateway** *access-list-name*  
**prefix-list** *prefix-list-name*

## Parameter

Parameter	Description
<b>Interface</b>	Name of the port The symbol (*) stands for all ports.
<b>in   out</b>	Filtrates the received or transmitted routes.
<b>access-list</b>	Designates the access list which is used to filtrate the routes.
<i>access-list-name</i>	Name of the access list
<b>Gateway</b>	Designates the gateway where the access list is used to filtrate the routes.
<i>access-list-name</i>	Name of the access list
<b>prefix-list</b>	Designates the prefix list to filtrate the routes.
<i>prefix-list-name</i>	Name of the prefix list

## Default

None

## Command mode

BGP configuration mode

## Explanation

The **access-list** option is used to designate the access list to filtrate the network prefix information of the route. The **gateway** option is used to designate the access list to filtrate the next hop of the route and the **prefix-list** option is used to designate the prefix list to filtrate the network prefix information.

The **access-list** option and the **prefix-list** option can not be used concurrently. They can be used together with the **gateway** option. The route, however, need pass through two examinations in this case.

The symbol (\*) stands for all interfaces. To configure the filtration regulations on a specific interface and then on all interfaces, the route must meet all filtration regulations.

If the inexistent **prefix-list** option and **access-list** option are designated, all routes are allowed to pass.

## Example

In the following example, the received routes on all ports are filtrated through the **prefix-list** and **gateway** options.

```
router bgp 109
filter * in prefix-list prefix-guize gateway gateway-guize
```

**Related command****neighbor distribute-list****neighbor filter-list****neighbor route-map****4.1.18 neighbor default-originate**

To enable the BGP session party (local router) to transmit default route 0.0.0.0 to the designated neighbor, run **neighbor default-originate**; to disable the default route to be transmitted, run **no neighbor default-originate**.

**neighbor** {*ip-address* | *peer-group-name*} **default-originate**

**no neighbor** {*ip-address* | *peer-group-name*} **default-originate**

**Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group

**Default**

The default route is not transmitted to the neighbor.

**Command mode**

BGP configuration mode

**Explanation**

The following three conditions are needed when the neighbor transmits the default route:

1. Default routes exist in the routing table.
2. Default routes are inserted to BGP.
3. Default route transmission is configured on the neighbor. Network 0.0.0.0/0 can be used when the default route is inserted into BGP.

The command has no business with route 0.0.0.0 generation in the BGP routing table.

**Example**

In the following example, route 0.0.0.0 exists in the routing table of the local router, the local router notifies neighbor 160.89.2.3.

```
router bgp 109
network 160.89.0.0
neighbor 160.89.2.1 remote-as 100
neighbor 160.89.2.3 remote-as 200
neighbor 160.89.2.3 default-originate
```

### Related command

**neighbor ebgp-multihop**

#### 4.1.19 neighbor description

To describe the neighbor, run **neighbor description**. To delete the command description, run **no neighbor description**.

**neighbor** {*ip-address* | *peer-group-name*} **description** **LINE**

**no neighbor** {*ip-address* | *peer-group-name*} **description** **LINE**

### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<b>Line</b>	Neighbor description line

### Default

There is no description about the neighbor.

### Command mode

BGP configuration mode

### Explanation

The description can make the configuration more accessible.

### Example

In the following example, the description of the neighbor is the peer of **abc.com**.

```
router bgp 109
network 160.89.0.0
neighbor 160.89.2.3 description peer with abc.com
```

#### 4.1.20 neighbor distribute-list

To filtrate the incoming/outgoing routes of the BGP neighbor through the access list, run **neighbor distribute-list**. To delete the previous configuration, run **no neighbor distribute-list**.

**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list name* } {**in** | **out**}

**no neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list name* } {**in** | **out**}

##### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>access-list name</i>	Name of the access list
<b>In</b>	Means that the access list is used to filtrate the incoming routes.
<b>Out</b>	Means that the access list is used to filtrate the outgoing routes.

##### Default

None

##### Command mode

BGP configuration mode

##### Explanation

One method is to use the **access-list** option to filtrate the network prefix information of the BGP route through the **neighbor distribute-lists** command; one method is to use the **aspath-list** option to filtrate the AS\_PATH feature of the BGP route through the **neighbor filter-list** command; the other method is to use the **prefix-list** option to filtrate the network prefix information of the BGP route through the **neighbor prefix-list** command.

If the inexistent access list is designated, all routes are allowed.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command. The command to designate the IP address will replace the values inherited from the peer group.

##### Example

The following example shows how to apply the Beijing list for filtrating the incoming routes of neighbor 120.23.4.1.

```
router bgp 109
network 131.108.0.0
neighbor 120.23.4.1 distribute-list beijing in
```

### Related command

**ip aspath-list**

**neighbor filter-list**

**ip prefix-list 1**

**neighbor prefix-list**

#### 4.1.21 neighbor ebgp-multihop

To permit EBGp neighbors in the indirectly-connected network, run **neighbor ebgp-multihop**. To resume the default settings, run **no neighbor ebgp-multihop**.

**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]

**no neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**

### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP session neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>ttl</i>	Number of hops ranging between 1 and 255

### Default

The EBGp neighbor only allows the direct connection. The value of TTL is 1 and the value of the TTL neighbor is 255.

### Command mode

BGP configuration mode

### Explanation

By default, the EBGp neighbor must be in the directly-connected network, or the BGP connection cannot be established. The maximum number of hops of the EBGp neighbor can be set through the **neighbor ebgp-multihop** command.

If the value of TTL is not specified by the command, the value of TTL is 255.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

### Example

The following example shows the connection with neighbor 131.108.1.1 is allowed though the neighbor is not in the directly-connected network.

```
router bgp 109:
neighbor 131.108.1.1 ebgp-multihop
```

### Related command

**neighbor default-originate**

#### 4.1.22 neighbor filter-list

To configure the AS-PATH list to filtrate the incoming/outgoing routes of the BGP neighbor, run **neighbor filter-list**. To forbid the previous function, run **no neighbor filter-list**.

**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *as-path-list name* {**in** | **out**}

**no neighbor** {*ip-address* | *peer-group-name*} **filter-list** *as-path-list name* {**in** | **out**}

### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>as-path-list name</i>	Name of the AS-PATH list which can be defined by the <b>ip as-path-list</b> command
<b>In</b>	Filtrates the incoming routes.
<b>Out</b>	Filtrates the outgoing routes.

### Default

None

### Command mode

BGP configuration mode



## Explanation

One method is to use the **access-list** option to filtrate the network prefix information of the BGP route through the **neighbor distribute-lists** command; one method is to use the **aspath-list** option to filtrate the AS\_PATH feature of the BGP route through the **neighbor filter-list** command; the other method is to use the **prefix-list** option to filtrate the network prefix information of the BGP route through the **neighbor prefix-list** command.

If the inexistent access list is designated, all routes are allowed.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command. The command to designate the IP address will replace the values inherited from the peer group.

## Example

In the following example, the routes forwarded through or coming from AS123 are not reported to neighbor 128.125.1.1.

```
ip as-path-list shanghai deny _123_  
ip as-path-list shanghai deny ^123$  
  
router bgp 109  
network 131.108.0.0  
neighbor 129.140.6.6 remote-as 123  
neighbor 128.125.1.1 remote-as 47  
neighbor 128.125.1.1 filter-list shanghai out
```

## Related command

**ip aspath-list**

**neighbor distribute-list**

**ip prefix-list**     **1**

**neighbor prefix-list**

### 4.1.23 neighbor maximum-prefix

To control the maximum number of network prefixes obtained from the neighbor, run **neighbor maximum-prefix**. To forbid the function, run **no neighbor maximum-prefix**.

**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum*

**no neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix**

**Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<b>peer-group-name</b>	Name of the BGP peer group
<i>Maximum</i>	Maximum number of network prefixes obtained from the neighbor

**Default**

There is no limitation to the number of network prefixes.

**Command mode**

BGP configuration mode

**Explanation**

The command allows to configure the maximum number of network prefixes obtained by the BGP router from the peer and provides a mechanism to control the prefix reception.

When the number of the received prefixes reaches the configured maximum number, the router terminates the session.

**Example**

The following example shows that the maximum number of prefixes obtained from neighbor 129.140.6.6 is set to 1000.

```
router bgp 109
network 131.108.0.0
neighbor 129.140.6.6 maximum-prefix 1000
```

**Related command**

**clear ip bgp**

**4.1.24 neighbor next-hop-self**

To activate the next-hop process of the BGP update and set itself as the address of the next hop, run **neighbor next-hop-self**. To disable the function, run **no neighbor next-hop-self**.

**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**

**no neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP session neighbor
<i>peer-group-name</i>	Name of the BGP peer group

## Default

The function is disabled by default.

## Command mode

BGP configuration mode

## Explanation

The Nexthop process in BGP is more complicated than that in IGP. During the process, three regulations must be followed: 1. As to the EBGp session, when the route is transmitted, set Nexthop to the local IP address of the BGP connection.

2. As to the IBGP session, if the route is generated locally, when the route is transmitted, the nexthop should be set to the local IP address of the BGP connection; if the route is obtained by EBGp, write the nexthop feature to the package without any change when the route is transmitted.

3. If the IP address of the nexthop feature of the route belongs to the network where the BGP session resides, the nexthop features are always the features of the previous nexthop.

The command is useful in the NBMA network because the BGP neighbor in the NBMA network probably cannot access other neighbors on the same IP subnet.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command. The command to designate the IP address will replace the values inherited from the peer group.

## Example

The following example shows that the next hop' addresses of all route updates which is mandatorily transmitted to router 131.108.1.1 are set to the routes themselves.

```
router bgp 109
neighbor 131.108.1.1 next-hop-self
```

## Related command

**set ip next-hop 18**

#### 4.1.25 neighbor password

To enable the MD5 option of TCP to perform the password authentication between the BGP neighbors, run **neighbor password**. To cancel the authentication, run **no neighbor password**.

**neighbor** {*ip-address* | *peer-group-name*} **password** *LINE*

**no neighbor** {*ip-address* | *peer-group-name*} **password**

##### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<b>password</b>	Performs the MD5 authentication.
<i>LINE</i>	Text password

##### Default

None

##### Command mode

BGP configuration mode

##### Explanation

Before the command is used, the neighbors must be designated by the **neighbor remote-as** command.

The command must be configured on the two neighborhood parties and the same password must be set. In this case, the neighbor connection can be established through MD5 authentication. The password contains any character except space and the length of the password must range between 1 and 20 characters.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

##### Example

The following example shows how to set the authentication password of neighbor 120.23.4.1 to **abcd**:

```
router bgp 109
 neighbor 120.23.4.1 remote-as 108
 neighbor 120.23.4.1 password abcd
```

**Related command****neighbor remote-as**4.1.26 **neighbor prefix-list**

To configure the prefix list to filtrate the route update of the neighbor, run **neighbor prefix-list**. To delete the previous configuration, run **no neighbor prefix-list**.

**neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-listname* {**in** | **out**}

**no neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-listname* {**in** | **out**}

**Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<b>prefix-list</b>	Applies the prefix list to the route update of the neighbor.
<i>prefix-listname</i>	Name of the prefix list.
<b>In</b>	Applies the incoming route update to the neighbor.
<b>Out</b>	Applies the outgoing route update to the neighbor.

**Default**

None

**Command mode**

BGP configuration mode

**Explanation**

One method is to use the **access-list** option to filtrate the network prefix information of the BGP route through the **neighbor distribute-lists** command; one method is to use the **aspath-list** option to filtrate the AS\_PATH feature of the BGP route through the **neighbor filter-list** command; the other method is to use the **prefix-list** option to filtrate the network prefix information of the BGP route through the **neighbor prefix-list** command.

If the inexistent access list is designated, all routes are allowed.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command. The command to designate the IP address will replace the values inherited from the peer group.

## Example

The following example shows how to apply **prefix list abc** to the incoming route update of neighbor 120.23.4.1.

```
router bgp 109
network 131.108.0.0
neighbor 120.23.4.1 prefix-list abc in
```

The following example shows how to apply **prefix list CustomerA** to the incoming route update of neighbor 120.23.4.1.

```
router bgp 109
network 131.108.0.0
neighbor 120.23.4.1 prefix-list CustomerA in
```

## Related command

**ip prefix-list**

**ip prefix-list description**

**ip prefix-list sequence-number**

**show ip prefix-list**

**clear ip prefix-list**

**neighbor filter-list**

### 4.1.27 neighbor remote-as

To establish the BGP neighbor and designate its autonomous system number, run **neighbor remote-as**. To cancel the neighbor and its configuration, run **no neighbor remote-as**.

**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*

**no neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>Number</i>	Number indicating which AS the neighbor belongs to

## Default

None

## Command mode

BGP configuration mode

## Explanation

The neighbor with the same AS number designated by the **router bgp** command is considered as IBGP, or the neighbor is considered as EBGP. The **neighbor remote-as** command is used to create the neighbor. Only after the neighbor is created, other commands relative with the neighbor can be configured. If the neighbor is configured, the number of the autonomous system can be modified and the BGP connection can be reset.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

## Example

In the following example, the local autonomous system is 109. Neighbors 131.108.200.1, 131.108.234.2 and 150.136.64.19 are configured and their autonomous systems are 167, 109 and 99.

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

## Related command

**neighbor peer-group (creating)**

### 4.1.28 neighbor route-map

To set the route map to filtrate the incoming and outgoing routes of the neighbor, run **neighbor route-map**. To cancel the configuration, run **no neighbor route-map**.

**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {*in* | *out*}

**no neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {*in* | *out*}

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>map-name</i>	路由映射名。

<b>in</b>	Means that the command is applied to the incoming routes.
<b>Out</b>	Means that the command is applied to the outgoing routes.

## Default

None

## Command mode

BGP configuration mode

## Explanation

The commands **distribute-list**, **prefix-list** and **as-path-list** can filtrate the routes based on the neighbors, while the **route-map** command can not only filtrate the routes based on the neighbor but also change the features of the route, enabling flexible routing strategies.

Different routes have different features and the route map can change the features of various routes. If the match-up regulation is configured for the route features that are not supported by the BGP route or the regulations are set and applied to the BGP route, these regulations will be omitted. The following regulations are valid when they are applied to the BGP route: match aspath-list, match community-list, match ip address, match ip nexthop, match ip prefix-list, match metric, match tag, set aggregator, set as-path, set atomic-aggregate, set community, set community-additive, set ip nexthop, set local-preference, set metric, set origin, set tag, set weight

If the inexistent route map is configured, all routes are allowed and have no changes.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command. The command to designate the IP address will replace the values inherited from the peer group.

## Example

The following example shows how to apply the **internal-map** route map to the incoming route from neighbor 198.92.70.24.

```
router bgp 5
neighbor 198.92.70.24 route-map internal-map in
route-map internal-map
match as-path abc
set local-preference 100
```

## Related command

**neighbor peer-group (creating)**

**route-map 1**



#### 4.1.29 neighbor route-reflector-client

To set the local router to be the BGP route reflector and designate the neighbor as the client, run **neighbor route-reflector-client**. To cancel the created client, run **no neighbor route-reflector-client**. When all clients are invalid, the local router is not the route reflector any more.

**neighbor** *ip-address* **route-reflector-client**

**no neighbor** *ip-address* **route-reflector-client**

##### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP neighbor

##### Default

There is no route reflector.

##### Command mode

BGP configuration mode

##### Explanation

By default, all IBGP session parties in AS are fully-connected; BGP session parties do not report the routes learned from the IBGP neighbor.

If the route reflector is used, all IBGP session parties need not be fully-connected. In route reflector mode, the route reflector transmits the route (learned from IBGP) to the clients. The solution deserts the necessity for each router to communicate with other routers.

The **neighbor route-reflector-client** command is used to set the local router to be the BGP route reflector and designate the neighbor as one client. All neighbors configured by the command are the members of the client group. The left IBGP peers are the members of the non-client group in the local route reflector.

##### Example

In the following example, the local router is a route reflector which transmits the learned IBGP route to neighbor 198.92.70.24.

```
router bgp 5
neighbor 198.92.70.24 route-reflector-client
```

##### Related command

**bgp cluster-id**

**show ip bgp****4.1.30 neighbor route-refresh**

To activate the route refresh function of the neighbor, run **neighbor route-refresh**. To disable the route refresh function, run **no neighbor route-refresh**.

**neighbor *ip-address* route-refresh**

**no neighbor *ip-address* route-refresh**

**Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the BGP neighbor

**Default**

The route refresh function is not run by default.

**Command mode**

BGP configuration mode

**Explanation**

By default, the BGP route is only exchanged once when the connection is created and afterwards only the modified route is exchanged. If the configuration of the route strategy is modified, it cannot validate immediately. In general, there are two methods: resetting the BGP connection and activating the soft-reconfiguration function. The first method is slow and the route change may be big; the second method requires plenty of storage space and takes more time. The two methods are not so good as route refreshing.

Route refreshing is a negotiation option during BGP connection establishment which is to require neighbors retransmitting all UPDATE packets through sending the route refresh request. In this case, resetting BGP connection or storing lots of routes is not required.

**Example**

In the following example, neighbor 198.92.70.24 is allowed to activate route refreshing.

```
router bgp 5
neighbor 198.92.70.24 route-refresh
```

**Related command**

**show ip bgp neighbors**

### 4.1.31 neighbor send-community

To allow the route updates with the community attribute to be sent to the BGP neighbor, run **neighbor send-community**.

**neighbor** {*ip-address* | *peer-group-name*} **send-community**

**no neighbor** {*ip-address* | *peer-group-name*} **send-community**

#### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group

#### Default

The route update transmitted to the neighbor can own the community attributes.

#### Command mode

BGP configuration mode

#### Explanation

BGP always receives the routes with community attributes. You can run **no neighbor send-community** to forbid the routes with community attributes to be sent to the neighbor.

The community attributes of the routes can be set through the **route-map** command or the **set community** command, or come from the route notification of the neighbor.

You can run **show ip bgp neighbors** to check whether the community attributes are allowed to send to the neighbor.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

#### Example

In the following example, the router belongs to AS109 and forbids the **communities** attributes to neighbor 198.92.70.23.

```
router bgp 109
no neighbor 198.92.70.23 send-community
```

**Related command****match community-list 4****neighbor peer-group (creating)****set community 15****set community-additive 17****4.1.32 neighbor shutdown**

To invalidate neighbors or peer groups, run **neighbor shutdown**. To reactivate neighbors or peer groups, run **no neighbor shutdown**.

**neighbor {ip-address | peer-group-name} shutdown****no neighbor {ip-address | peer-group-name} shutdown****Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group

**Default**

None

**Command mode**

BGP configuration mode

**Explanation**

The **neighbor shutdown** command is used to shut down the session of the designated neighbor or the peer group and delete all relative routing information. In the case of peer groups, it means that a lot of sessions terminate abruptly.

You can run **show ip bgp summary** or **show ip bgp neighbors** to check the information about the BGP neighbors and the peer groups. The neighbor that is closed by the **neighbor shutdown** command is in the **shutdown** state.

**Related command****show ip bgp summary****show ip bgp neighbors**

### 4.1.33 neighbor soft-reconfiguration

To enable the route update storage function, run **neighbor soft-reconfiguration**. To delete the route update and stop storing the route update, run **no neighbor soft-reconfiguration**.

**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [*inbound*]

**no neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [*inbound*]

#### Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP session neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<b>inbound</b>	Stores the incoming route update.

#### Default

The incoming route update is not stored but the outgoing route update is stored.

#### Command mode

BGP configuration mode

#### Explanation

The outgoing route updates are always stored, while the incoming route updates are stored only after relative configuration is performed. The stored route updates can validate after the routing strategy is modified without BGP session resetting. The BGP session resetting will bring lots of data exchange and many routes will therefore fluctuate, while soft-reconfiguration can avoid the previous shortness.

The outgoing route updates are always stored, while the incoming route updates are not stored by default. To validate the new configuration after the local configuration strategy is modified, you can adopt the following methods:

Firstly, resetting related BGP sessions; secondly, locally conducting the soft reconfiguration of the incoming route, **clear ip bgp a.b.c.d soft in**; thirdly, the peer conducting the soft reconfiguration of the outgoing route, **clear ip bgp a.b.c.d soft out**.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

## Example

The following example shows that the inbound soft-reconfiguration of neighbor 131.108.1.1 is activated and all route updates received from the neighbor will be stored without any change.

```
router bgp 100
neighbor 131.108.1.1 remote-as 200
neighbor 131.108.1.1 soft-reconfiguration inbound
```

## Related command

**clear ip bgp**

**neighbor peer-group (creating)**

### 4.1.34 neighbor timers

To set the timer for a detailed BGP peer or peer group, run **neighbor timers**. To cancel the timer of a detailed BGP peer or peer group, run **no neighbor timers**.

**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive* *holdtime*

**no neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive* *holdtime*

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP peer
<i>peer-group-name</i>	Name of the BGP peer group
<b>Keepalive</b>	Value of the keepalive timer whose unit is second
<i>Holdtime</i>	Value of the Holdtime timer whose unit is second, which is 0 or a value larger than 3

## Default

Keepalive is 60 seconds.

Holdtime is 180 seconds.

## Command mode

BGP configuration mode

## Explanation

The timer of a detailed neighbor or peer group replaces the timer of the default BGP neighbor. In general, the **holdtime** parameter is set to a value three times of the value of the **keepalive** parameter. If the **keepalive** parameter and the **holdtime** parameter are set to 0, the keepalive packets are forbidden to forward. In this case, the TCP connection administrator is required to notify the BGP module of the connection's state change.

## Example

The following example shows how to respectively set the keepalive timer and the holdtime timer of BGP peer 192.98.47.10 to 70 seconds and 210 seconds.

```
router bgp 109
neighbor 192.98.47.10 timers 70 210
```

### 4.1.35 neighbor update-source

To allow the BGP session to create the TCP connection through the designated interface, run **neighbor update-source**. To resume the autonomously-selected interface, run **no neighbor update-source**.

**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*

**no neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the BGP session neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<b>Interface</b>	Name of the interface

## Default

The IP address of the local interface which is calculated by the route is used to create the TCP connection.

## Command mode

BGP configuration mode

## Explanation

By default, the IP module decides the local IP address of the TCP connection when BGP triggers the connection. The IP module decides the outgoing interface through the route, binds the main IP address of the interface and takes it as the local address

of the TCP connection. The **update-source** function can be used to bind the main IP address of the designated local interface when the TCP connection is established.

The loopback interface is normally designated, because the loopback interface's protocol state is always up to stabilize the BGP session and prevent route fluctuation.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

## Example

The following example shows the BGP connection of the neighbor is designated to use the IP address of the loopback interface.

```
router bgp 110
network 160.89.0.0
neighbor 160.89.2.3 remote-as 110
neighbor 160.89.2.3 update-source Loopback0
```

## Related command

**neighbor peer-group (creating)**

### 4.1.36 neighbor weight

To endow the BGP connection with a weight value, run **neighbor weight**. To delete the endowed weight value, run **no neighbor weight**.

**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*

**no neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the neighbor
<i>peer-group-name</i>	Name of the BGP peer group
<i>Weight</i>	Weight value, ranging from 0 to 65535

## Default

The default weight value of the route which is obtained from the BGP peer is 0, while the default weight value of the route generated by the local router is 32768.

## Command mode

BGP configuration mode



## Explanation

The weight value of the BGP route is an important element to consider when you choose the route. The default weight values of all neighbor-learned routes are 0. Through the command, you can set the weight value for a route from a neighbor. The route map is another method to modify the weight value.

If the **peer-group-name** parameter is used to designate the BGP peer group, all members of the peer group will inherit all the features configured by the command.

## Example

The following example shows that the weight value of the route learned by neighbor 151.23.12.1 is set to 50:

```
router bgp 109 neighbor 151.23.12.1 weight 50
```

## Related command

**neighbor peer-group (creating)**

**set weight 23**

### 4.1.37 network (BGP)

To insert the network route into BGP, run **network**. To cancel the configuration, run **no network**.

**network A.B.C.D/n route-map map-name backdoor**

**no network A.B.C.D/n route-map map-name backdoor**

## Parameter

Parameter	Description
<b>A.B.C.D/n</b>	Adds the network prefix to BGP.
<b>route-map</b>	Designates the route map.
<i>map-name</i>	Name of the route map
<b>backdoor</b>	Stands for the backdoor network.

## Default

By default, no network prefix is added to BGP.

## Command mode

BGP configuration mode

## Explanation

There are three methods to add the routes to the BGP routing table:

1. Dynamically add the routes through the **redistribute** command.
2. Statically add the routes through the **network** command.
3. Statically add the routes through the **aggregate** command. The routes generated through the three methods are thought to be the locally-generated routes which can be notified of other peers but cannot be added to the IP routing table.

The premise for the network configured by the **network** command to validate is that a same route exists in the main IP routing table.

The premise for the network configured by the **network** command to validate is that at least one more accurate route or a same route exists in the local BGP routing table.

If the mask's length is not designated, the length will be generated according to the settings of the standard network type.

The route-map can be used to set the attributes of the route when the route is generated.

The backdoor network is not used to generate the route, but to modify the distance of the route. The default distance of the route from the neighbor is changed to the distance of the local route whose default value is 200.

The maximum configuration times for the **network** command are up to the source of the router, such as the configured NVRAM or RAM.

## Example

The following example shows how to add route 131.108.0.0/8 to BGP:

```
router bgp 120
network 131.108.0.0/8
```

## Related command

**redistribute (BGP)**

**aggregate-address**

### 4.1.38 redistribute (BGP)

To add a route to BGP, run **redistribute**. To forbid the route to be added to BGP, run **no redistribute**.

**redistribute** *protocol* [*process-id*] [*route-map map-name*]

**no redistribute** *protocol* [*process-id*] [*route-map map-name*]

## Parameter

Parameter	Description
<b>protocol</b>	Type of the routing protocol
<i>process-id</i>	Process ID of the routing protocol
<b>route-map</b>	Sets the route's attribute through the route map.
<i>map-name</i>	Name of the route map

## Default

The route forwarding is forbidden.

## Command mode

BGP configuration mode

## Explanation

There are three methods to add the routes to the BGP routing table:

1. Dynamically add the routes through the **redistribute** command.
2. Statically add the routes through the **network** command.
3. Statically add the routes through the **aggregate** command. The routes generated through the three methods are thought to be the locally-generated routes which can be notified of other peers but cannot be added to the IP routing table.

You can run redistribute command to dynamically add the route to BGP. The change of the route source will be autonomously reflected in BGP. Other neighbors will be notified of the dynamically-added routes. The routes of the designated type in the routing table will be rechecked after the **redistribute** command is set. The exterior routes in OSPF will not be added to BGP.

The route-map can be used to set the attributes of the route when the route is generated.

## Example

The following example shows how to forward OSPF process 23 to BGP:

```
router bgp 109
 redistribute ospf 23
```

## Related command

**route-map 1**

### 4.1.39 router bgp

To start the BGP process or enter the BGP mode, run **router bgp**. To disable the BGP process, run **no router bgp**.

**router bgp** *as-number*

**no router bgp** *as-number*

#### Parameter

Parameter	Description
<i>as-number</i>	Number of the autonomous system

#### Default

The BGP process is shut down by default.

#### Command mode

Global configuration mode

#### Explanation

Only one BGP process can be configured in the system. The BGP task in the system is created when the system is initialized. After the BGP process is started, the BGP task is activated. If the BGP process is not configured, the BGP task only receives the information from the command module, which has no relation with the routing module and other modules. Relative commands **show** and **clear** are invalid.

The BGP process can be deleted through the **no router bgp** command. At the same time, the other configurations such as neighbors will also be deleted. The BGP route in the routing table should also be deleted.

After the BGP process is configured, you can run **show running** or **show ip bgp summary** to observe it.

#### Example

The following example shows how to start the BGP process and designate the autonomous system's number to 200.

```
router bgp 200
```

#### Related command

**neighbor remote-as**

#### 4.1.40 show ip bgp

To display the items in the BGP routing table, run **show ip bgp**.

**show ip bgp** [*network*]

##### Parameter

Parameter	Description
<i>network</i>	Displays the designated routing information.

##### Command mode

EXEC

##### Explanation

If the network is not designated, the whole BGP routing table is not displayed. After the network is designated, only the information about the network is displayed.

##### Example

The information about a BGP group is displayed in the following. The first two lines shows some mark information.

Status code shows the meaning of the mark in front of the route; the letter “s” means that the route is restrained by the aggregation configuration (the restrained route is an invalid route); the letter “H” stands for the history route which is saved because of the route fluctuation (In fact, a real route does not exist. The history route is also not a valid route); the mark “\*” stands for a valid route which can be selected as the optimal route; the mark “>” stands for the optimal route which is chosen from all valid routes; the letter “i” stands for an internal route which is from the IBGP neighbor.

Origin codes show the **origin** attribute of the route. The letter “i” stands for IGP, the letter “e” stands for EGP and the mark “?” stands for uncertainty.

The following attributes of each route will be displayed: state, destination address, gateway’s address, MED, local-preference, weight and AS path. The gateway address of the locally-generated route is 0.0.0.0. If the metric is not well configured, it will not be displayed, or its value will be displayed. The default value of the **local-preference** parameter for the IBGP route is 100. The other **local-preference** parameters which are not displayed also contain the default value or the set value will be displayed. The weight of the locally-generated route is 32768, set value or 0. The **AS Path** domain will display the **AS Path** attribute of the route including the AS list and the **origin** attribute. The content in the bracket is AS-set or the sub autonomous system in the autonomous system ally.

The last line shows how many routes are displayed, including all invalid or valid routes.

B3710\_118#show ip bgp

Status codes: s suppressed, d damped, h history, \* valid, > best, i internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
* 192.168.10.0/24	192.168.69.5			0	10 400 i
*>i192.168.10.0/24	192.168.69.14	100		0	(65030) 400 i
*>i192.168.11.0/24	192.168.69.14	100		0	(65030) 400 i
* 192.168.65.0/30	192.168.69.1	100		0	(65020) 10 ?
*> 192.168.65.0/30	192.168.69.5			0	10 ?
* 192.168.65.4/30	192.168.69.1	100		0	(65020) 10 ?
*> 192.168.65.4/30	192.168.69.5			0	10 ?
* 192.168.65.8/30	192.168.69.1	100		0	(65020) 10 ?
*> 192.168.65.8/30	192.168.69.5			0	10 ?
* 192.168.66.0/30	192.168.66.2	100		0	(65020) ?
*> 192.168.66.0/30	0.0.0.0			32768	?
* i192.168.66.4/30	192.168.66.6	100		0	?
*> 192.168.66.4/30	0.0.0.0			32768	?
*>i192.168.66.8/30	192.168.66.6	100		0	?
*>i192.168.67.0/30	192.168.69.18	200	100	0	500 ?

Number of displayed routes: 15

## Related command

**show ip bgp community**

**show ip bgp neighbors**

**show ip bgp paths**

**show ip bgp prefix-list**

**show ip bgp regexp**

**show ip bgp summary**

### 4.1.41 show ip bgp community

To display the statistics information about the BGP community structure, run **show ip bgp community**.

**show ip bgp community**

## Parameter

None

## Command mode

EXEC

**Explanation**

The command is used to display the statistics information about the BGP community structure.

**Related command**

**show ip bgp**

**show ip bgp neighbors**

**show ip bgp paths**

**show ip bgp prefix-list**

**show ip bgp regexp**

**show ip bgp summary**

4.1.42 **show ip bgp neighbors**

To display information about the neighbors, run **show ip bgp neighbors**.

**show ip bgp neighbors** [*ip-address*] [**received-routes** | **routes** | **advertised-routes**]

**Parameter**

Parameter	Description
<i>ip-address</i>	IP address of the neighbor If the parameter is omitted, all neighbors will be displayed.
<b>received-routes</b>	Displays all routes received from the designated neighbor (received or declined).
<b>routes</b>	Displays all routes which are accepted after they are received from the designated neighbor.
<b>advertised-routes</b>	Displays all routes which are reported by the router to the neighbor.

**Command mode**

EXEC

**Explanation**

You can check the detailed information about the neighbor, its current state, and some configuration information. The routes relative with the neighbor will be displayed after corresponding keywords are designated.

**Related command**

**show ip bgp**

**show ip bgp community**

**show ip bgp paths**

**show ip bgp prefix-list**

**show ip bgp regexp**

**show ip bgp summary**

#### 4.1.43 **show ip bgp paths**

To display the statistics information about the BGP path's structure, run **show ip bgp paths**.

**show ip bgp paths**

##### **Parameter**

None

##### **Command mode**

EXEC

##### **Explanation**

The command is used to display the statistics information about the BGP path's structure.

##### **Related command**

**show ip bgp**

**show ip bgp community**

**show ip bgp neighbors**

**show ip bgp prefix-list**

**show ip bgp regexp**

**show ip bgp summary**

#### 4.1.44 **show ip bgp prefix-list**

To display the BGP routing information which matches the designated prefix list, run **show ip bgp prefix-list**.

**show ip bgp prefix-list {*prefix-list name*}**



**Parameter**

Parameter	Description
<i>prefix-list name</i>	Name of the prefix list

**Command mode**

EXEC

**Explanation**

The command is used to filter the content displayed by the **show ip bgp** command through designating the prefix list. Only the routes which match the prefix list can be displayed.

**Related command**

**show ip bgp**  
**show ip bgp community**  
**show ip bgp neighbors**  
**show ip bgp prefix-list**  
**show ip bgp regexp**  
**show ip bgp summary**  
**ip prefix-list**  
**ip prefix-list description**  
**ip prefix-list sequence-number**  
**show ip prefix-list**  
**clear ip prefix-list**

4.1.45 **show ip bgp regexp**

To display the routes which match the designated regular express, run the following command:

**show ip bgp regexp *regular-expression***

**Parameter**

Parameter	Description
<i>regular-expression</i>	Regular express of the AS path

**Command mode**

EXEC

**Explanation**

The command can filter the content displayed by the **show ip bgp** command through the designated regular expression of the AS path. The routes which match the regular expression can be displayed.

**Related command****show ip bgp****show ip bgp community****show ip bgp neighbors****show ip bgp prefix-list****show ip bgp regexp****show ip bgp summary****4.1.46 show ip bgp summary**

To display the summary information about all BGP connections, run **show ip bgp summary**.

**show ip bgp summary****Parameter**

The command has no parameters or keywords.

**Command mode**

EXEC

**Explanation**

You can run **show ip bgp summary** to check the global configuration of the BGP protocol.

**Example**

The following information is displayed after the **show ip bgp summary** command is run.

```

router bgp 4
BGP local AS is 4
Router ID is 192.168.20.72
IGP synchronization is enabled
Distance: external 20 internal 200

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/Pref
192.168.20.12	4	5	0	0	0	0	0	never	Connect

### Related command

```

show ip bgp
show ip bgp community
show ip bgp neighbors
show ip bgp paths
show ip bgp prefix-list
show ip bgp regexp
show ip bgp summary

```

#### 4.1.47 synchronization

To enable the synchronization function between BGP and IGP, run **synchronization**.  
To disable the function, run **no synchronization**.

```

synchronization
no synchronization

```

### Parameter

None

### Default

Synchronization is effective.

### Command mode

BGP configuration mode

## Explanation

IGP synchronization means that a route which is received by BGP from IBGP will be reported to other EBGP neighbors after the route appears in the routing table in form of IGP. If the IGP synchronization does not work, the route received by BGP from IBGP will be reported to other EBGP neighbors no matter whether the route is in form of BGP. The IGP referred here include the straight-through route, static route, RIP route, OSPF route and routes relative with other internal gateway protocols.

By default, the IGP function is enabled.

Different from the **synchronization** command, the **no synchronization** command can be used to report the network route no matter whether IGP exists or not.

## Example

The following example shows that the router begins to broadcast the routes without waiting for the IGP synchronization.

```
router bgp 120
no synchronization
```

## Related command

**router bgp**

### 4.1.48 table-map

To set the route-map which is added to the routing table and modify some attributes of the route, run **table-map**. To delete the configuration, run **no table-map**.

**table-map** <name>

**no table-map** <name>

## Parameter

Parameter	Description
<i>name</i>	Name of the route map

## Default

None

## Command mode

BGP configuration mode

**Explanation**

Through setting the table map, you can filter the routes or modify their attributes when BGP adds the routes to the routing table.

**Example**

None

**Related command**

None

**4.1.49 timers**

To modify the default timers of the BGP neighbor, run **timers bgp <keepalive> <holdtime>**. To resume the default value, run **no timers bgp <keepalive> <holdtime>**.

**timers bgp <keepalive> <holdtime>**

**no timers bgp <keepalive> <holdtime>**

**Parameter**

Parameter	Description
<i>keepalive</i>	Default keepalive interval of the BGP neighbor
<i>holdtime</i>	Default holdtime interval of the BGP neighbor

**Default**

Keepalive: 60 seconds

Holdtime: 180 seconds

**Command mode**

BGP configuration mode

**Explanation**

You can modify the default timer settings by globally configuring the timer of the BGP neighbor. The neighbor's settings are prior to the global settings.

**Example**

The following example shows that the default timer is set to 10 and 40.

```
router bgp 100  
timers bgp 10 40
```

**Related command**

**neighbor timers**

## Chapter 5 Public Routing Configuration Commands

### 5.1 Ip aspath-list Configuration Commands

#### 5.1.1 ip as-path access-list

To create the as-path list, run **ip as-path access-list <name> <deny | permit> <regex>**. To cancel the configured as-path list, run **no ip as-path access-list <name> [deny | permit] [regex]**.

**ip as-path access-list <name> <deny | permit> <regex>**

**no ip as-path access-list <name> [deny | permit] [regex]**

#### Parameter

Parameter	Description
<i>name</i>	Name of the as-path list
<b>deny   permit</b>	Attribute of the as-path list
<i>regex</i>	Regular expression of the as-path

#### Default

All as-path expressions except those having a clear explanation on the **permit** regulation are declined by default.

#### Command mode

Global configuration mode

#### Explanation

The AS-path list is used to filter the AS-PATH attribute of the BGP route. The AS-PATH attribute of the BGP route is a number sequence which is expressed in form of the character string. The number at the right end is the autonomous system number for the route starting, while the numbers leftwards in turn are the numbers of the autonomous systems which the BGP route passes. For example, character string 22 23 98 means that the BGP route is transmitted from autonomous system 98, passes through autonomous system 23 and autonomous system 22, and finally reaches the local autonomous system.

The AS-path list in the system is identified with the name. The total number of AS-path lists which are allowed to configure in the system is limited by the resource of the system. The same AS-path list can be configured with multiple matchup regulation.

The procedure to apply the AS-path list is to check whether the match-up is successful or not according to the configuration order. Once a match-up is found to be successful, the following check-up will be stopped and the nature of the regulation (deny/permit) is then returned. If the match-up of all regulations is not successful, the nature of the regulation, **deny**, will be returned. Each regulation is organized according to their configuration order.

The as-path expression is normally the regular expression. The special characters which are always used in the expression are shown in the following table:

Character	Symbol	Meaning
Full stop	.	Matches any single character, including space.
Asterisk	*	Matches the <b>0</b> sequence or more sequences.
Plus	+	Matches the <b>1</b> sequence or more sequences.
Question mark	!	Follows the number 0 or 1.
Addition character	^	Starting point of the match-up character string
Dollar	\$	End point of the match-up character string
Underline	—	Matches these symbols: “””, “{ }”, “( )”, “^”, “\$” and “space”.
Square bracket	[ ]	Stands for the range of the single-character mode.
Hyphen	-	Separates a range.

With the aid of the presentation methods of the AS-PATH attribute, the correct usage of the regular expression can help create the powerful AS-path list. The following examples are given:

- . \*           Representing any attribute of the AS path.
- ^\$           Representing the attributes of the null path
- ^22\$           Representing the path attributes of autonomous system 22
- ^22\_           Representing the path attribute starting with 22
- \_22\$           Representing the path attribute starting with 22, such as 22, 34 22 and 99 45 22
- \_22\_           Representing the path attribute containing 22, like 23 22 45 and 442 22 23 44

The **as-path list** command can be used together with the **match as-path** command and the **neighbor filter-list** command.

### Example

In the following example, the defined **as-path list hell** command permits all path attributes starting with 23 or containing 22:

```
ip as-path access-list hell permit ^23
ip as-path access-list hell permit _22_
```



Or:

```
ip as-path access-list guangzhou deny ^300
ip as-path access-list guangzhou deny _300_
ip as-path access-list guangzhou permit .*
```

The AS-PATH attributes starting with 300 or containing 300 will be declined, while other AS-PATH attributes can pass. If the defined order is different, the results will be totally different. The following AS-PATH attributes can pass.

```
ip as-path access-list guangzhou permit .*
ip as-path access-list guangzhou deny ^300
ip as-path access-list guangzhou deny _300_
```

Related command

**match as-path**

**neighbor filter-list**

### 5.1.2 show ip aspath-list

To display the AS-path list configured in the system, run the following command:

**show ip as-path-list** *<name>*

Parameter

Parameter	Description
<i>name</i>	Name of the as-path list

Default

None

Command mode

EXEC

Explanation

If the name of the as-path list is not designated, all configured as-path lists in the system will be displayed.

举例

The following example shows that all as-path lists in the system will be displayed:

```
show ip as-path-list
```

Related command

**ip as-path access-list**

## 5.2 ip community-list Configuration Commands

### 5.2.1 ip community-list

To create the regulations for the community list of the BGP route, run **ip community list**. To cancel the regulations for the community list, run **no community list**.

**ip community-list** <name> <deny | permit> [aa:nn | 1-4294967295 | local-AS | no-advertise | no-export ]

**no ip community-list** <name> <deny | permit> [aa:nn | 1-4294967295 | local-AS | no-advertise | no-export ]

Parameter

Parameter	Description
<i>name</i>	Name of the community list
<b>deny   permit</b>	Attribute of the community list
<1-4294967295>	Value of the community, which is a 32-bit integer
<i>aa:nn</i>	New form of the community value <b>aa</b> stands for a 16-bit value. <b>nn</b> stands for the next 16-bit value.
<b>no-advertise</b>	Means that no neighbor will be reported.
<b>local-AS</b>	Means that the EBGp neighbor outside of the local autonomous system or in the same autonomous system ally will not be reported.
<b>no-export</b>	Means that the neighbors in the local autonomous system or the autonomous system ally will not be reported.

Default

All communities except those having a clear explanation on the **permit** regulation are declined by default.

Command mode

Global configuration mode

Explanation

The community list is used to filter or set the community attribute of the BGP route. The community attribute is a group number or a community group number. A

community number is a 4-byte value. The community numbers between 0x00000000 and 0x0000FFFF or between 0xFFFF0000 and 0xFFFFFFFF are reserved. These community numbers are globally accepted. The frequently-used community numbers are the following ones:

**NO\_EXPORT (0xFFFFFFFF01)**: After the route with this community number is received, the peers outside the autonomous system or autonomous system ally will not be reported.

**NO\_ADVERTISE(0xFFFFFFFF02)**: After the route with the community number is received, no peers will be reported.

**NO\_EXPORT\_SUBCONFED (0xFFFFFFFF03)**: It is always called as LOCAL\_AS. After the route with the community number is received, the peers outside the local autonomous system are not reported.

The community list in the system is identified by a name. The total number of the community lists which can be configured in the system is limited by the system's resource. The same community list can be configured with multiple matchup regulations. The procedure to apply the community list is to check whether the matchup is successful or not according to the configuration order. Once a matchup is found to be successful, the following check-up will be stopped and the nature of the regulation (deny/permit) is then returned. If the matchup of all regulations is not successful, the nature of the regulation, **deny**, will be returned. The order to check each regulation is the configuration order.

One community-list regulation has three elements: name, regulation's attribute (deny/permit) and community number sequence. The community number sequence is a set of a group of community numbers. If all community numbers in the community attribute are in the community sequence with designated regulations, the matchup is successful. If not, the matchup fails and the next regulation will be matched.

The **community list** command can be used together with commands **route-map** and **match community**.

### Example

In the following example, the community will be declined by the **ip community-list yall** command if the value of the community is 5 or 10; the community will be accepted by the **ip community-list yall** command if the value of the community is 15 or 20.

```
ip community-list yall deny 5 10
ip community-list yall permit 15 20
```

### Related command

**match community-list 4**

## 5.2.2 show ip community-list

To display the community list configured in the system, run the following command:

**show ip community-list <name>**

**Parameter**

Parameter	Description
<i>name</i>	Name of the community list

**Default**

None

**Command mode**

EXEC

**Explanation**

If the name of the community list is not designated, all configured community lists in the system will be displayed.

**Example**

The following example shows that all community lists in the system will be displayed:

```
Show ip community-list
```

**Related command**

**ip community-list**

## 5.3 ip prefix-list commands

### 5.3.1 clear ip prefix-list

To delete the statistics information about the designated prefix list, run the following command:

**clear ip prefix-list** [<name> [<prefix>]]

**Parameter**

Parameter	Description
<i>name</i>	Name of the prefix list
<i>prefix</i>	Network prefix which is in the A.B.C.D/n format n here stands for the length of the mask.

**Default**

None

**Command mode**

EXEC

**Explanation**

If the prefix is not designated, all statistics information in the prefix list will be canceled.

**Example**

None

**Related command****ip prefix-list description****ip prefix-list sequence-number****show ip prefix-list****clear ip prefix-list****5.3.2 ip prefix-list**

To establish a prefix list or add a prefix-list regulation, run **ip prefix-list <name> [<seq> <seq\_number>] <deny | permit> <prefix | any> [<ge> <value>] [<le> <value>]**. To cancel the configuration, run **no ip prefix-list <name> [<seq> <seq\_number>] <deny | permit> <prefix | any> [<ge> <value>] [<le> <value>]**.

**ip prefix-list <name> [<seq> <seq\_number>] <deny | permit> <prefix | any> [<ge> <value>] [<le> <value>]**

**no ip prefix-list <name> [<seq> <seq\_number>] <deny | permit> <prefix | any> [<ge> <value>] [<le> <value>]**

**Parameter**

Parameter	Description
name	Name of the prefix list
seq	Designates the sequence number.
seq_number	Value of the sequence number
deny   permit	Attribute of the prefix list

prefix   any	Designated prefix or any prefix
ge	Designates the minimum length of the matched prefix.
value	Length of the prefix which ranges from 0 to 32
le	Designates the maximum length of the matched prefix.
value	Length of the prefix which ranges from 0 to 32

## Default

None

## Command mode

Global configuration mode

## Explanation

The prefix list is a set of regulations for filtrating the network prefix. Each regulation has five elements: sequence, deny/permit, prefix and length (a.b.c.d/n), upper limitation (le y) and bottom limitation (ge x). All regulations are sorted according to the sequence. When the prefix list is applied, the regulation of the smallest sequence is first checked. If the matchup is successful, other regulations stop the matchup operation and the matched regulation's attribute (deny/permit) is returned.

When you check whether a regulation matches a designated network prefix, you should not only check the length of the network prefix but also check whether the network prefixes have the same length in the designated length. For example, to check whether a regulation of a prefix list, **ip prefix-list test seq 5 A.B.C.D/M ge X le Y**, matches the designated network **a.b.c.d/n**, the following procedure will be taken.

First, check whether the mask length of the network (n) meets the requirement of the expression:  $X \leq n \leq Y$  (if **ge X** is not designated, the expression is **M <= n <= Y**; if the **le Y** is not designated, the expression is **X <= n <= 32**; if both **ge X** and **le Y** are not designated, the expression is **n == M**). If the mask length meets the requirements of the expression, the next operation will be performed. If the mask length does not meet the regulation, the following regulation will be used.

Check whether network a.b.c.d/n and the first M bit of A.B.C.D are same. If they are same, the regulation is met and the attribute of the regulation is returned; if the regulation is not met, the next regulation will be seen whether it is met.

If all regulations are not met, the **deny** attribute will be returned.

## Example

The following are destination routes and prefix lists:

Destination route 1: 120.120.0.0/14

Destination route 2: 120.120.0.0/16

Destination route 3: 120.120.0.0/25

Destination route 4: 130.130.0.0/16

Destination route 5: 130.130.0.0/8

Destination route 6: 130.130.0.0/24

Destination route 7: 12.0.0.0/8

Prefix-list:

ip prefix-list sample permit 120.120.0.0/8 ge 16 le 24

ip prefix-list sample deny 130.130.0.0/16

The following are the matchup results:

Destination route 1: unsuccessful, deny

Destination route 2: successful, permit

Destination route 3: unsuccessful, deny

Destination route 4: successful, deny

Destination route 5: unsuccessful, deny

Destination route 6: unsuccessful, deny

Destination route 7: unsuccessful, deny

## Related command

**ip prefix-list description**

**ip prefix-list sequence-number**

**show ip prefix-list**

**clear ip prefix-list**

### 5.3.3 ip prefix-list description

To configure the description of the prefix list, run **ip prefix-list <name> <description> <strings>**. To cancel the description of the prefix list, run **no ip prefix-list <name> <description> <strings>**.

**ip prefix-list <name> <description> <strings>**

**no ip prefix-list <name> <description>**

## Parameter

Parameter	Description
-----------	-------------

<i>name</i>	Name of the prefix list
<b>description</b>	Designates the description information of the prefix list.
<i>strings</i>	Description information

**Default**

None

**Command mode**

Global configuration mode

**Explanation**

None

**Example**

The following example shows how to add the description information to **prefix-list hard** for convenient reading:

```
ip prefix-list hard deny any
```

```
ip prefix-list hard description This prefix-list is used to filter routes from neighbor hard
```

**Related command**

```
ip prefix-list description
```

```
ip prefix-list sequence-number
```

```
show ip prefix-list
```

```
clear ip prefix-list
```

**5.3.4 ip prefix-list sequence-number**

To enable the prefix list to use the sequence, run **ip prefix-list sequence-number**. To cancel the sequence, run **no ip prefix-list sequence-number**.

```
ip prefix-list sequence-number
```

```
no ip prefix-list sequence-number
```

**Parameter**

None



## Default

The sequence is used by default.

## Command mode

Global configuration mode

## Explanation

The command is used to decide whether each regulation of the prefix list has been allocated with a sequence. After the sequence is used, the same sequence corresponds to only one regulation. Hence, if a regulation with a same sequence is newly generated, the previously old regulation will be deleted. If the sequence is not used, you have to run a command to delete the regulation. The sequence may not be designated during configuration. The system then allocates the sequence for all regulations. The sequence starts from 5 and adds 5 each time.

## Example

None

## Related command

**ip prefix-list description**

**ip prefix-list sequence-number**

**show ip prefix-list**

**clear ip prefix-list**

### 5.3.5 show ip prefix-list

To display the information about the prefix list or all prefix lists, including the configuration information and statistics information about the prefix list, run the following command:

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

## Parameter

Parameter	Description
<b>summary</b>	Summary information
<b>detail</b>	Detailed information
<i>name</i>	Name of the prefix list

**Default**

None

**Command mode**

EXEC

**Explanation**

If the name of the prefix list is not designated, all information about the prefix list will be displayed.

**Example**

The following example shows that a prefix list is configured.

```
ip prefix-list yell permit 130.12.19.0/24
ip prefix-list yell permit 140.20.0.0/16 ge 16 le 24
```

The following information is shown after the **show ip prefix-list detail** command is run:

```
Prefix-list with the last deletion/insertion: yell
ip prefix-list yell: 2 entries
count: 2, range entries: 1, sequences: 5 - 10
seq 5 permit 130.12.19.0/24 (hit count: 0, refcount: 10)
seq 10 permit 140.20.0.0/16 ge 16 le 24 (hit count: 0, refcount: 10)
```

The first information line indicates that the recently-modified prefix list is **yell**.

Starting from the second information line, all information about the prefix list is listed. Here only one prefix list is configured, whose name is **yell**.

Count: 2, indicating that the prefix list has two options.

Range entries: 1, indicating that the number of network range defined in the prefix list is 1.

Sequences: 5-10, indicating the sequence range of each option in the prefix list

The following are the definition of each option and the statistics information.

Hit count: 0, indicating that the times of option matchup is 0

Refcount: 10, meaning that the times of option matchup are 10

**Related command**

**ip prefix-list description**

**ip prefix-list sequence-number**

**show ip prefix-list**

**clear ip prefix-list**

## 5.4 route-map Commands

### 5.4.1 route-map

To create a route map or define a route-map item, run **route-map [name seq] [deny | permit]**. To delete the created route map or the defined route-map item, run **no route-map [name seq] [deny | permit]**.

**route-map [name seq] [deny | permit]**

**no route-map [name seq] [deny | permit]**

#### Parameter

Parameter	Description
<i>name</i>	Name of the route map
<i>seq</i>	Sequence of the route map whose default value is 0
<b>deny   permit</b>	Attribute of the route map whose default value is <b>permit</b>

#### Default

By default, the value of the **seq** parameter is 10 and the attribute is **permit**.

#### Command mode

Global configuration mode

#### Explanation

The route map is used to modify the route's attribute or the filtration route. The route map is always used for the strategy of the dynamic routing protocol, such as redistribute route, filtration route, setting the route's attribute for strategic routing, and so on.

The same route map may have multiple items. The total number of the route map in the system is limited by the system's resource.

Each item in the same route map can be designated with the sequence or the system will automatically generate the sequence for each item.

Each item has a kind of attribute (deny/permit); each item can be conducted with the matchup regulation (match), regulations (set) and exit strategies (on-match).

The matchup regulation is used to check whether a feature of an object meets a certain rule. If the object meets all matchup regulations in the item, the object matches the item successfully, or the item matchup fails. If an item is not configured with the matchup regulation, any object cannot match the item. If the matchup regulation

adopts other lists such as the access list, prefix list, community list or as-path list to check whether an object is matched, the returned value of the list is the result of regulation matchup.

The setting regulation is used to set an attribute of an object. If an object matches the item successfully and the attribute of the item is **permit**, the setting regulations configured under the item are used to modify the attribute of the object. If the object matches the item and the attribute of the item is **deny**, the exit strategy will be checked. If the object fails to match the item, the next item matchup will be conducted until the matchup succeeds.

The exit strategy decides the actions after the object matches the item successfully. If an object matches an item successfully and the item have not configured with the exit strategy, the checking to other items will be stopped and the attribute of the item (deny/permit) will be returned. If **on-match next** is configured, the checking on the next item will be continued. If **on-match goto N** is configured, the designated item, item N, will be the first one to be checked; if the designated item does not exist, the attribute of the item (deny/permit) will be returned.

Under the same item, only one matchup regulation of the same attribute or the settings regulation can be configured. The following matchup regulation or settings regulation configured will replace the previous one. The following configuration can be done for the same item:

```
match metric 34
```

```
set metric 100
```

In the previous example, there is only one **match** regulation and the **set** regulation.

To realize multiple values for matching the same attribute, you can use the exit regulations.

```
route-map match-multi-metric 10 permit
match metric 10
on-match goto 30
route-map match-multi-metric 20 permit
match metric 20
on-match goto 30
route-map match-multi-metric 30 permit
set metric 100
```

In the same example, the route whose metric is 10 or 20 is matched and its metric will be set to 100.

During configuration, the system can automatically generate a sequence for each item, starting from 10 by default and then adding 10 in turn. When the route map is applied, the system will check the sequence of the item from small to big.

The route map can handle different types of routes, some **match** regulations and **set** regulations only suitable for parts of routes. If you try to use the unsupported **match** regulations or **set** regulations to match or modify the objects, the system will omit these regulations.

If there is no name behind the **no route map** command, the whole route map will be deleted, or the designated item will be deleted.

## Example

The following example shows the route map is used to filter the routes forwarded by OSPF and to set the relative attributes.

```
router bgp 20
 redistribute ospf 3 route-map redist-ospf
 route-map redist-ospf
  match tag 139009
  set local-preference 300
```

## Related command

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

**show route-map**

### 5.4.2 match as-path

To set a **match** regulation of the route map and check the attributes of the BGP route through the AS-path map, run **match as-path <as-path-list-name>**. To delete the configuration you have just done, run **no match as-path <as-path-list-name>**.

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

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

#### Parameter

Parameter	Description
<i>as-path-list-name</i>	Name of the as-path list

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

The designated AS path list is used to match the object or to filter the AS-PATH attribute of the BGP route.

#### Example

The following example shows how to check the whether the BGP route is matched using **as-list1**.

```
route-map match-asp  
match as-path as-list1
```

#### Related command

**route-map**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

**show route-map**

### 5.4.3 match community

To set a **match** regulation of the route map and check the attributes of the BGP route through the community list, run **match community <community-list-name>**. To delete the configuration you have just done, run **no match community <community-list-name>**.

**match community <community-list-name>**

**no match community <community-list-name>**

#### Parameter

Parameter	Description
<i>community-list-name</i>	Name of the community list

#### Default

None

#### Command mode

Route-map configuration mode

## Explanation

The designated community list is used to match the object and to filter the community attribute of the BGP route.

## Example

The following example shows how to check the whether the BGP route is matched using **as-list1**.

```
route-map match-comm  
match community comm-list1
```

## Related command

- route-map**
- match as-path**
- match ip address**
- match ip next-hop**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set as-path**
- set atomic-aggregate**
- set community**
- set community-additive**
- set ip next-hop**
- set local-preference**
- set metric**
- set origin**
- set tag**
- set weight**
- show route-map**



#### 5.4.4 match ip address

To set a route-map **match** regulation and match the destination network's address, run **match ip address <name>**. To delete the configuration you have just done, run **no match ip address <name>**.

**match ip address <name>**

**no match ip address <name>**

##### Parameter

Parameter	Description
<i>name</i>	Name of the IP access list

##### Default

None

##### Command mode

Route-map configuration mode

##### Explanation

The access list is used to filter the network address of the route, which is suitable for all IP routes and packets.

##### Example

In the following example, the route checked by the access list is set to metric.

```
route-map set-metric
match ip address acl-metric
set metric 100
```

##### Related command

**route-map**

**match as-path**

**match community-list**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

**show route-map**

#### 5.4.5 match ip next-hop

To set a route-map **match** regulation and check whether the next hop address of the route matches with the address of the designated next hop, run **match ip next-hop <a.b.c.d>**. To delete the configuration you have just done, run **no match ip next-hop <a.b.c.d>**.

**match ip next-hop <a.b.c.d>**

**no match ip next-hop <a.b.c.d>**

#### Parameter

Parameter	Description
<i>a.b.c.d</i>	IP address

#### Default

None

#### Command mode

Route-map configuration mode

## Explanation

The access list is used to check the attribute of the next hop, which is suitable for all IP routes.

## Example

In the following example, the route with the next hop's address 192.121.13.28 matches item 20 of the route map.

```
route-map beijing 10 permit
match ip nexthop 172.12.29.98
set metric 100
route-map beijing 20 permit
match ip nexthop 192.121.13.28
set metric 20
```

## Related command

- route-map**
- match as-path**
- match community-list**
- match ip address**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set as-path**
- set atomic-aggregate**
- set community**
- set community-additive**
- set ip next-hop**
- set local-preference**
- set metric**
- set origin**
- set tag**

**set weight**

**show route-map**

#### 5.4.6 match ip address prefix-list

To set a route-map **match** regulation and match the destination network's address, run **match ip address prefix list <name>**. To delete the configuration you have just done, run **no match ip address prefix-list <name>**.

**match ip address prefix-list <name>**

**no match ip address prefix-list <name>**

#### Parameter

Parameter	Description
<i>name</i>	Name of the prefix list

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

This command is suitable to all IP routes.

#### Example

The following example shows that the route whose destination address is 192.121.0.0 matches **route-map match-prefix**.

```
ip prefix-list beijing permit 192.121.0.0/16
route-map match-prefix
match ip address prefix-list beijing
set metric 100
```

#### Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**  
**match ip next-hop**  
**match metric**  
**match tag**  
**on-match**  
**set aggregator**  
**set as-path**  
**set atomic-aggregate**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.7 match length

To set a route-map **match** regulation and check whether the route's metric matches the address of the designated metric, run **match length <minimum-length> <maximum-length>**. To delete the configuration you have just done, run **no match length <minimum-length> <maximum-length>**.

**match length <minimum-length> <maximum-length>**

**no match length <minimum-length> <maximum-length>**

#### Parameter

Parameter	Description
<i>minimum-length</i>	Minimum length of the packet
<i>maximum-length</i>	Maximum length of the packet

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is suitable to the strategy route.

**Related command**

**route-map**

**5.4.8 match metric**

To set a route-map **match** regulation and check whether the route's metric matches the address of the designated metric, run **match metric <value>**. To delete the configuration you have just done, run **no match metric <value>**.

**match metric <value>**

**no match metric <value>**

**Parameter**

Parameter	Description
<i>value</i>	Metric value

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is suitable to all routes.

## Example

The following example shows that the routes whose metric values are 120 are declined because they match item 20 of the route map.

```
route-map beijing 10 permit
match ip nexthop 172.12.29.98
set metric 100
route-map beijing 20 deny
match ip metric 120
```

## Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

**show route-map**

### 5.4.9 match tag

To set a route-map **match** regulation and check whether the route's tag matches the designated tag, run **match tag <value>**. To delete the configuration you have just done, run **no match tag <value>**.

**match tag <value>**

**no match tag <value>**

#### Parameter

Parameter	Description
<i>value</i>	Value of the Tag

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

This command is suitable to all routes.

#### Example

The following example shows that the routes whose tags' values are 120923 are declined because they match item 20 of the route map.

```
route-map huang 10 permit
match ip nexthop 172.12.29.98
set metric 100
route-map huang 20 deny
match ip tag 120923
```

#### Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**



**match ip prefix-list**  
**match metric**  
**on-match**  
**set aggregator**  
**set as-path**  
**set atomic-aggregate**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.10 on-match

To configure the exit strategy of the route-map item, run **on-match {next | goto *n*}**. To cancel the configuration, run **no on-match {next | goto *n*}**.

**on-match {next | goto *n*}**

**no on-match {next | goto *n*}**

#### Parameter

Parameter	Description
<i>n</i>	Sequence of the item

#### Default

None

#### Command mode

Route-map configuration mode

## Explanation

The command is used to configure the exit strategy of the route-map item. If a route-map item is successfully matched and the item has not configured with the exit strategy, the checking to other items will be stopped and the attribute of the item (deny/permit) will be returned. If **on-match next** is configured, the checking on the next item will be continued. If **on-match goto N** is configured, the designated item, item N, will be the first one to be checked; if the designated item does not exist, the attribute of the item (deny/permit) will be returned.

## Example

The following example shows that all routes are set to **aggregator**.

```
route-map huang
set aggregator as 200 192.12.90.82
```

## Related command

- route-map
- match as-path
- match community-list
- match ip address
- match ip next-hop
- match ip prefix-list
- match metric
- match tag
- set aggregator
- set as-path
- set atomic-aggregate
- set community
- set community-additive
- set ip next-hop
- set local-preference
- set metric
- set origin
- set tag

**set weight****show route-map****5.4.11 set aggregator**

To configure a route-map setting regulation and set the BGP route to **aggregator**, run **set aggregator <as> <as-number> <a.b.c.d>**. To delete the configuration you have just done, run **no set aggregator <as> <as-number> <a.b.c.d>**.

**set aggregator <as> <as-number> <a.b.c.d>****no set aggregator <as> <as-number> <a.b.c.d>****Parameter**

Parameter	Description
<i>as-number</i>	Number of the autonomous system of the route aggregator
<i>a.b.c.d</i>	IP address of the route aggregator

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is only suitable to the BGP route.

**Example**

The following example shows that all routes are set to **aggregator**.

```
route-map huang
set aggregator as 200 192.12.90.82
```

**Related command****route-map****match as-path****match community-list****match ip address**

**match ip next-hop**  
**match ip prefix-list**  
**match metric**  
**match tag**  
**on-match**  
**set as-path**  
**set atomic-aggregate**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.12 set as-path

To configure a route-map setting regulation and add AS before the **as-path** attribute of the BGP route, run **set as-path <prepend> <as>**. To delete the configuration you have just done, run **no set as-path <prepend> <as>**.

**set as-path <prepend> <as>**

**no set as-path <prepend> <as>**

#### Parameter

Parameter	Description
<b>prepend</b>	Means that AS is added before the <b>as-path</b> attribute.
<b>as</b>	Number of the autonomous system

#### Default

None

## Command mode

Route-map configuration mode

## Explanation

This command is only suitable to the BGP route.

## Example

In the following example, the length of the **as-path** attribute is added by adding the autonomous system number before the **as-path** attribute for each route and the result of routing choice is herein changed.

```
route-map add-as
set as-path prepend 200 200 200 200
```

## Related command

- route-map**
- match as-path**
- match community-list**
- match ip address**
- match ip next-hop**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set atomic-aggregate**
- set community**
- set community-additive**
- set ip next-hop**
- set local-preference**
- set metric**
- set origin**

**set tag**

**set weight**

**show route-map**

#### 5.4.13 set atomic-aggregate

To configure a route-map setting regulation and set the BGP route to **aggregator**, run **set atomic-aggregate**. To delete the configuration you have just done, run **no set atomic-aggregate**.

**set atomic-aggregate**

**no set atomic-aggregate**

#### Parameter

None

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

This command is only suitable to the BGP route. If the aggregation of information loss is generated when a system transmits the route, you need set the route to **atomic-aggregate**.

#### Example

In the following example, the length of the **as-path** attribute is added by adding the autonomous system number before the **as-path** attribute for each route and the result of routing choice is herein changed.

```
route-map tee
set atomic-aggregate
```

#### Related command

**route-map**

**match as-path**

**match community-list**  
**match ip address**  
**match ip next-hop**  
**match ip prefix-list**  
**match metric**  
**match tag**  
**on-match**  
**set aggregator**  
**set as-path**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.14 set community

To configure a route-map setting regulation and set the BGP route to **community**, run **set community <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**. To delete the configuration you have just done, run **no set community <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**.

**set community <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**

**no set community <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**

#### Parameter

Parameter	Description
<i>aa:nn</i>	Format of the community value
1-4294967295	Value range of the <b>community</b> parameter

<b>no-advertise</b>	Means that any neighbor will not be reported.
<b>local-AS</b>	Means that the EBGp neighbor outside of the local autonomous system or in the same autonomous system ally will not be reported.
<b>no-export</b>	Means that the neighbors in the local autonomous system or the autonomous system ally will not be reported.

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is only suitable to the BGP route. The newly-set community attribute will replace the previous community attribute of the route.

**Example**

In the following example, all routes from neighbor 193.12.202.12 will be set to **local-AS community**, enabling these routes not to be reported to other autonomous systems.

```
router bgp 200
neighbor 193.12.202.12 remote 100
neighbor 193.12.202.12 route-map tee in
route-map tee
set community local-AS
```

**Related command**

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**



**on-match**  
**set aggregator**  
**set as-path**  
**set atomic-aggregate**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.15 set community-additive

To configure a route-map setting regulation and add a value to the community attribute of the BGP route, run **set community-additive <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**. To delete the configuration you have just done, run **no set community-additive <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**.

**set community-additive <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**

**no set community-additive <aa:nn / 1-4294967295 / local-AS / no-advertise / no-export>**

#### Parameter

Parameter	Description
<i>aa:nn</i>	Format of the community value
1-4294967295	Value of the <b>community</b> parameter
<b>no-advertise</b>	Means that any neighbor will not be reported.
<b>local-AS</b>	Means that the EBGp neighbor outside of the local autonomous system or in the same autonomous system ally will not be reported.
<b>no-export</b>	Means that the neighbors in the local autonomous system or the autonomous system ally will not be reported.

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is only suitable to the BGP route. The newly-set community attribute will be added to the previous community attribute of the route.

**Example**

In the following example, all routes from neighbor 193.12.202.12 will be set to **local-AS community**, enabling these routes not to be reported to other autonomous systems.

```
router bgp 200
neighbor 193.12.202.12 remote 100
neighbor 193.12.202.12 route-map tee in
route-map tee
set community-additive local-AS
```

**Related command**

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.16 set dampening

To set the fluctuation control parameter of the BGP route and not to modify the attributes of the route, run **set dampening** [*half-time*|*reuse-value*|*suppress-value*|*hold-time*]. To delete the configuration you have just done, run **no set dampening** [*half-time*|*reuse-value*|*suppress-value*|*hold-time*].

**set dampening** [*half-time*|*reuse-value*|*suppress-value*|*hold-time*]

**no set dampening** [*half-time*|*reuse-value*|*suppress-value*|*hold-time*]

#### Parameter

Parameter	Description
<i>half-time</i>	Means the half punishment time of route attenuation.
<i>reuse-value</i>	Means the punishment value for BGP to reuse wave-limited routes.
<i>suppress-value</i>	Punishment value for BGP to limit the wave route
<i>hold-time</i>	Maximum hold time for the wave limitation of BGP route (unit: minute)

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

It is used to provide parameters for the control of BGP fluctuation route.

## Example

None

## Related command

**route-map**

### 5.4.17 set default

To set the default information for the strategy route, run **set default interface <interface-name>**. To cancel the configuration, run **no set default interface <interface-name>**.

**set default interface <interface-name>**

**no set default interface <interface-name>**

## Parameter

Parameter	Description
<i>interface-name</i>	Name of the designated interface

## Default

None

## Command mode

Route-map configuration mode

## Explanation

This command is suitable to the strategy route. The default outgoing interface of the strategy route is configured through the command. Only when the interface is in the **use** state can this command validate. The interface must meet two conditions before it is used.

第一： The UP protocol is running on the interface.

第二： The interface has the IP address or the negotiation IP address, or the interface is the NULL interface.

## Related command

**route-map**

### 5.4.18 set interface

To set the outgoing interface for the strategy route, run **set interface <interface-name>**. To cancel the configuration, run **no set interface <interface-name>**.

**set interface** <interface-name>

**no set interface** <interface-name>

#### Parameter

Parameter	Description
<i>interface-name</i>	Name of the designated interface

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

This command is suitable to the strategy route. The default outgoing interface of the strategy route is configured through the command. Only when the interface is in the **use** state can this command validate. The interface must satisfy two conditions before it is used.

Firstly: The UP protocol is running on the interface.

Secondly: The interface has the IP address or the negotiation IP address, or the interface is the NULL interface.

#### Related command

**route-map**

### 5.4.19 set ip default

To set the default next hop for the strategy route, run **set ip default nexthop <A.B.C.D>**. To cancel the configuration, run **no set ip default nexthop <A.B.C.D>**.

**set ip default nexthop** <A.B.C.D>

**no set ip default nexthop** <A.B.C.D>

**Parameter**

Parameter	Description
<i>A.B.C.D</i>	Gateway's address

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is suitable to the strategy route. Only when the next hop arrives can this command be valid.

**Example**

None

**Related command**

**route-map**

**5.4.20 set ip precedence**

To set the precedence for the strategy route, run **set ip precedence <0-7>**. To cancel the configuration, run **no set ip precedence <0-7>**.

**set ip precedence <0-7>**

**no set ip precedence <0-7>**

**Parameter**

Parameter	Description
0-7	Precedence which is set for the packet

**Default**

None

## Command mode

Route-map configuration mode

## Explanation

This command is suitable to the strategy route. When the suitable route is found by the strategy route for routing, the precedence can also be set. If the strategy route fails, the precedence cannot be set. The precedence of the IP packet is defined as follows:

routine	0
priority	1
immediate	2
flash	3
flash-override	4
critical	5
internet	6
network	7

## Related command

**route-map**

## 5.4.21 set ip tos

To set the precedence for the strategy route, run **set ip tos <0-15>**. To cancel the configuration, run **no set ip tos <0-15>**.

**set ip tos <0-15>****no set ip tos <0-15>**

## Parameter

Parameter	Description
0-15	TOS which is set for the packet

## Default

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is suitable to the strategy route. When the suitable route is found by the strategy route for routing, TOS can also be set. If the strategy route fails, the TOS cannot be set. Different TOS' can be set according to their order or can be set together:

normal	0
min-monetary	1
max-reliability	2
max-throughput	4
min-delay	8

**Related command****route-map****5.4.22 set ip next-hop**

To configure a route-map setting regulation and set the next-hop address of the route, run **set ip next-hop <a.b.c.d>**. To delete the configuration you have just done, run **no set ip next-hop <a.b.c.d>**.

**set ip next-hop <a.b.c.d>****no set ip next-hop <a.b.c.d>****Parameter**

Parameter	Description
<i>a.b.c.d</i>	IP address

**Default**

None

**Command mode**

Route-map configuration mode



## Explanation

This command is suitable to all IP routes.

## Example

In the following example, the next-hop addresses of all routes from neighbor 193.12.202.12 are set to 193.12.202.1:

```
router bgp 200
neighbor 193.12.202.12 remote 100
neighbor 193.12.202.12 route-map tee in
route-map tee
set ip next-hop 193.12.202.1
```

## Related command

- route-map**
- match as-path**
- match community-list**
- match ip address**
- match ip next-hop**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set as-path**
- set atomic-aggregate**
- set community**
- set community-additive**
- set local-preference**
- set metric**
- set origin**
- set tag**
- set weight**

**show route-map****5.4.23 set local-preference**

To configure a route-map setting regulation and set the local preference of the BGP route, run **set local-preference <value>**. To delete the configuration you have just done, run **no set local-preference <value>**.

**set local-preference <value>**

**no set local-preference <value>**

**Parameter**

Parameter	Description
<i>value</i>	Value of the local preference

**Default**

None

**Command mode**

Route-map configuration mode

**Explanation**

This command is only suitable to the BGP route.

**Example**

The following example shows that the route map can set **local-preference** to 200:

```
route-map set-local-pref
set local-preference 200
```

**Related command**

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**  
**match tag**  
**on-match**  
**set aggregator**  
**set as-path**  
**set atomic-aggregate**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set metric**  
**set origin**  
**set tag**  
**set weight**  
**show route-map**

#### 5.4.24 set metric

To configure a route-map setting regulation and set the metric of the route, run **set metric <value>**. To delete the configuration you have just done, run **no set metric <value>**.

**set metric <value>**

**no set metric <value>**

#### Parameter

Parameter	Description
<i>value</i>	Value of the metric

#### Default

None

#### Command mode

Route-map configuration mode

## Explanation

This command is suitable to all IP routes.

## Example

The following example shows that the route map can set **metric** to 120:

```
route-map set-metric  
set metric 120
```

## Related command

- route-map**
- match as-path**
- match community-list**
- match ip address**
- match ip next-hop**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set as-path**
- set atomic-aggregate**
- set community**
- set community-additive**
- set ip next-hop**
- set local-preference**
- set origin**
- set tag**
- set weight**
- show route-map**

### 5.4.25 set metric-type

To set the value of the **metric-type** parameter for supporting the **external type** OSPF route, run **set metric-type [type-1 | type2]**. To delete the configuration you have just done, run **no set metric-type [type-1 | type2]**.

**set metric-type [type-1 | type2]**

**no set metric-type [type-1 | type2]**

#### Parameter

Parameter	Description
<i>Type-1</i>	External type-1 of OSPF metric
<i>Type-2</i>	External type-2 of OSPF metric

#### Default

None

#### Command mode

Route-map configuration mode

#### Explanation

This command is only suitable to external OSPF routes.

#### Example

The following example shows that the route map can set **metric-type** to **type1**:

```
route-map set-metric-type
set metric-type type1
```

#### Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

**show route-map**

#### 5.4.26 set origin

To set the **origin** attribute of the BGP route, run **set origin [igp | egp | incomplete]**. To delete the configuration you have just done, run **no set origin [igp | egp | incomplete]**.

**set origin [igp | egp | incomplete]**

**no set origin [igp | egp | incomplete]**

#### Parameter

Parameter	Description
<b>igp</b>	Internal route of the autonomous system
<b>egp</b>	External route of the autonomous system
<b>incomplete</b>	Uncertain route

#### Default

**igp** is the default route locally configured through the **network** command, **Incomplete** is the default route locally configured through the **aggregate** command or the **redistribute** command.

## Command mode

Route-map configuration mode

## Explanation

This command is only suitable to the BGP route.

## Example

The following example shows how the defined route map sets the BGP route with a 10-starting **original** attribute to **igp**.

```
ip as-path-list self permit ^10
route-map set-origin
match as-path self
set origin igp
```

## Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set tag**

**set weight**

**show route-map**

#### 5.4.27 set tag

To set the tag of the route, run **set tag <value>**. To delete the configuration you have just done, run **no set tag <value>**.

**set tag <value>**

**no set tag <value>**

#### Parameter

Parameter	Description
<i>value</i>	Value of the tag

#### Default

The default tag value is 0.

#### Command mode

Route-map configuration mode

#### Explanation

This command is suitable to all IP routes.

#### Example

The following example shows how to set **tag** to 120980 through the route map:

```
route-map set-tag
set tag 120980
```

#### Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**



**match ip next-hop**  
**match ip prefix-list**  
**match metric**  
**match tag**  
**on-match**  
**set aggregator**  
**set as-path**  
**set atomic-aggregate**  
**set community**  
**set community-additive**  
**set ip next-hop**  
**set local-preference**  
**set metric**  
**set origin**  
**set weight**  
**show route-map**

#### 5.4.28 set weight

To set the weight of the BGP route, run **set weight <value>**. To delete the configuration you have just done, run **no set weight <value>**.

**set weight <value>**

**no set weight <value>**

#### Parameter

Parameter	Description
<i>value</i>	Value of the weight

#### Default

The default weight value of the locally-generated BGP route is 32768 and the weight value obtained from the neighbor is 0.

**Command mode**

Route-map configuration mode

**Explanation**

This command is only suitable to the BGP route.

**Example**

The following example shows how to set the weight to 230 through the route map:

```
route-map set-weight  
set weight 230
```

**Related command**

- route-map**
- match as-path**
- match community-list**
- match ip address**
- match ip next-hop**
- match ip prefix-list**
- match metric**
- match tag**
- on-match**
- set aggregator**
- set as-path**
- set atomic-aggregate**
- set community**
- set community-additive**
- set ip next-hop**
- set local-preference**
- set metric**
- set origin**

**set tag**

**show route-map**

#### 5.4.29 show route-map

To display the information about the route map, run the following command:

**show route-map** [*name*]

#### Parameter

Parameter	Description
<i>name</i>	Name of the route map

#### Default

None

#### Command mode

EXEC

#### Explanation

If the name of the route map is not designated, all configured route maps in the system will be displayed.

#### Example

The following example shows that all route maps in the system are displayed:

Show ip route-map

#### Related command

**route-map**

**match as-path**

**match community-list**

**match ip address**

**match ip next-hop**

**match ip prefix-list**

**match metric**

**match tag**

**on-match**

**set aggregator**

**set as-path**

**set atomic-aggregate**

**set community**

**set community-additive**

**set ip next-hop**

**set local-preference**

**set metric**

**set origin**

**set tag**

**set weight**

## Chapter 6 RSVP Configuration Command

### 6.1.1 debug ip rsvp local

To display the transmission and reception of RSVP signaling information, run **debug ip rsvp local**. To forbid the output of the RSVP signaling information, run **[no] debug ip rsvp local [call | upcall] [detail]**.

**[no] debug ip rsvp local [call | upcall] [detail]**

#### Parameter

Parameter	Description
<b>call   upcall</b>	Displays the local task or the RSVP request which is transmitted or received by user.
<b>detail</b>	Displays the detailed information.

#### Default

None

#### Command mode

EXEC

#### Explanation

This command can track the interaction information about the local RSVP. If the **call** parameter is in the command sentence, the local task or the user-transmitted RSVP request will be displayed. If the **upcall** parameter is in the command sentence, the local task or the user-received RSVP request will be displayed. If both parameters are not in the command sentence, all information will be displayed. If the **detail** parameter is in the command sentence, the detailed information about RSVP interaction will be displayed.

#### Example

The following information is displayed after the **debug ip rsvp local call** command is run:

```
ROUTER# debug ip rsvp local call
```

```
RSVP:RSVP trace call on
```

```
RSVP: <Session 1> session from api // 本地 RSVP api 建立 session
```

If you run **debug ip rsvp locall call** to track the state of the local RSVP session, the session ID and actions will be displayed.

The following information is displayed after the **debug ip rsvp call detail** command is run:

```
ROUTER# debug ip rsvp call detail
RSVP:RSVP trace call detail on
RSVP: session from api // the local RSVP api creates the session

Session ID: 2

Session Addr:33.33.33.33
Session Port:4554
Session Pid :17
```

If you run **debug ip rsvp locall call** to track the state of the local RSVP session, the session ID and actions will be displayed. Also the detailed content in the actions will be explained. The explanation is in the format similar to the RAPI format in the RSVP protocol.

The following information is displayed after the **debug ip rsvp local upcall** command is run:

```
ROUTER# debug ip rsvp upcall
RSVP:RSVP trace upcall on
RSVP: <Session 1> confirm upcall
```

The **debug ip rsvp local upcall** command is similar to the **debug ip rsvp local call** command.

#### Related command

**debug ip rsvp packet**

#### 6.1.2 debug ip rsvp packet

To display the transmission and reception of RSVP signaling information on the interface of the router, run **debug ip rsvp packet**. To forbid the output of the RSVP signaling information, run **[no] debug ip rsvp local [call | upcall] [detail]**.

**[no] debug ip rsvp packet [detail]**

#### Parameter

Parameter	Description
<b>detail</b>	Displays the detailed information.

#### Default

None

## Command mode

EXEC

## Explanation

This command can trace the information about the transmission and reception of the RSVP packet. If the **detail** parameter is in the command sentence, the detailed information about RSVP packet will be displayed.

## Example

The following information is displayed after the **debug ip rsvp packet** command is run on interface f0/0:

```
ROUTER# debug ip rsvp packet
RSVP:RSVP trace on
RSVP: Receive RSVP PATH packet for 192.168.20.44 from local application
RSVP: Send RSVP PATH packet for 192.168.20.44 to 192.168.20.44
```

The **debug ip rsvp packet** command can be used to trace the RSVP packet, displaying the type, source address and destination address of the packet.

The following information is displayed after the **debug ip rsvp packet detail** command is run:

```
ROUTER# debug ip rsvp packet detail
RSVP:RSVP trace detail on
RSVP: Send RSVP PATH packet for 192.168.20.44 to 192.168.20.44 //transmitting RSVP PATH
command header: version:1 flags:0000 type:PATH cksum:5073 ttl:128 reserved:0 length:180
// RSVP header
SESSION      type 1  length 12: C0A8142C : 0601014D
RSVP_HOP      type 1  length 12: C0A81463 : 00000001
TIME_VALUES   type 1  length 8: 00007530
SENDER_TEMPLATE type 1  length 12: C0A81463 : 0000014D
SENDER_TSPEC   type 2  length 36
version: 0  length: 7
service header id:1  length:6
parameter header id:127 flags:0  length:5
    average rate(Bps)      :125
    burst depth(byte)      :1000
    peak rate(Bps)         :125
    min unit(byte)         :0
    max unit(byte)         :0
ADSPEC        type 2  length 92
version: 0  length: 21
    general parameters break bit:0  length:8
    IS hop cnt              :1
    minimum path bandwidth(Bps) :10000000
    minimum path latency(byte) :0
```

```

composed MTU(byte)           :1500
guaranteed service break bit:0 length:8
path delay(ms)               :192000
path jitter(ms)              :12000
path delay since shaping(ms) :192000
path jitter since shaping(ms) :12000
Control Load service break bit:0 length:2
minimum path bandwidth(Bps)   :1000

```

The **deb ip rsvp packet detail** command is used to trace the RSVP packets and explain the content of the packet according to the RSVP protocol.

Related command

**debug ip rsvp local**

### 6.1.3 ip rsvp bandwidth

To start the RSVP protocol on an interface, run **ip rsvp bandwidth**. To forbid the RSVP protocol running on the interface, run **no ip rsvp bandwidth**.

**ip rsvp bandwidth** [*interface-kbps* | *single-flow-kbps*]

**no ip rsvp bandwidth** [*interface-kbps* | *single-flow-kbps*]

#### Parameter

Parameter	Description
<i>interface-kbps</i>	Cap of the reserved resource that the whole interface can apply for
<i>single-flow-kbps</i>	Cap of the reserved resource that a single data flow can apply for

#### Default

RSVP is forbidden on the interface by default.

#### Command mode

Interface configuration mode

#### Explanation

The command can enable an interface to have the RSVP function. The **single-flow-kbps** parameter can be used to set the cap of reserved source which a single data flow can apply for. If the command sentence has no parameter, the cap is 75% of the total resource of the interface by default.



## Example

The following example shows that the total reserved RSVP bandwidth on interface f0/0 is 1M and the bandwidth of a single data flow is 200K.

```
interface f0/0
ip rsvp bandwidth 1000 200
```

## Related command

**ip rsvp neighbor**

### 6.1.4 ip rsvp local reservation

To enable users to interact with other hosts through RSVP, the capability to send the RSVP RESV message to the peer must be owned. The **ip rsvp local reservation** command can meet the requirement. Before this command is invoked to send the RSVP RESV message, you must run **ip rsvp local session** to create the RSVP session. To cancel the configuration, run **no ip rsvp local reservation session-id**.

**ip rsvp local reservation session-id sender-ip-address sender-sport [guarantee | load] [bandwidth] [burst-size]**

**no ip rsvp local reservation session-id**

## Parameter

Parameter	Description
<b>session-id</b>	ID of the session
<i>sender-ip-address</i>	Address of the sender in the RSVP flow
<i>sender-sport</i>	Interface of the sender in the RSVP flow
<b>guarantee   load</b>	Type of rsvp reservation
<i>bandwidth</i>	Average rate of the reserved resource
<i>burst-size</i>	Size of the maximum burst data

## Default

The user has not set the command.

## Command mode

Global configuration mode

## Explanation

The command can enable the user to send the RESV information to the outside. The “no” form of the command can send the RESV TEAR message. By default, the reservation type is **Control Load**. The average rate of the reserved resource and the input size of the maximum burst data is 1K.

## Example

The following example shows how to use the command:

```
ip rsvp local reservation 1 1.0.0.2 3000 load 100 60
ip rsvp local reservation 2 2.0.3.2 4000 guarantee 150 65
```

## Related command

**ip rsvp local session**

**ip rsvp local sender**

### 6.1.5 ip rsvp local sender

To enable users to interact with other hosts through RSVP, the capability to send the RSVP PATH message to the peer must be owned. The **ip rsvp local sender** command can meet the requirement. Before this command is invoked to send the RSVP PATH message, you must run **ip rsvp local session** to create the RSVP session. To cancel the configuration, run **no ip rsvp local sender session-id**.

```
ip rsvp local sender session-id sender-ip-address sender-sport
[bandwidth]burst-size
```

```
no ip rsvp local sender session-id
```

## Parameter

Parameter	Description
<b>session-id</b>	ID of the session
<i>sender-ip-address</i>	Address of the sender in the RSVP flow
<i>sender-sport</i>	Interface of the sender in the RSVP flow
<i>bandwidth</i>	Average rate of the reserved resource
<i>burst-size</i>	Size of the maximum burst data

## Default

The user has not set the command.

**Command mode**

Global configuration mode

**Explanation**

The command can enable the user to send the PATH information to the outside. The “no” form of the command can send the PATH TEAR message. By default, the average rate of the reserved resource and the input size of the maximum burst data is 1K.

**Example**

The following example shows how to use the command:

```
ip rsvp local sender 1 1.0.0.2 3000 100 60
ip rsvp local sender 2 2.0.3.2 4000 150 65
```

**Related command**

**ip rsvp local session**

**ip rsvp local reservation**

**6.1.6 ip rsvp local session**

To create the RSVP session, run **ip rsvp local session**. To cancel the configuration, run **no ip rsvp local session**.

**ip rsvp local session** *session-ip-address session-dport {tcp | udp}*

**no ip rsvp local session** *session-id*

**Parameter**

Parameter	Description
<b>session-id</b>	ID of the session
<i>session-ip-address</i>	Address of the destination host
<i>session-dport</i>	Port ID of the destination host
<b>tcp   udp</b>	Protocol number of the to-be-reserved data flow

**Default**

The user has not set the command.

**Command mode**

Global configuration mode

### Explanation

You can run this command to create a new RSVP session, which can be used by other commands.

### Example

The following example shows how to use the command:

```
ip rsvp local session 1.0.0.3 3000 UDP
ip rsvp local session 3.4.4.3 5600 TCP
```

### Related command

**ip rsvp local sender**

**ip rsvp local reservation**

## 6.1.7 ip rsvp neighbor

To accept RSVP requests from other hosts on an interface, run **ip rsvp neighbor *access-list-name***. To cancel the configuration, run **no ip rsvp neighbor *access-list-name***.

**ip rsvp neighbor *access-list-name***

**no ip rsvp neighbor *access-list-name***

### Parameter

Parameter	Description
<i>access-list-name</i>	Name of the IP access control list

### Default

All RSVP packets are accepted on the interface.

### Command mode

Interface configuration mode

### Explanation

Requests from some RSVP hosts are accepted, while requests from other hosts are declined. The command is used to control the running state of the RSVP. After the command is configured, the RSVP requests of the hosts which comply with the access control list are received, or the RSVP requests will be declined.

## Example

The following example shows that the RSVP requests of the RSVP host which satisfy the requirements of access list ABC are allowed on interface f0/0.

```
interface f0/0
ip rsvp neighbor ABC
```

## Related command

**ip rsvp bandwidth**

### 6.1.8 ip rsvp precedence

To configure the precedence of the data flow, run **ip rsvp precedence {conform|exceed} precedence-value**.

**ip rsvp precedence {conform|exceed} precedence-value**

**no ip rsvp precedence {conform|exceed}**

## Parameter

Parameter	Description
<b>conform exceed</b>	The <b>conform</b> parameter is used to set the TOS when the data traffic volume is smaller than the reserved value. The <b>exceed</b> parameter is used to set the TOS when the data traffic volume exceeds the reserved value.
<i>precedence-value</i>	Value of the precedence

## Default

The user has not set the command.

## Command mode

Interface configuration mode

## Explanation

This command is used to set the precedence of the reserved flow.

## Example

The following example shows how to use the command:

```
ip rsvp precedence conform 6
ip rsvp precedence exceed 5
```

Related command

**ip rsvp tos**

### 6.1.9 ip rsvp tos

To configure the TOS of the data flow, run **ip rsvp tos {conform|exceed} *tos-value***.

**ip rsvp tos {conform|exceed} *tos-value***

**no ip rsvp tos {conform|exceed}**

#### Parameter

Parameter	Description
<b>conform exceed</b>	The <b>conform</b> parameter is used to set the TOS when the data traffic volume is smaller than the reserved value. The <b>exceed</b> parameter is used to set the TOS when the data traffic volume exceeds the reserved value.
<i>tos-value</i>	Value of the TOS

#### Default

The user has not set the command.

#### Command mode

Interface configuration mode

#### Explanation

This command is used to set the TOS of the reserved flow.

#### Example

The following example shows how to use this command:

```
ip rsvp tos conform 6
ip rsvp tos exceed 5
```

Related command

**ip rsvp precedence**

### 6.1.10 show ip rsvp installed

To display the reserved information on the RSVP interface, run the following command:

**show ip rsvp installed** [*type-number*]**Parameter**

Parameter	Description
<i>type-number</i>	Interface ID of the router

**Default**

None

**Command mode**

EXEC

**Explanation**

This command is used to display the detailed information about the reserved RSVP flow on the interface of the router. If the command sentence has no relative parameters, the detailed information about all reserved flows on the RSVP-functioned interfaces will be displayed.

**Example**

The following information is displayed after the **show ip rsvp installed** command is run on interface f0/0:

```
ROUTER# show ip rsvp installed f0/0
```

```
f0/0 :
```

allocate	SessAddr	SessPort	SrcAddr	SrcPort	ProtId
20K	12.3.3.45	1000	30.2.3.2	2000	TCP

The following information is displayed after the **show ip rsvp interface** command is run:

```
ROUTER# show ip rsvp installed
```

```
f0/0 :
```

allocate	SessAddr	SessPort	SrcAddr	SrcPort	ProtId
20K	12.3.3.45	1000	30.2.3.2	2000	TCP

```
api :
```

allocate	SessAddr	SessPort	SrcAddr	SrcPort	ProtId
10K	12.43.3.45	1030	40.2.3.2	2040	UDP

**Related command****show ip rsvp interface****show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp local**

**show ip rsvp tos**

**show ip rsvp precedence**

### 6.1.11 **show ip rsvp interface**

To display the reserved information on the RSVP interface, run the following command:

**show ip rsvp interface [*type-number*]**

#### **Parameter**

Parameter	Description
type-number	Interface ID of the router

#### **Default**

RSVP is forbidden on the interface by default.

#### **Command mode**

EXEC

#### **Explanation**

This command is used to display the information about the reserved RSVP flow on the interface of the router.

If the command sentence has no relative parameters, the detailed information about all RSVP-functioned interfaces will be displayed.

#### **Example**

The following information is displayed after the **show ip rsvp interface** command is run on interface f0/0:

```
ROUTER# show ip rsvp interface f0/0
interface  allocate  i/f max  flow max
f0/0      30K      7500K   7500K
```

The following information is displayed after the **show ip rsvp interface** command is run:

```
ROUTER# show ip rsvp interface
interface  allocate  i/f max  flow max
```



f0/0	30K	7500K	7500K
api	20K	-	-

Related command

**show ip rsvp installed**

**show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp local**

**show ip rsvp tos**

**show ip rsvp precedence**

#### 6.1.12 **show ip rsvp local**

To display the information about reservation in the router's database, run **show ip rsvp local**.

**show ip rsvp local [session-id]**

##### Parameter

Parameter	Description
<b>session-id</b>	ID of the local RSVP session

##### Default

None

##### Command mode

EXEC

##### Explanation

You can run **show ip rsvp local** to display the information about the local session in the router's database. If there is no other parameter, the information about all local sessions of the router will be displayed.

##### Example

The following information is displayed after the **show ip rsvp local 20** command is run:

```
ROUTER# show ip rsvp local 20
```

Sid	SessAddr	SrcAddr	Pro	DPort	Sport	Type	BPS	Bytes	User
20	23.44.33.44	33.33.44.33	TCP	2200	3333	GU	2	2	SYS

The following information is displayed after the **show ip rsvp local** command is run:

```
ROUTER# show ip rsvp local
```

Sid	SessAddr	SrcAddr	Pro	DPort	Sport	Type	BPS	Bytes	User
20	23.44.33.44	33.33.44.33	TCP	2200	3333	GU	2	2	SYS
42	24.54.36.64	34.63.77.53	UDP	2500	3773	LD	5	5	USR

#### Related command

**show ip rsvp interface**

**show ip rsvp installed**

**show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp tos**

**show ip rsvp precedence**

### 6.1.13 show ip rsvp neighbor

To display the reservation information in the router's database, run **show ip rsvp neighbor**.

**show ip rsvp neighbor** [*type-number*]

#### Parameter

Parameter	Description
<i>type-number</i>	Interface ID of the router

#### Default

None

#### Command mode

EXEC

## Explanation

This command is used to display the information about the address of the RSVP host or RSVP router in the router's database adjacent to an interface. If there is no other parameter, the information about all interfaces will be displayed.

## Example

The following information is displayed after the **show ip rsvp neighbor** command is run on interface f0/0:

```
ROUTER# show ip rsvp neighbor f0/0
```

```
f0/0 :
```

Neighbor	Encapsulation
192.168.20.43	RAW

The following information is displayed after the **show ip rsvp neighbor** command is run:

```
ROUTER# show ip rsvp neighbor
```

```
f0/0 :
```

Neighbor	Encapsulation
192.168.20.43	RAW

```
  a0 :
```

Neighbor	Encapsulation
193.148.20.43	UDP

## Related command

**show ip rsvp interface**

**show ip rsvp installed**

**show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp local**

**show ip rsvp tos**

**show ip rsvp precedence**

### 6.1.14 **show ip rsvp precedence**

To display the information about the precedence of the RSVP flow on the router's interface, run the following command:

```
show ip rsvp precedence [type-number]
```

**Parameter**

Parameter	Description
<i>type-number</i>	Interface number of the router

**Default**

None

**Command mode**

EXEC

**Explanation**

This command is used to display the information about the precedence of the RSVP flow on the interface of the router. If there is no other parameters in this command, all precedence settings of the RSVP flow on all interfaces of the router will be displayed.

**Example**

The following information is displayed after the **show ip rsvp precedence f0/0** command is run:

```
ROUTER# show ip rsvp precedence f0/0
```

Interface	Conform	Exceed
f0/0	4	-

The following information is displayed after the **show ip rsvp precedence** command is run:

```
ROUTER# show ip rsvp precedence
```

Interface	Conform	Exceed
f0/0	4	-
e1/1	-	4

**Related command**

**show ip rsvp interface**

**show ip rsvp installed**

**show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp local**

**show ip rsvp tos**

### 6.1.15 show ip rsvp reservation

To display the reservation information in the router's database, run the following command:

**show ip rsvp reservation** [*dest-ip-address*]

#### Parameter

Parameter	Description
<i>dest-ip-address</i>	Destination IP address of the RSVP session

#### Default

None

#### Command mode

EXEC

#### Explanation

You can run **show ip rsvp reservation** to display the reservation information in the router's database. If there is no other parameter, all reservation information on the router will be displayed.

#### Example

The following information is displayed after the **show ip rsvp reservation 192.14.3.2** command is run:

ROUTER# show ip rsvp reservation 192.14.3.2

SessAddr	DP	Pid	SrcAddr	SP	NextHop	Int	Fi	Sv	Bps	byte
192.14.3.2	1000	TCP	122.3.4.6	2000	133.3.3.4	a0	FF	GU	10K	10K
193.14.3.2	1030	TCP	124.3.3.7	2300	143.3.5.4	f0/0	FF	LD	1K	2K

The following information is displayed after the **show ip rsvp reservation** command is run:

ROUTER# show ip rsvp reservation

SessAddr	DP	Pid	SrcAddr	SP	NextHop	Int	Fi	Sv	Bps	byte
192.14.3.2	1000	TCP	122.3.4.6	2000	133.3.3.4	a0	FF	GU	10K	10K
193.14.3.2	1030	TCP	124.3.3.7	2300	143.3.5.4	f0/0	FF	LD	1K	2K

#### Related command

**show ip rsvp interface**

**show ip rsvp installed**  
**show ip rsvp sender**  
**show ip rsvp neighbor**  
**show ip rsvp local**  
**show ip rsvp tos**  
**show ip rsvp precedence**

#### 6.1.16 show ip rsvp sender

To display the information about the sender in the router's database, run the following command:

**show ip rsvp sender** [*dest-ip-address*]

#### Parameter

Parameter	Description
dest-ip-address	Destination IP address of the RSVP session

#### Default

None

#### Command mode

EXEC

#### Explanation

You can run **show ip rsvp sender** to display the information about the sender in the router's database. If there is no other parameters in this command, the information about all senders on the router will be displayed.

#### Example

The following information is displayed after the **show ip rsvp sender** command is run at destination address 192.14.3.2:

```

ROUTER# show ip rsvp sender 192.14.3.2
  SessAddr      DP    Pid  SrcAddr      SP    PrevHop      Intf  Bps  byte
  192.14.3.2    1000 TCP 122.3.4.6    2000  133.3.3.4      f0/0
10K 10K
  193.14.3.2    1030 TCP 124.3.3.7    2300  143.3.5.4      e0/0
1K 2K

```

The following information is displayed after the **show ip rsvp sender** command is run:

```
ROUTER# show ip rsvp sender
```

SessAddr	DP	Pid	SrcAddr	SP	PrevHop	Intf	Bps	byte
192.14.3.2	1000	TCP	122.3.4.6	2000	133.3.3.4			f0/0
10K 10K								
193.14.3.1	1230	TCP	124.3.3.7	2300	143.3.5.4			e0/0
1K 2K								

Related command

**show ip rsvp interface**

**show ip rsvp installed**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp local**

**show ip rsvp tos**

**show ip rsvp precedence**

### 6.1.17 show ip rsvp tos

To display the information about TOS settings of the RSVP flow on the router's interface, run the following command:

**show ip rsvp tos** [*type-number*]

#### Parameter

Parameter	Description
<i>type-number</i>	Interface ID of the router

#### Default

None

#### Command mode

EXEC

#### Explanation

This command is used to display the information about the TOS settings of the RSVP flow on the interface of the router. If there are no other parameters in this command, all TOS settings of the RSVP flow on all interfaces of the router will be displayed.

## Example

The following information is displayed after the **show ip rsvp tos** command is run on interface f0/0:

```
ROUTER# show ip rsvp tos f0/0
```

Interface	Conform	Exceed
f0/0	4	-

The following information is displayed after the **show ip rsvp tos** command is run:

```
ROUTER# show ip rsvp tos
```

Interface	Conform	Exceed
f0/0	4	-
e1/1	-	3

## Related command

**show ip rsvp interface**

**show ip rsvp installed**

**show ip rsvp sender**

**show ip rsvp reservation**

**show ip rsvp neighbor**

**show ip rsvp local**

**show ip rsvp precedence**



## Chapter 7 PBR Configuration Commands

### 7.1 PBR Configuration Commands

HTTP configuration commands include:

debug ip policy

ip policy route-map

match ip address

match length

set default interface

set interface

set ip default next-hop

set ip next-hop

route-map

Debug ip policy

ip local policy

ip policy

ip route-weight

show ip local policy

show ip policy

#### 7.1.1 debug ip policy

To check the results of applying the policy route, run **debug ip policy**.

**debug ip policy**

**no debug ip policy**

#### Parameter

None

**Default**

By default, the results of policy route application will not be printed.

**Command mode**

EXEC

**Explanation**

This command can be used to check whether the IP packets received from the interface have been applied with the policy route.

Because the results of policy routing application for each interface-received IP packet will be printed after this command is run, please use this command when the network traffic is low.

**Example**

The following example shows after the **debug ip policy** command is run:

```
Router# debug ip policy
2004-1-16 15:32:54 PBR: s=10.1.1.2 (FastEthernet0/0), d=99.1.1.1, len 84, policy rejected --
normal forwarding
2004-1-16 15:32:54 PBR: s=10.1.1.21 (FastEthernet0/0), d=99.1.1.1 (FastEthernet0/0.13), len=
84, gate=13.1.1.99 policy routed
```

**Related command**

None

**7.1.2 ip policy route-map**

To apply the policy route to the interface-received IP packet, run **ip policy route-map route-map name** in interface configuration mode. To cancel the policy route on the interface, run **no ip policy route-map route-map name**.

**ip policy route-map route-map name**

**no ip policy route-map route-map name**

**Parameter**

Parameter	Description
<i>route-map name</i>	Name of the route map

**Default**

None

**Command mode**

Interface configuration mode

**Explanation**

If you want to apply the policy route to the interface-received IP packet, you need to run the **ip policy route-map** command.

**Example**

The following example shows how to enable the policy route on interface f0/0.

```
Router_config#int f0/0
```

```
Router_config_f0/0#ip policy route-map pbr
```

**Related command**

**route-map**

**7.1.3 match ip address**

To apply the matchip policy based on source IP address, run **match ip address access-list name**.

**match ip address** *access-list name*

**no match ip address** [*access-list name*]

**Parameter**

Parameter	Description
<i>access-list name</i>	Name of the standard IP access control list

**Default**

The access list is not designated by default.

**Command mode**

Route-map configuration mode

**Explanation**

If the route map is applied to the policy route, the source address of the IP packet will be used to match the configured access list. If the source address does match the access list, the **set** regulation is then applied; otherwise, the next sequence number of the same route map will be used.

## Example

The following example shows that the packets whose source IP addresses are allowed by access list **net1** will be transmitted to interface s0/0:

```
interface f0/0
ip policy route-map moon
!
route-map moon
match ip address net1
set interface s0/0
```

## Related command

**set default interface**

**set interface**

**set ip default next-hop**

**set ip next-hop**

**route-map**

### 7.1.4 match length

To set the route policy according to the length of the IP packet, run **match length**.

**match length** *minimum-length maximum-length*

**no match length** *minimum-length maximum-length*

## Parameter

Parameter	Description
<i>minimum-length</i>	Designates the minimum length of the matched packet.
<i>maximum-length</i>	Designates the maximum length of the matched packet.

## Default

It is not configured by default.

## Command mode

Route-map configuration mode

## Explanation

This command is used to conduct the policy routing according to the size of the IP packet.

## Example

The following example shows that the IP packet whose size ranges between 1000 bytes to 1500 bytes will be transmitted to interface s0/0.

```
interface f0/0
ip policy route-map moon
!
route-map moon
match length 1000 1500
set interface s0/0
```

## Related command

**match ip address**

**set default interface**

**set interface**

**set ip default next-hop**

**set ip next-hop**

**route-map**

### 7.1.5 set default interface

To set the default next-hop interface for the matched IP packet, run **set default interface**.

**set default interface *interface name* [...*interface name*] [/load-balance]**

**no set default interface *interface name* [...*interface name*] [/load-balance]**

## Parameter

Parameter	Description
<i>interface name</i>	Name of the interface

## Default

It is not configured by default.

## Command mode

Route-map configuration mode

## Explanation

Before you set the default next-hop interface for the matched IP packet through the **set default interface** command, the following conditions must be satisfied:

The **set ip next-hop** command is not configured, or the **set ip next-hop** command is configured but the route of the next hop designated by **set ip next-hop** is not in the routing table.

If the **set interface** command is not configured or the **set interface** command is configured but these interfaces cannot be routed (the interface is down or there is no IP address).

The **set ip default next-hop** command or the **set ip default next-hop** command is not configured but the route of the next hop designated by **set ip default next-hop** is not in the routing table.

## Example

None

## Related command

**match ip address**

**match length**

**set interface**

**set ip default next-hop**

**set ip next-hop**

**route-map**

### 7.1.6 set interface

To set the next-hop interface for the matched IP packet, run **set interface**.

**set interface *interface name* [...*interface name*] [*load-balance*]**

**no set interface *interface name* [...*interface name*] [*load-balance*]**

**Parameter**

Parameter	Description	Default
<i>interface name</i>	Name of the interface	

It is not configured by default.

**Command mode**

Route-map configuration mode

**Explanation**

Before you set the next-hop interface for the matched IP packet through the `set interface` command, the following conditions must be satisfied:

The **set ip next-hop** command or the **set ip next-hop** command is not configured, and the route of the next hop designated by **set ip next-hop** is not in the routing table.

The interface is in the routing state (the protocol on the interface is up and the IP address exists).

**Example**

None

**Related command**

**match ip address**

**match length**

**set default interface**

**set ip default next-hop**

**set ip next-hop**

**route-map**

**7.1.7 set ip default next-hop**

To set the default next-hop for the matched IP packet, run **set ip default next-hop**.

**set ip default next-hop A.B.C.D [...A.B.C.D] [Load-balance]**

**no set ip default next-hop A.B.C.D [...A.B.C.D] [Load-balance]**

**Parameter**

Parameter	Description
<i>A.B.C.D</i>	Address of the next hop

**Default**

It is not configured by default.

**Command mode**

Route-map configuration mode

**Explanation**

Before you set the default next hop for the matched IP packet through the **set ip default next-hop** command, the following conditions must be satisfied.

The **set ip next-hop** command or the **set ip next-hop** command is not configured, and the route of the next hop designated by **set ip next-hop** is not in the routing table.

If the **set interface** command is not configured or the **set interface** command is configure but these interfaces cannot be routed (the interface is down or there is no IP address).

The route of the next hop designated by the **set ip default next-hop** command exists in the routing table.

**Related command**

**set default interface**

**set interface**

**set ip next-hop**

**route-map**

**7.1.8 set ip next-hop**

To set the next hop for the matched IP packet, run **set ip next-hop**.

**set ip next-hop A.B.C.D [...A.B.C.D] [Load-balance]**

**no set ip next-hop A.B.C.D [...A.B.C.D] [Load-balance]**

**Parameter**

Parameter	Description
-----------	-------------



A.B.C.D	Address of the next hop
---------	-------------------------

**Default**

It is not configured by default.

**Command mode**

Route-map configuration mode

**Explanation**

Before you set the next hop for the matched IP packet through the **set ip next-hop** command, the following conditions must be satisfied:

The route of the next hop designated by the **set ip next-hop** command exists in the routing table.

**Related command**

**set default interface**

**set interface**

**set ip default next-hop**

**set ip next-hop**

**route-map**

**7.1.9 route-map**

**route-map** *route-map name* [*sequence-number*] [**permit** | **deny**]

**no route-map** *route-map name* [*sequence-number*] [**permit** | **deny**]

**Parameter**

Parameter	Description
<i>route-map name</i>	Name of the route map
<i>sequence-number</i>	Sequence number of the designated route map, which is optional
<b>permit</b>	Means that the route or the policy route is allowed to be forwarded if the IP packet is matched. The parameter is optional.
<b>deny</b>	Means that the route or the policy route is forbidden to be forwarded if the IP packet is matched. The parameter is optional.

## Default

There is no static routes by default.

## Command mode

Global configuration mode

## Explanation

The **route-map** command is used to configure the route map.

## Example

The following example shows that route map **pbr** is configured.

```
route-map pbr 10 permit
match ip address net1
set ip next-hop 13.1.1.99
!
route-map pbr 20 permit
match ip address net2
set ip next-hop 14.1.1.99
!
route-map pbr 30 permit
match ip address net3
set ip next-hop 13.1.1.99 14.1.1.99 load-balance
```

## Related command

**match ip address**

**match length**

**set default interface**

**set interface**

**set ip default next-hop**

**set ip next-hop**

7.1.10 **debug ip policy**

**debug ip policy**

**no debug ip policy**

**Parameter**

None

**Default**

The trace function of the policy route is not enabled by default.

**Command mode**

EXEC

**Explanation**

The **debug ip policy** command is used to open the trace function of the policy route, while the **no debug ip policy** command is used to shut down the trace function of the policy route.

**Example**

None

**Related command****ip local policy****ip policy****show ip local policy****show ip policy****7.1.11 ip local policy**

To open the policy route of the local packet, run **ip local policy route-map [name]**. To shut down the policy route of the local packet, run **no ip local policy route-map [name]**.

**ip local policy route-map [name]****no ip local policy route-map [name]****Parameter**

Parameter	Description
<i>name</i>	Name of the route map used by the policy route

## Default

The policy routing function of the local packet is shut down by default.

## Command mode

Global configuration mode

## Explanation

The policy route can be applied to the locally-transmitted packets or the forwarded packets. The route applied to the locally-transmitted packets are called as the local policy route. After the **ip local policy route-map <name>** command and a proper route map are configured in global configuration mode, you can apply the policy route to the locally-transmitted packets.

The policy route checks whether the packets are the broadcast packets, and the broadcast packets also checks the corresponding policy route. Among the results of the policy route, only an outgoing interface or a next hop is shown. The route-to-multiport condition does not exist.

The route map which is used for the policy route can match the packet according to the access list or the packet's length. The policy routing is conducted by setting the next hop or the outgoing interface. Various policies can be satisfied using the access list according to the routes, such as the route of the source address and the application route.

The policy route can be used to set the outgoing interface, next hop, TOS and precedence of the packet. The order to choose the policy route is as follows: nexthop, default nexthop, interface and default interface. The normal route can be adopted when all the four types of previous policy routes are unavailable.

If **nexthop** is available, it means that a route can be found in the routing table for **nexthop**. If **interface** is available, it means that the IP protocol on the interface is up and the legal IP address exists.

## Example

The following example shows that the policy routing is conducted to the locally-transmitted packets. The packets from the network whose destination address is 100.0.0.0/8 will be transmitted to interface s0/0:

```
ip local policy route-map Policy
!
route-map Policy
match ip address Policy-ACL
set interface s1/0
!
ip access-list extended
permit ip any 100.0.0.0 255.0.0.0
!
```

## Related command

**ip policy****show ip local policy****show ip policy**

## 7.1.12 ip policy

To open the policy route on an interface, run **ip policy route-map [name]**. To shut down the local policy route, run **no ip policy route-map [name]**.

**ip local policy route-map [name]****no ip policy route-map [name]**

## Parameter

Parameter	Description
<i>name</i>	Name of the route map used by the policy route

## Default

The policy routing function on an interface is shut down by default.

## Command mode

Port configuration mode

## Explanation

The policy route can be applied to the locally-transmitted packets or the forwarded packets

The policy route is to check whether the packet is the broadcast packet, while the broadcast packet is also to check the corresponding policy route.

The route map which is used to match the policy route can match the packet according to the access list or the packet's length. Various policy requirements can be satisfied through the usage of the access list, such as source-address-based routing and application-based routing.

You can set the egress port, nexthop, tos and precedence for the policy route. When the policy route is used, the order to select the route is: set ip nexthop, set interface, non-default normal route, set ip default nexthop, set default interface, normal route or default route. The policy route can set tos and precedence uniquely for normal routes.

The availability of nexthop means that the nexthop can be used to find a route in the routing table. The interface availability means that the IP protocol on the interface is up and the interface has a legal IP address.

### Example

The following example shows that the policy routing can be conducted to a packet received by interface s1/1 and the packet whose destination address is 100.0.0.0/8 can be transmitted to interface s1/0:

```
interface s1/1
ip policy route-map Policy
!
route-map Policy
match ip address Policy-ACL
set interface s1/0
!
ip access-list extended
permit ip any 100.0.0.0 255.0.0.0
!
```

### Related command

**ip local policy**

**show ip local policy**

**show ip policy**

### 7.1.13 ip route-weight

To configure the route weight on an interface, run **ip route-weight**. To resume the original route weight on an interface, run **no route-weight**. The original value of the route weight is 1.

**ip route-weight** [*value*]

**no ip route-weight**

### Parameter

Parameter	Description
<i>value</i>	Route weight

### Default

The default value of the route weight is 1.

## Command mode

Port configuration mode

## Explanation

You can configure the **ip route-weight** command on an interface to realize rate-based flow distribution.

At first, you need to configure the **ip route load-balance** command in global mode; then, you need to configure the route weight at the egress port of the equivalence route according to the flow distribution rate. In this way, the packet can be transmitted on different egress ports of the equivalence route according to the configured rate. In this case, you must disable the ip cache function.

## Example

The following example shows that the packet is transmitted at a rate of 3:2 on interface f0/0 and interface e1/1 after it arrives destination network 5.0.0.0.

```
interface FastEthernet0/0
ip route-weight 3
ip address 3.0.0.1 255.0.0.0
no ip directed-broadcast
!
interface Ethernet1/1
ip route-weight 2
ip address 8.0.0.1 255.0.0.0
no ip directed-broadcast
duplex half
!
ip route load-balance
ip route 5.0.0.0 255.0.0.0 FastEthernet0/0 1.2.3.5 2
ip route 5.0.0.0 255.0.0.0 Ethernet1/1 2.2.3.5 2
```

The route weight of interface f0/0 is set to 3, while the route weight of interface 1/1 is set to 2.

## Related command

**ip route load-balance**

**ip route-cache**

7.1.14 **show ip local policy**

**show ip policy**

Parameter

None

Default

None

Command mode

EXEC mode

Explanation

**show ip local policy** 命令用来显示本地策略路由的配置状态。

Example

None

Related command

**ip local policy**

**ip policy**

**show ip policy**

7.1.15 show ip policy

**show ip policy**

Parameter

None

Default

None

Command mode

EXEC mode



### Explanation

The **show ip policy** command is used to display the configuration state of the policy route.

### Example

None

### Related command

**ip local policy**

**ip policy**

**show ip local policy**

## Chapter 8 DNSR Configuration Commands

### 8.1 DNSR Configuration Commands

Terminal configuration commands include:

ip domain lookup

ip domain name-server

ip domain name

ip domain list

ip host

ip domain retry

ip domain timeout

clear ip host

ip domain primary-server

ip domain dynamic enable

ip domain dynamic period

ip domain bind

show ip host

debug ip domain

#### 8.1.1 ip domain lookup

**ip domain lookup**

**no ip domain lookup**

##### **Parameter**

None

##### **Default**

The DNS-based name address resolution is activated by default.

## Command mode

Global configuration mode

## Explanation

The DNS-based name address resolution is activated by default. To delete the DNS-based name address resolution, run **no ip domain lookup**.

## Example

The following example shows how to activate the DNS-based name address resolution.

```
ip domain lookup
```

### 8.1.2 ip domain name-server

The command is used to specify the address of the DNS. If you want to delete the DNS, you can run the “no” form of the command.

```
ip domain name-server ip-address
```

```
no ip domain name-server [ip-address]
```

## Parameter

Parameter	Description
<i>ip-address</i>	IP address of the <b>syslog</b> server

## Default

The DNS is not configured.

## Command mode

Global configuration mode

## Explanation

Through the command you can specify up to six domain name servers among which the previously-designated server will be first check and the next one will be checked if the previous server is not found. If the **ip-address** parameter is not at the end the “no” form of the command, all domain name servers will be deleted.

## Example

The following example shows how to set the IP address of the DNS to 192.168.1.3.

```
ip domain name-server 192.168.1.3
```

### 8.1.3 ip domain name

The command is used to designate a default domain name. If you want to delete a default domain name, you need run the “no” form of the command.

**ip domain name** *name*

**no ip domain name**

#### Parameter

Parameter	Description
name	Default domain name

#### Default

The default domain name is not specified.

#### Command mode

Global configuration mode

#### Explanation

Only when the domain name list does not exist can the default domain name be used.

#### Example

The following example shows how to set the default domain name to **bdcom.com.cn**.

```
ip domain name bdcom.com.cn
```

### 8.1.4 ip domain list

The command is used to define the name of the domain name list. If you want to delete the domain name list, you can run the “no” form of the command.

**ip domain list** *name*

**no ip domain list** [*name*]

#### Parameter

Parameter	Description
<i>name</i>	Name of the domain name list

**Default**

The domain name list is not configured.

**Command mode**

Global configuration mode

**Explanation**

The resolver can complete the incomplete host's name through the configured domain name. The resolver will try the domain names in the domain name list one by one when it resolves the domain name until the corresponding host is found or the domain name list is completely checked. If a domain list exists, the default domain name will not be used. Up to six domain lists can be configured. If you run **no ip domain list *name***, the domain name will be deleted; if you run **no ip domain list**, all the domain names in the domain name list will be deleted.

**Example**

The following example shows that the **com.cn** domain list and the **edu.cn** domain list.

```
ip domain list com.cn
ip domain list edu.cn
```

**8.1.5 ip host**

To define the name-address mapping of the static host, run **ip host name *hostname* *address***. To delete the name-address mapping of the static host, run **no ip host name *hostname***.

```
ip host name address1 [address2, ...]
```

```
no ip host name [address1, ...]
```

**Parameter**

Parameter	Description
<i>name</i>	Name of the host
<i>address</i>	IP address of the host

**Default**

No mapping is configured.

**Command mode**

Global configuration mode

## Explanation

If the **no ip host *name*** command is not followed by the IP address, all hosts whose names are ***name*** will be deleted.

## Example

The following example shows how to set the name of the host with IP address 202.96.1.3 to **dns-server**.

```
ip host dns-server 202.96.1.3
```

One host name can be mapped to multiple IP addresses.

```
router_config# ip host djh 172.16.20.209
```

```
router_config# ip host djh 172.16.20.210
```

Or,

```
router_config# ip host djh 172.16.20.209 172.16.20.210
```

One or multiple IP addresses can be deleted at the same time, even the host can be deleted too.

```
router_config# no ip host djh 172.16.20.209          /*One IP address of djh will be
deleted*/
```

```
router_config# no ip host djh 172.16.20.209 172.16.20.210 /*The second IP address of djh
will be deleted*/
```

```
router_config# no ip host djh                        /*The djh host will be deleted*/
```

### 8.1.6 ip domain retry

The command is used to set the retry times of the DNS query. If you want to resume the default value of the retry times, you can use the “no” form of the command.

```
ip domain retry count
```

```
no ip domain retry
```

## Parameter

Parameter	Description
<i>count</i>	Retry times Range: 1-4094

## Default

The default value of the retries is three times.

## Command mode

Global configuration mode

## Example

The following example shows that the retry times is set to 5.

```
ip domain retry 5
```

### 8.1.7 ip domain timeout

The command is used to set the timeout time of DNS query retry.

**ip domain timeout** *seconds*

**no ip domain timeout**

## Parameter

Parameter	Description
<i>seconds</i>	Timeout time of the timeout retry Range: 1-4094

## Default

The default timeout time of the timeout retry is 2 seconds.

## Command mode

Global configuration mode

## Example

The following example shows how to set the timeout time of the retry to three seconds.

```
ip domain timeout 3
```

### 8.1.8 clear ip host

The command is used to delete the host's name and the address mirroring option.

**clear ip host** *name*

**clear ip host \***

## Parameter

Parameter	Description
<i>name</i>	Host's name in the to-be-deleted cache
<i>*</i>	Deletes all hosts in the cache

**Command mode**

EXEC

**Explanation**

The queried hosts will be stored in the **resolve** cache. One or all host's names and address mirroring options in the cache will be deleted. The command cannot be used to delete the statically-configured host's name and address mirroring option.

**Example**

The following example shows how to delete the [www.sina.com.cn](http://www.sina.com.cn) host in the cache.

```
clear ip host www.sina.com.cn
```

**Related command**

```
show ip host
```

**8.1.9 ip domain primary-server**

The command is used to specify the IP address of the primary DNS. If you want to delete the primary DNS, you can run the "no" form of the command.

```
ip domain primary-server address
```

```
no ip domain primary-server
```

**Parameter**

Parameter	Description
<i>address</i>	IP address of the primary domain name server (DNS).

**Default**

The primary DNS is not specified.

**Command mode**

Global configuration mode

**Explanation**

Only one primary DNS can be configured; if multiple primary domain name servers are configured, the last one will replace all previously-configured domain name servers.



### Example

The following example shows how to set the IP address of the primary DNS to 192.168.1.8.

```
ip domain primary-server 192.168.1.8
```

#### 8.1.10 ip domain dynamic enable

The command is used to activate the dynamic update function of the DNS resolver. The “no” form of the command is used to delete the dynamic update function.

**ip domain dynamic enable**

**no ip domain dynamic enable**

### Parameter

None

### Default

The dynamic update function is shut down.

### Command mode

Global configuration mode

### Example

The following example shows how to activate the dynamic update function on the DNS resolver.

```
ip domain dynamic enable
```

#### 8.1.11 ip domain dynamic period

The command is used to set the timeout value for regular domain name update and address binding. If you want to resume the default timeout value, you can use the “no” form of the command.

**ip domain dynamic period *seconds***

**no ip domain dynamic period**

### Parameter

Parameter	Description
<i>seconds</i>	Timeout value for regular domain name update and address

	binding
--	---------

## Default

The timeout time for regular domain name update and address binding is 60 seconds by default.

## Command mode

Global configuration mode

## Example

The following example shows how to set the timeout time for regular domain name update and address binding to 3600 seconds.

```
ip domain dynamic period 3600
```

### 8.1.12 ip domain bind

To bind the primary IP address and the domain name (only one domain name can be bound on one port), run **ip domain bind *name* interface *number* [*singly*]**.

The latter bound domain name will replace the former bound domain name.)

**ip domain bind *name* interface *number* [*singly*]**

**no ip domain bind *name* interface *number***

Bind/unbind the IP address to the domain name (Note: A domain name can correspond to multiple IPs and the domain name can be same to the domain name of the port.).

**ip domain bind *name* ip\_addr [*singly*]**

**no ip domain bind *name* ip\_addr**

Delete all hosts whose domain names are **name**.

**no ip domain bind *name***

## Parameter

Parameter	Description
<i>name</i>	To-be-bound domain name
<i>number</i>	To-be-bound interface number
<i>ip_addr</i>	To-be-bound IP address
singly	An optional parameter meaning that only one domain name and IP address are bound. If the parameter is followed, other hosts

	whose domain names are name on the primary DNS will be deleted.
--	---

**Default**

unbind

**Command mode**

Global configuration mode

**Example**

ip peanuthull aaa

**8.1.13 show ip host**

To display the default domain name, some relative characteristics and domain-address item in the cache, run the following commands:

**show ip hosts [detail]**

**Parameter**

Parameter	Description
<b>detail</b>	(optional) an optional parameter to display the previous content and also some information about the upper-layer uncongested-DNS-invoking application, such as message ID (or call-back function's address) and notification time.

**Command mode**

EXEC

**Example**

The following example shows how to display all hostname-address mappings:

show ip hosts

**Related command**

**clear ip host**

**8.1.14 debug ip domain**

To enable the debugging function about the DNS module, run **debug ip domain {packet|query|update|cache|proxy|peanuthull transaction|peanuthull**

**state|dyndns|all}**. To disable the debugging function of the DNS module, run **no debug ip domain {packet|query|update|cache|proxy|peanuthull transaction|peanuthull state|dyndns|all}**.

**debug ip domain**

### Parameter

None

### Command mode

EXEC

### Example

The following example shows how to enable all debugging functions of the DNSR module.

debug ip domain

## Chapter 9 PeanutHull Configuration Commands

The following commands are PeanutHull configuration commands:

```
ip peanuthull
enable
server
domain
port
username
pssword
bind
show ip peanuthull
debug ip domain
```

### 9.1.1 ip peanuthull

To add the configuration group of a designated name and enter the DHRP configuration state, run the following commands in global mode:

**ip peanuthull** *name*

To delete a name-designated configuration group, run **no ip peanuthull** *name*.

**no ip peanuthull** *name*

#### Parameter

Parameter	Description
<i>name</i>	To-be-configured name

#### Default

None

#### Command mode

Global configuration mode

**Example**

```
ip peanuthull aaa
```

**9.1.2 enable**

The DNS-based name address resolution is activated by default.

**enable**

The DNS-based name address resolution is activated by default.

**no enable****Parameter**

None

**Default**

None

**Command mode**

DHRP configuration state

**Example**

```
enable
```

**9.1.3 server**

To designate the name of a server, run the following command:

**server** *name*

To delete the name of a server, run the following command:

**no server**

**Parameter**

Parameter	Description
<i>name</i>	Name of the server

**Default**

None

**Command mode**

Peanuthull configuration state

**Explanation**

Only one primary DNS can be configured; if multiple primary domain name servers are configured, the last one will replace all previously-configured domain name servers.

**Example**

```
server ph031.oray.net
```

**9.1.4 domain**

The command is used to designate a default domain name.

**domain** *name*

The command is used to designate a default domain name.

**no domain**

**Parameter**

Parameter	Description
<i>name</i>	A domain name

**Default**

None

**Command mode**

Peanuthull configuration state

**Explanation**

To designate a default domain name, run the following commands in DHRP configuration mode:

**Example**

```
domain marstorm.vicp.net
```

### 9.1.5 port

To designate the port ID, run the following command:

**port** *num*

To delete the designated port ID, run the following command:

**no port**

#### Parameter

Parameter	Description
<i>num</i>	Port number

#### Default

None

#### Command mode

Peanuthull configuration state

#### Explanation

If the servers are different, their corresponding port numbers are also different when you set an ID for a connection port.

#### Example

port 6060

### 9.1.6 username

To set a username, run the following command:

**username** *name*

To delete a username, run the following command:

**no username**

#### Parameter

Parameter	Description
<i>name</i>	A user name



Default

None

### Command mode

Peanuthull configuration mode

### Explanation

To set the username, run **username name**. The username which requires at the registration need be maintained at the peanut-hull-designated website.

### Example

```
username bdcorn
```

## 9.1.7 password

To set a password, run the following command:

```
password password
```

To delete a password, run the following command:

```
no password
```

### Parameter

Parameter	Description
<i>password</i>	password

Default

None

### Command mode

Peanuthull configuration state

### Explanation

To set the password, run **password name**. The password which requires at the registration need be maintained at the peanut-hull-designated website.

**Example**

```
password bdcorn
```

**9.1.8 bind**

To bind an interface, run the following command:

```
bind interface number
```

To bind an IP address, run the following command:

```
bind ip_addr
```

**Parameter**

Parameter	Description
<i>number</i>	Port number
<i>ip_addr</i>	IP address

**Default**

None

**Command mode**

Peanuthull configuration state

**Explanation**

To bind the local IP address or the port to the registered domain name, run the **bind domain {inter *number*|*ip-address*}**. Multiple IP addresses may exist on a port, so the system need select the primary IP address. If the port has no IP address, the peanut-hull client cannot be started; if the IP address is allocated to the port or the IP address of the port changes, the peanut-hull client will be re-registered.

**Example**

```
bind interface f0/2
```

**9.1.9 show ip peanuthull**

To display the default domain name, domain server, host name and the running state of the client, run **show ip peanuthull**.

#### 9.1.10 debug ip domain

To display the debug information about this module, run **debug ip domain**.

## Chapter 10 Network Management Configuration Commands

### 10.1 Network Management Configuration Commands

The following commands are network management configuration commands:

distance

filter in

filter on

redistribute

#### 10.1.1 distance

To define a management distance, run **distance weight [address mask [access-list-name]]**. To delete a definition of a management distance, run **no distance weight [address mask [access-list-name]]**.

**distance weight** [address mask [access-list-name]]

**no distance weight** [address mask [access-list-name]]

#### Parameter

Parameter	Description
<b>weight</b>	Management distance, ranging between 1 and 255. You are recommended to set the value range between 10 and 255 (values from 0 to 9 reserved). If the parameter is used alone, the router will take it as the default management distance if the router does not have relative regulations about routing information. The router whose management distance is 255 will not be added to the routing table.
<i>address</i>	An optional parameter, meaning the source IP address in the <b>aa.bb.cc.dd</b> form.
<i>mask</i>	An optional parameter, meaning the mask of the IP address in the <b>aa.bb.cc.dd</b> form. If one bit is 0, the router will omit the value of the corresponding bit in the address.
<i>access-list-name</i>	An optional parameter which updates the applied standard access list name for the incoming routes.

## Default

The following table lists the default management distances:

Route Source	Default Distance
Connected	0
Static	1
External BGP	20
BIGP	90
OSPF	110
RIP	120
Internal BGP	200

## Command mode

Routing configuration mode

## Explanation

Management distance is an integer from 0 to 255. In general, the bigger the value is, the less incredible the value is. If the management distance is 255, the route source cannot be trusted and, hence, should be omitted.

If it is RIP or BEIGRP, the **address/mask** parameter indicates the IP address of the neighbor; however, the **address/mask** parameter in OSPF declares the router ID of relative LSA.

If the **access-list-name** parameter is used in the command, the access control list is applied when a route is added to the routing table. In this way, you can filter the paths of some network according to the address of the router provided by the routing information. For example, it can be used to filter the error routing information on a router even if the router is not managed by you.

The input order of the management distance values may cause unexpected effects to the allocated management distances. For details, see the following example.

The value of the **weight** parameter is subjective. No definite method can be adopted to select the value.

## Example

In the following example, you can set the RIP route by running **router rip**; you can designate the RIP route to network 192.31.7.0 and network 128.88.0.0 by running **network**; then you can run **distance** to set the default management distance of the first router to 255, which indicates that all route updates from a management-distance-unshown router will be omitted. The first **distance** command is used to set the management distance of all routers in the C-type network 192.31.7.0. The second **distance** command is used to set the management distance of router 128.88.1.3 to 120:

```

router rip
network 192.168.7.0
network 133.8.0.0
distance 255
distance 90 192.168.7.0 0.0.0.255
distance 120 133.8.1.3 0.0.0.0

```

### 10.1.2 filter in

To filter the networks received during route update, run **filter**. To modify or cancel the filter, run **no filter**.

**filter \* in access-list** {*access-list-name*}

**filter \* in gateway** {*access-list-name*}

**filter \* in prefix** {*prefix-list-name*}

**filter type number in access-list** {*access-list-name*}

**filter type number in gateway** {*access-list-name*}

**filter type number in prefix** {*prefix-list-name*}

**no filter \* in**

**no filter type number in**

#### Parameter

Parameter	Description
<i>access-list-name</i>	Name of the standard IP access control list, which defines what type of networks will be accepted and what type of networks will be limited during the route update
<i>prefix-list-name</i>	Name of the standard IP prefix list, which defines what type of networks will be accepted and what type of networks will be limited during the route update
<b>In</b>	Applies the access control list to the incoming route weight.
<b>type</b>	Type of the interface (optional)
<i>number</i>	An optional parameter, which specifies an interface where the incoming/outgoing update access list is applied. If the interface is not specified, the incoming/outgoing update access list will be applied on all incoming/outgoing interfaces.

#### Default

Invalid state

## Command mode

Routing configuration mode

## Explanation

This command is used to filter the networks in the received update information.

## Example

The following example shows that the RIP route accepts only network 0.0.0.0 and network 131.108.0.0:

```
access-list 1 permit 0.0.0.0
access-list 1 permit 131.108.0.0
access-list 1 deny 0.0.0.0 255.255.255.255
router rip
network 131.108.0.0
filter * in 1
```

## Related command

**filter out**

### 10.1.3 filter out

To restrain some networks from being declared in the update, run **filter out**. To cancel this function, run **no filter out**.

**filter \* out access-list** {*access-list-name*}

**filter \* out gateway** {*access-list-name*}

**filter \* out prefix** { *prefix-list-name*}

**filter type number out access-list** {*access-list-name*}

**filter type number out gateway** {*access-list-name*}

**filter type number out prefix** {*prefix-list-name*}

**no filter \* out**

**no filter type number out**

## Parameter

Parameter	Description
<i>access-list-name</i>	Name of the standard IP access control list, which defines what

	type of networks will be accepted and what type of networks will be limited during the route update
<i>prefix-list-name</i>	Name of the standard IP prefix list, which defines what type of networks will be accepted and what type of networks will be limited during the route update
<b>Out</b>	Applies the access control list to the outgoing route weight.
<i>Interface-name</i>	Name of an interface, which is an optional parameter

**Default**

Invalid state

**Command mode**

Routing configuration mode

**Explanation**

When the network is reallocated, the name of a route process can be designated by the **filter** command as an optional affix parameter. Hence, the access control list is applied to those routes which are obtained from the designated route process. After the process-relative access control list is applied, any designated access control list without process name is applied. The address which is not designated during the filtration will not be declared in the outgoing route update.

**Note:**

If you want to filter the networks in the received update information, you should run **filter in**.

**Example**

The following example shows that only network 131.108.0.0 can be declared by the RIP routing process:

```
access-list 1 permit 131.108.0.0
access-list 1 deny 0.0.0.0 255.255.255.255
router rip
network 131.108.0.0
filter * out 1
```

**Related command**

**filter in**



## 10.1.4 redistribute

To reallocate a route from one route domain to another route domain, run **redistribute protocol [process-id] [route-map map-name]**; to cancel reallocating the route, run **no redistribute protocol [process-id] [route-map map-name]**.

**redistribute protocol [process-id] [route-map map-name]**

**no redistribute protocol [process-id] [route-map map-name]**

## Parameter

Parameter	Description
<b>protocol</b>	<p>Source protocol for redistributing the routes, which contains one of the following keywords such as BGP, OSPF, static[ip], connected or RIP.</p> <p>The <b>static [ip]</b> keyword is used to redistribute the static IP route. The optional keyword <b>ip</b> will be used when the route is redistributed to IS-IS.</p> <p>The keyword <b>connected</b> means those routes which are automatically established on the interface after IP is activated. The routing protocols, such as OSPF and IS-IS, are called as exterior routes of the automatic system and will be redistributed.</p>
<b>process-id</b>	<p>An optional parameter which is a 16-bit autonomous system number for BGP or BIGP</p> <p>For OSPF, the parameter is an ID of the OSPF process whose routing key is redistributed. In this way, the routing process is identified. It is a non-zero decimal number.</p> <p>For RIP, the process identifier <b>process-id</b> is not required.</p>
<i>route-map</i>	<p>An optional parameter which enables the route mapping to filter the routes imported to the current routing protocol from the source protocol. If the parameter is not given, all routes will be redistributed. If the parameter is given but the identifier of the route mapping is not, no route will be imported.</p>

## Default

The redistributed route is in the invalid state.

**protocol**—means that no routing protocol is defined.

**process-id**—means that no process ID is defined.

**Route-map map-tag**—If the **route-map** parameter is not given, all routes will be redistributed. If the **map-tag** parameter is not entered, no route will be imported.

## Command mode

Routing configuration mode

## Explanation

If one keyword is modified or invalidated, other keywords would not be affected.

When a router receives a link-state packet with an internal route weight, it will take the sum of the weight value of the redistributed router and the weight value of destination arrival declaration as its weight value, while the external route weight value only considers the weight value of destination arrival in the declaration.

The redistributed route information will be filtered by this command, which guarantees that only the administrator-designated routes can enter the received routing protocol.

No matter when you run **redistribute** or **default-information** to redistribute the routes to OSPF, the router will automatically function as the autonomous-system boundary router (ASBR). However, ASBR does not generate a default route towards the OSPF route domain.

When a route is being redistributed in the OSPF process, the OSPF route weight will be used.

When a route is redistributed to OSPF and if the **metric** keyword is not used to designate the route weight, for the routes from other protocols (BGP protocol excluded), OSPF's default route weight is 20. Further speaking, when a route is redistributed between two OSPF processes in the same router and if the default route weight is not designated, a route weight in a route process will be applied to the redistributed process.

When a route is redistributed to OSPF and if the **subnets** keyword is not given, those new routes without subnet can be redistributed.

The **connected** routes affected by the **redistribute** command are not designated by the **network** command. The **default-metric** command cannot be used to affect and declare the weight of the connected route.

### Note:

The routes from IGP/EGP/BGP must not be redistributed unless the **default-information originate** command is run.

## Example

The following example shows how to redistribute the OSPF route to the BGP route domain:

```
router bgp 109
redistribute ospf...
```

The following example shows how to redistribute the route in the designated RIP to the OSPF domain:

```
router ospf 109
```

```
redistribute rip
```

The following example shows that network 20.0.0.0 in OSPF1 is declared as an external cost-100 link state.

```
interface ethernet 0
ip address 20.0.0.1 255.0.0.0
ip ospf cost 100
interface ethernet 1
ip address 10.0.0.1 255.0.0.0
!
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
redistribute ospf 2
router ospf 2
network 20.0.0.0 0.255.255.255 area 0
```