



Configuring Internal BGP Features

This module describes how to configure internal Border Gateway Protocol (BGP) features. Internal BGP (iBGP) refers to running BGP on networking devices within one autonomous system. BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains (autonomous systems) that contain independent routing policies. Many companies now have large internal networks, and there are many issues involved in scaling the existing internal routing protocols to match the increasing traffic demands while maintaining network efficiency.

- [Information About Internal BGP Features, on page 1](#)
- [How to Configure Internal BGP Features, on page 6](#)
- [Configuration Examples for Internal BGP Features, on page 19](#)
- [Additional References for Internal BGP Features, on page 21](#)
- [Feature Information for Configuring Internal BGP Features, on page 23](#)

Information About Internal BGP Features

BGP Routing Domain Confederation

One way to reduce the internal BGP (iBGP) mesh is to divide an autonomous system into multiple subautonomous systems and group them into a single confederation. To the outside world, the confederation looks like a single autonomous system. Each autonomous system is fully meshed within itself and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have external BGP (eBGP) sessions, they exchange routing information as if they were iBGP peers. Specifically, the next hop, Multi Exit Discriminator (MED) attribute, and local preference information are preserved. This feature allows the you to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

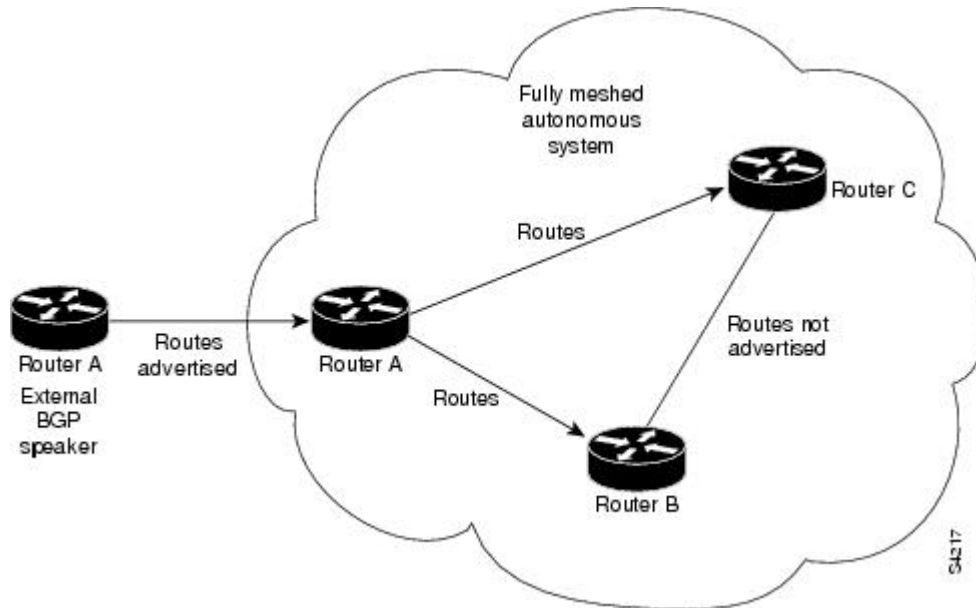
To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number.

BGP Route Reflector

BGP requires that all iBGP speakers be fully meshed. However, this requirement does not scale well when there are many iBGP speakers. Instead of configuring a confederation, another way to reduce the iBGP mesh is to configure a route reflector.

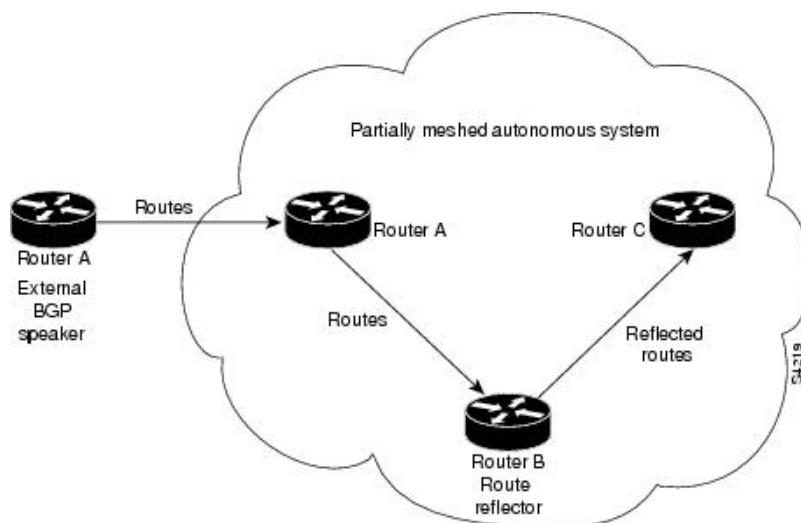
The figure below illustrates a simple iBGP configuration with three iBGP speakers (Routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both routers B and C. Routers B and C do not readvertise the iBGP learned route to other iBGP speakers because the routers do not pass on routes learned from internal neighbors to other internal neighbors, thus preventing a routing information loop.

Figure 1: Three Fully Meshed iBGP Speakers



With route reflectors, all iBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an iBGP peer is configured to be a route reflector responsible for passing iBGP learned routes to a set of iBGP neighbors. In the figure below, Router B is configured as a route reflector. When the route reflector receives routes advertised from Router A, it advertises them to Router C, and vice versa. This scheme eliminates the need for the iBGP session between Routers A and C.

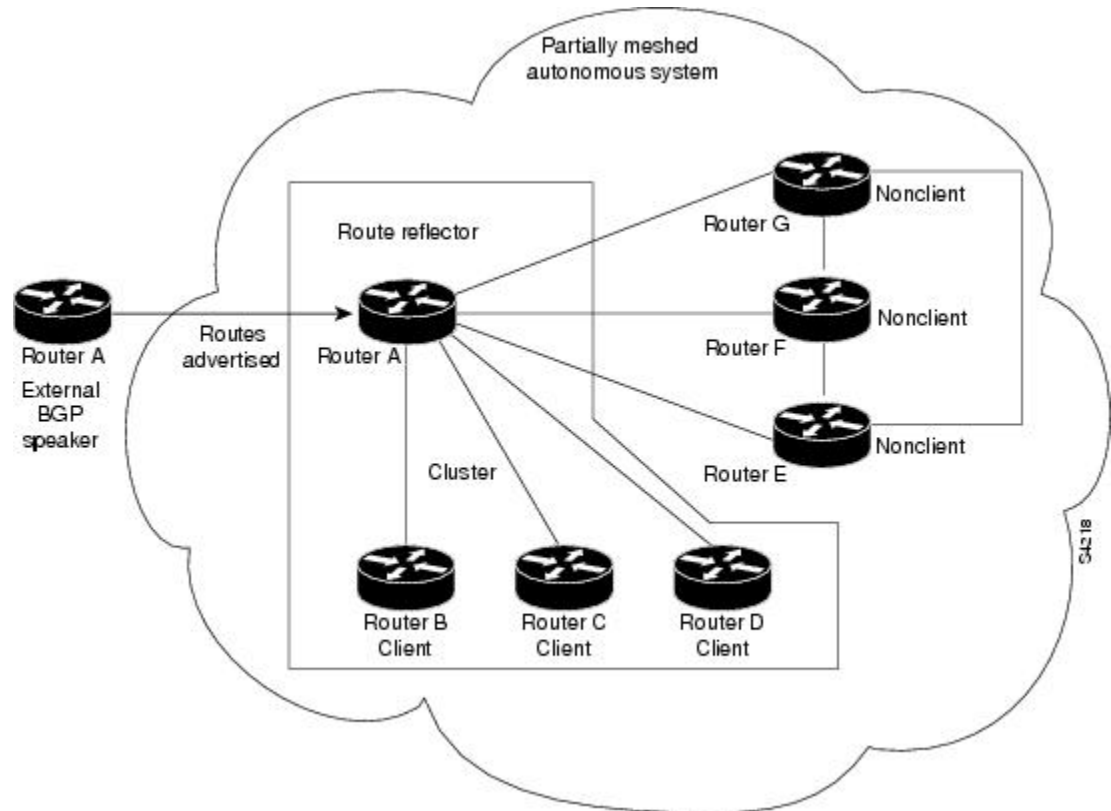
Figure 2: Simple BGP Model with a Route Reflector



The internal peers of the route reflector are divided into two groups: client peers and all the other routers in the autonomous system (nonclient peers). A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster*. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with iBGP speakers outside their cluster.

The figure below illustrates a more complex route reflector scheme. Router A is the route reflector in a cluster with routers B, C, and D. Routers E, F, and G are fully meshed, nonclient routers.

Figure 3: More Complex BGP Route Reflector Model



When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups allowing an easy and gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All the other iBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group

or nonclient group. In a simple configuration, the backbone could be divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all the route reflectors will be fully meshed). The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

Usually a cluster of clients will have a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

Route Reflector Mechanisms to Avoid Routing Loops

As the iBGP learned routes are reflected, routing information may loop. The route reflector model has the following mechanisms to avoid routing loops:

- Originator ID is an optional, nontransitive BGP attribute. It is a 4-byte attribute created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.
- Cluster-list is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, and vice versa, it appends the local cluster ID to the cluster list. If the cluster list is empty, a new cluster list is created. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster list, the advertisement is ignored.
- The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map that is acted upon is the **set ip next-hop** clause.

BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer

The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.

The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map on a route reflector (RR) that is acted upon is the **set ip next-hop** clause. The **set ip next-hop** clause is applied to reflected routes.

Configuring an RR with an outbound route map allows a network administrator to modify the next hop attribute for a reflected route. By configuring a route map with the **set ip next-hop** clause, the administrator puts the RR into the forwarding path, and can configure iBGP multipath load sharing to achieve load balancing. That is, the RR can distribute outgoing packets among multiple egress points. See the “Configuring iBGP Multipath Load Sharing” module.



Caution Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should be attempted only by someone who has a good understanding of the design implications.

BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.



Note No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

Route Dampening Minimizes Route Flapping

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

BGP Route Dampening Terms

The following terms are used when describing route dampening:

- **Flap**—A route whose availability alternates repeatedly.
- **History state**—After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- **Penalty**—Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
- **Damp state**—In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- **Suppress limit**—A route is suppressed when its penalty exceeds this limit. The default value is 2000.
- **Half-life**—Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.

- Reuse limit—As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
- Maximum suppress limit—This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevent the iBGP peers from having a higher penalty for routes external to the autonomous system.

BGP Route Map Next Hop Self

The BGP Route Map Next Hop Self feature provides a way to override the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath` selectively. These settings are global for an address family. For some routes this may not be appropriate. For example, static routes may need to be redistributed with a next hop of self, but connected routes and routes learned via Interior Border Gateway Protocol (iBGP) or Exterior Border Gateway Protocol (eBGP) may continue to be redistributed with an unchanged next hop.

The BGP route map next hop self functionality modifies the existing route map infrastructure to configure a new `ip next-hop self` setting, which overrides the `bgp next-hop unchanged` and `bgp next-hop unchanged allpaths` settings.

The `ip next-hop self` setting is applicable only to VPNv4 and VPNv6 address families. Routes distributed by protocols other than BGP are not affected.

You configure a new `bgp route-map priority` setting to inform BGP that the route map will take priority over the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath`. The `bgp route-map priority` setting only impacts BGP. The `bgp route-map priority` setting has no impact unless you configure the `bgp next-hop unchanged` or `bgp next-hop unchanged allpaths` settings.

How to Configure Internal BGP Features

Configuring a Routing Domain Confederation

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number. To configure a BGP confederation identifier, use the following command in router configuration mode:

Command	Purpose
Router (config-router) # bgp confederation identifier <i>as-number</i>	Configures a BGP confederation.

In order to treat the neighbors from other autonomous systems within the confederation as special eBGP peers, use the following command in router configuration mode:

Command	Purpose
Router(config-router)# bgp confederation peers <i>as-number</i> [<i>as-number</i>]	Specifies the autonomous systems that belong to the confederation.

For an alternative way to reduce the iBGP mesh, see "[Configuring a Route Reflector, on page 7.](#)"

Configuring a Route Reflector

To configure a route reflector and its clients, use the following command in router configuration mode:

Command	Purpose
Router(config-router)# neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-reflector-client	Configures the local router as a BGP route reflector and the specified neighbor as a client.

If the cluster has more than one route reflector, configure the cluster ID by using the following command in router configuration mode:

Command	Purpose
Router(config-router)# bgp cluster-id <i>cluster-id</i>	Configures the cluster ID.

Use the **show ip bgp** command to display the originator ID and the cluster-list attributes.

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, the route reflector need not reflect routes to clients.

To disable client-to-client route reflection, use the **no bgp client-to-client reflection** command in router configuration mode:

Command	Purpose
Router(config-router)# no bgp client-to-client reflection	Disables client-to-client route reflection.

Configuring a Route Reflector Using a Route Map to a Set Next Hop for an iBGP Peer

Perform this task on an RR to set a next hop for an iBGP peer. One reason to perform this task is when you want to make the RR the next hop for routes, so that you can configure iBGP load sharing. Create a route map that sets the next hop to be the RR's address, which will be advertised to the RR clients. The route map is applied only to outbound routes from the router to which the route map is applied.



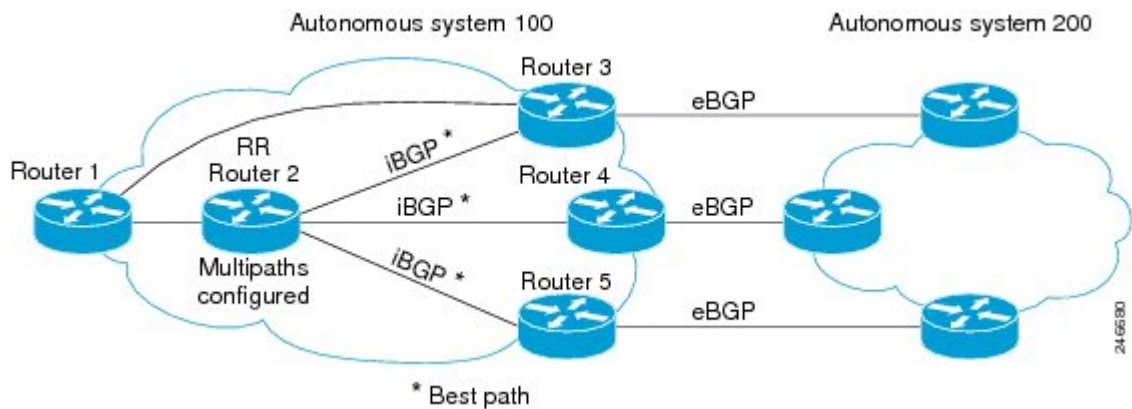
Caution Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.



Note Do not use the **neighbor next-hop-self** command to modify the next hop attribute for an RR. Using the **neighbor next-hop-self** command on the RR will modify next hop attributes only for non-reflected routes and not the intended routes that are being reflected from the RR clients. To modify the next hop attribute when reflecting a route, use an outbound route map.

This task configures the RR (Router 2) in the scenario illustrated in the figure below. In this case, Router 1 is the iBGP peer whose routes' next hop is being set. Without a route map, outbound routes from Router 1 would go to next hop Router 3. Instead, setting the next hop to the RR's address will cause routes from Router 1 to go to the RR, and thus allow the RR to perform load balancing among Routers 3, 4, and 5.

Figure 4: Route Reflector Using a Route Map to a Set Next Hop for an iBGP Peer



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag*
4. **set ip next-hop** *ip-address*
5. **exit**
6. **router bgp** *as-number*
7. **address-family ipv4**
8. **maximum-paths ibgp** *number*
9. **neighbor** *ip-address* **remote-as** *as-number*
10. **neighbor** *ip-address* **activate**
11. **neighbor** *ip-address* **route-reflector-client**
12. **neighbor** *ip-address* **route-map** *map-name* **out**
13. Repeat Steps 12 through 14 for the other RR clients.
14. **end**

15. show ip bgp neighbors

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	route-map map-tag Example: Router(config)# route-map rr-out	Enters route map configuration mode to configure a route map. <ul style="list-style-type: none">• The route map is created to set the next hop for the route reflector client.
Step 4	set ip next-hop ip-address Example: Router(config-route-map)# set ip next-hop 10.2.0.1	Specifies that for routes that are advertised where this route map is applied, the next-hop attribute is set to this IPv4 address. <ul style="list-style-type: none">• For this task, we want to set the next hop to be the address of the RR.
Step 5	exit Example: Router(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 6	router bgp as-number Example: Router(config)# router bgp 100	Enters router configuration mode and creates a BGP routing process.
Step 7	address-family ipv4 Example: Router(config-router-af)# address-family ipv4	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.
Step 8	maximum-paths ibgp number Example: Router(config-router)# maximum-paths ibgp 5	Controls the maximum number of parallel iBGP routes that can be installed in the routing table.

	Command or Action	Purpose
Step 9	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: Router(config-router-af)# neighbor 10.1.0.1 remote-as 100	Adds an entry to the BGP neighbor table.
Step 10	neighbor <i>ip-address</i> activate Example: Router(config-router-af)# neighbor 10.1.0.1 activate	Enables the exchange of information with the peer.
Step 11	neighbor <i>ip-address</i> route-reflector-client Example: Router(config-router-af)# neighbor 10.1.0.1 route-reflector-client	Configures the local router as a BGP route reflector, and configures the specified neighbor as a route-reflector client.
Step 12	neighbor <i>ip-address</i> route-map <i>map-name</i> out Example: Router(config-router-af)# neighbor 10.1.0.1 route-map rr-out out	Applies the route map to outgoing routes from this neighbor. <ul style="list-style-type: none"> • Reference the route map you created in Step 3.
Step 13	Repeat Steps 12 through 14 for the other RR clients.	You will not be applying a route map to the other RR clients.
Step 14	end Example: Router(config-router-af)# end	Exits address family configuration mode and enters privileged EXEC mode.
Step 15	show ip bgp neighbors Example: Router# show ip bgp neighbors	(Optional) Displays information about the BGP neighbors, including their status as RR clients, and information about the route map configured.

Adjusting BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the Cisco software declares a peer dead. By default, the keepalive timer is 60 seconds, and the hold-time timer is 180 seconds. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated hold time and the configured keepalive time.

To adjust BGP timers for all neighbors, use the following command in router configuration mode:

Command	Purpose
Device(config-router)# timers bgp <i>keepalive holdtime</i>	Adjusts BGP timers for all neighbors.

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in router configuration mode:

Command	Purpose
Device(config-router)# neighbor [<i>ip-address</i> <i>peer-group-name</i>] timers <i>keepalive holdtime</i>	Sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group.



Note The timers configured for a specific neighbor or peer group override the timers configured for all BGP neighbors using the **timers bgp** router configuration command.

To clear the timers for a BGP neighbor or peer group, use the **no** form of the **neighbor timers** command.

Configuring the Router to Consider a Missing MED as the Worst Path

To configure the router to consider a path with a missing MED attribute as the worst path, use the following command in router configuration mode:

Command	Purpose
Router(config-router)# bgp bestpath med missing-as-worst	Configures the router to consider a missing MED as having a value of infinity, making the path without a MED value the least desirable path.

Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths

To configure the router to consider the MED value in choosing a path, use the following command in router configuration mode:

Command	Purpose
Router(config-router)# bgp bestpath med confed	Configures the router to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.

The comparison between MEDs is made only if there are no external autonomous systems in the path (an external autonomous system is an autonomous system that is not within the confederation). If there is an external autonomous system in the path, then the external MED is passed transparently through the confederation, and the comparison is not made.

The following example compares route A with these paths:

```

path= 65000 65004, med=2
path= 65001 65004, med=3
path= 65002 65004, med=4
path= 65003 1, med=1

```

In this case, path 1 would be chosen if the **bgp bestpath med confed router configuration** command is enabled. The fourth path has a lower MED, but it is not involved in the MED comparison because there is an external autonomous system in this path.

Configuring the Router to Use the MED to Choose a Path in a Confederation

To configure the router to use the MED to choose the best path from among paths advertised by a single subautonomous system within a confederation, use the following command in router configuration mode:

Command	Purpose
Router(config-router)# bgp deterministic med	Configures the router to compare the MED variable when choosing among routes advertised by different peers in the same autonomous system.



Note If the **bgp always-compare-med** router configuration command is enabled, all paths are fully comparable, including those from other autonomous systems in the confederation, even if the **bgp deterministic med** command is also enabled.

Enabling and Configuring BGP Route Dampening

Perform this task to enable and configure BGP route dampening.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp as-number**
4. **address-family ipv4 [unicast | multicast | vrf vrf-name]**
5. **bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name]**
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp as-number Example: <pre>Router(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 4	address-family ipv4 [unicast multicast vrf vrf-name] Example: <pre>Router(config-router)# address-family ipv4 unicast</pre>	Specifies the IPv4 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the unicast keyword is not specified with the address-family ipv4 command. • The multicast keyword specifies IPv4 multicast address prefixes. • The vrf keyword and <i>vrf-name</i> argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name] Example: <pre>Router(config-router-af)# bgp dampening 30 1500 10000 120</pre>	Enables BGP route dampening and changes the default values of route dampening factors. <ul style="list-style-type: none"> • The <i>half-life</i>, <i>reuse</i>, <i>suppress</i>, and <i>max-suppress-time</i> arguments are all position dependent; if one argument is entered then all the arguments must be entered. • Use the route-map keyword and <i>map-name</i> argument to control where BGP route dampening is enabled.
Step 6	end Example: <pre>Router(config-router-af)# end</pre>	Exits address family configuration mode and enters privileged EXEC mode.

Monitoring and Maintaining BGP Route Dampening

You can monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life. To display flap statistics, use the following commands as needed:

Command	Purpose
Router# show ip bgp dampening flap-statistics	Displays BGP flap statistics for all paths.
Router# show ip bgp dampening flap-statistics regexp <i>regexp</i>	Displays BGP flap statistics for all paths that match the regular expression.
Router# show ip bgp dampening flap-statistics filter-list access- <i>list</i>	Displays BGP flap statistics for all paths that pass the filter.
Router# show ip bgp dampening flap-statistics ip-address mask	Displays BGP flap statistics for a single entry.
Router# show ip bgp dampening flap-statistics ip-address mask longer-prefix	Displays BGP flap statistics for more specific entries.

To clear BGP flap statistics (thus making it less likely that the route will be dampened), use the following commands as needed:

Command	Purpose
Router# clear ip bgp flap-statistics	Clears BGP flap statistics for all routes.
Router# clear ip bgp flap-statistics regexp <i>regexp</i>	Clears BGP flap statistics for all paths that match the regular expression.
Router# clear ip bgp flap-statistics filter-list <i>list</i>	Clears BGP flap statistics for all paths that pass the filter.
Router# clear ip bgp flap-statistics <i>ip-address mask</i>	Clears BGP flap statistics for a single entry.
Router# clear ip bgp <i>ip-address</i> flap-statistics	Clears BGP flap statistics for all paths from a neighbor.



Note The flap statistics for a route are also cleared when a BGP peer is reset. Although the reset withdraws the route, there is no penalty applied in this instance, even if route flap dampening is enabled.

Once a route is dampened, you can display BGP route dampening information, including the time remaining before the dampened routes will be unsuppressed. To display the information, use the following command:

Command	Purpose
Router# show ip bgp dampening dampened-paths	Displays the dampened routes, including the time remaining before they will be unsuppressed.

You can clear BGP route dampening information and unsuppress any suppressed routes by using the following command:

Command	Purpose
Router# clear ip bgp dampened-paths [<i>ip-address network-mask</i>]	Clears route dampening information and unsuppresses the suppressed routes.

Configuring BGP Route Map next-hop self

Perform this task to modify the existing route map by adding the ip next-hop self setting and overriding the bgp next-hop unchanged and bgp next-hop unchanged allpaths settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* **permit** *sequence-number*
4. **match source-protocol** *source-protocol*
5. **set ip next-hop self**
6. **exit**
7. **route-map** *map-tag* **permit** *sequence-number*
8. **match route-type internal**
9. **match route-type external**
10. **match source-protocol** *source-protocol*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **next-hop unchanged allpaths**
17. **neighbor** *ip-address* **route-map** *map-name* **out**
18. **exit**
19. **address-family ipv4** [*unicast* | *multicast*] **vrf** *vrf-name*]
20. **bgp route-map priority**
21. **redistribute** *protocol*
22. **redistribute** *protocol*
23. **exit-address-family**
24. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	route-map map-tag permit sequence-number Example: Device(config)# route-map static-nexthop-rewrite permit 10	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 4	match source-protocol source-protocol Example: Device(config-route-map)# match source-protocol static	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.
Step 5	set ip next-hop self Example: Device(config-route-map)# set ip next-hop self	Configure local routes (for BGP only) with next hop of self.
Step 6	exit Example: Device(config-route-map)# exit	Exits route-map configuration mode and enters global configuration mode.
Step 7	route-map map-tag permit sequence-number Example: Device(config)# route-map static-nexthop-rewrite permit 20	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 8	match route-type internal Example: Device(config-route-map)# match route-type internal	Redistributes routes of the specified type.

	Command or Action	Purpose
Step 9	match route-type external Example: <pre>Device(config-route-map)# match route-type external</pre>	Redistributes routes of the specified type.
Step 10	match source-protocol <i>source-protocol</i> Example: <pre>Device(config-route-map)# match source-protocol connected</pre>	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.
Step 11	exit Example: <pre>Device(config-route-map)# exit</pre>	Exits route-map configuration mode and enters global configuration mode.
Step 12	router bgp <i>autonomous-system-number</i> Example: <pre>Device(config)# router bgp 45000</pre>	Enters router configuration mode and creates a BGP routing process.
Step 13	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: <pre>Device(config-router)# neighbor 172.16.232.50 remote-as 65001</pre>	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 14	address-family vpnv4 Example: <pre>Device(config-router)# address-family vpnv4</pre>	Specifies the VPNv4 address family and enters address family configuration mode.
Step 15	neighbor <i>ip-address</i> activate Example: <pre>Device(config-router-af)# neighbor 172.16.232.50 activate</pre>	Enables the exchange of information with a Border Gateway Protocol (BGP) neighbor.
Step 16	neighbor <i>ip-address</i> next-hop unchanged allpaths Example: <pre>Device(config-router-af)# neighbor 172.16.232.50 next-hop unchanged allpaths</pre>	Enables an external EBGp peer that is configured as multihop to propagate the next hop unchanged.
Step 17	neighbor <i>ip-address</i> route-map <i>map-name</i> out Example:	Applies a route map to an outgoing route.

	Command or Action	Purpose
	<pre>Device(config-router-af)# neighbor 172.16.232.50 route-map static-nextthop-rewrite out</pre>	
Step 18	<p>exit</p> <p>Example:</p> <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode and enters router configuration mode.
Step 19	<p>address-family ipv4 [unicast multicast vrf vrf-name]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv4 unicast vrf inside</pre>	Specifies the IPv4 address family and enters address family configuration mode.
Step 20	<p>bgp route-map priority</p> <p>Example:</p> <pre>Device(config-router-af)# bgp route-map priority</pre>	Configures the route map priority for the local BGP routing process
Step 21	<p>redistribute protocol</p> <p>Example:</p> <pre>Device(config-router-af)# redistribute static</pre>	Redistributes routes from one routing domain into another routing domain.
Step 22	<p>redistribute protocol</p> <p>Example:</p> <pre>Device(config-router-af)# redistribute connected</pre>	Redistributes routes from one routing domain into another routing domain.
Step 23	<p>exit-address-family</p> <p>Example:</p> <pre>Device(config-router-af)# exit address-family</pre>	Exits address family configuration mode and enters router configuration mode .
Step 24	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Exits router configuration mode and enters privileged EXEC mode.

Configuration Examples for Internal BGP Features

Example: BGP Confederation Configurations with Route Maps

This section contains an example of the use of a BGP confederation configuration that includes BGP communities and route maps. For more examples of how to configure a BGP confederation, see the “Example: BGP Confederation” section in this module.

This example shows how BGP community attributes are used with a BGP confederation configuration to filter routes.

In this example, the route map named *set-community* is applied to the outbound updates to neighbor 172.16.232.50 and the local-as community attribute is used to filter the routes. The routes that pass access list 1 have the special community attribute value local-as. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers outside autonomous system 200.

```
router bgp 65000
 network 10.0.1.0 route-map set-community
 bgp confederation identifier 200
 bgp confederation peers 65001
 neighbor 172.16.232.50 remote-as 100
 neighbor 172.16.233.2 remote-as 65001
!
route-map set-community permit 10
 match ip address 1
 set community local-as
!
```

Example: BGP Confederation

The following is a sample configuration that shows several peers in a confederation. The confederation consists of three internal autonomous systems with autonomous system numbers 6001, 6002, and 6003. To the BGP speakers outside the confederation, the confederation looks like a normal autonomous system with autonomous system number 500 (specified via the **bgp confederation identifier** router configuration command).

In a BGP speaker in autonomous system 6001, the **bgp confederation peers** router configuration command marks the peers from autonomous systems 6002 and 6003 as special eBGP peers. Hence peers 172.16.232.55 and 172.16.232.56 will get the local preference, next hop, and MED unmodified in the updates. The router at 10.16.69.1 is a normal eBGP speaker and the updates received by it from this peer will be just like a normal eBGP update from a peer in autonomous system 6001.

```
router bgp 6001
 bgp confederation identifier 500
 bgp confederation peers 6002 6003
 neighbor 172.16.232.55 remote-as 6002
 neighbor 172.16.232.56 remote-as 6003
 neighbor 10.16.69.1 remote-as 777
```

In a BGP speaker in autonomous system 6002, the peers from autonomous systems 6001 and 6003 are configured as special eBGP peers. 10.70.70.1 is a normal iBGP peer and 10.99.99.2 is a normal eBGP peer from autonomous system 700.

Example: Route Reflector Using a Route Map to Set a Next Hop for an iBGP Peer

```

router bgp 6002
  bgp confederation identifier 500
  bgp confederation peers 6001 6003
  neighbor 10.70.70.1 remote-as 6002
  neighbor 172.16.232.57 remote-as 6001
  neighbor 172.16.232.56 remote-as 6003
  neighbor 10.99.99.2 remote-as 700

```

In a BGP speaker in autonomous system 6003, the peers from autonomous systems 6001 and 6002 are configured as special eBGP peers. 10.200.200.200 is a normal eBGP peer from autonomous system 701.

```

router bgp 6003
  bgp confederation identifier 500
  bgp confederation peers 6001 6002
  neighbor 172.16.232.57 remote-as 6001
  neighbor 172.16.232.55 remote-as 6002
  neighbor 10.200.200.200 remote-as 701

```

The following is a part of the configuration from the BGP speaker 10.200.200.205 from autonomous system 701 in the same example. Neighbor 172.16.232.56 is configured as a normal eBGP speaker from autonomous system 500. The internal division of the autonomous system into multiple autonomous systems is not known to the peers external to the confederation.

```

router bgp 701
  neighbor 172.16.232.56 remote-as 500
  neighbor 10.200.200.205 remote-as 701

```

Example: Route Reflector Using a Route Map to Set a Next Hop for an iBGP Peer

The following example is based on the figure above. Router 2 is the route reflector for the clients: Routers 1, 3, 4, and 5. Router 1 is connected to Router 3, but you don't want Router 1 to forward traffic destined to AS 200 to use Router 3 as the next hop (and therefore use the direct link with Router 3); you want to direct the traffic to the RR, which can load share among Routers 3, 4, and 5.

This example configures the RR, Router 2. A route map named rr-out is applied to Router 1; the route map sets the next hop to be the RR at 10.2.0.1. When Router 1 sees that the next hop is the RR address, Router 1 forwards the routes to the RR. When the RR receives packets, it will automatically load share among the iBGP paths. A maximum of five iBGP paths are allowed.

Router 2

```

route-map rr-out
  set ip next-hop 10.2.0.1
!
interface gigabitethernet 0/0
  ip address 10.2.0.1 255.255.0.0
router bgp 100
  address-family ipv4 unicast
  maximum-paths ibgp 5
  neighbor 10.1.0.1 remote-as 100
  neighbor 10.1.0.1 activate
  neighbor 10.1.0.1 route-reflector-client
  neighbor 10.1.0.1 route-map rr-out out
!
neighbor 10.3.0.1 remote-as 100

```

```

neighbor 10.3.0.1 activate
neighbor 10.3.0.1 route-reflector-client
!
neighbor 10.4.0.1 remote-as 100
neighbor 10.4.0.1 activate
neighbor 10.4.0.1 route-reflector-client
!
neighbor 10.5.0.1 remote-as 100
neighbor 10.5.0.1 activate
neighbor 10.5.0.1 route-reflector-client
end

```

Example: Configuring BGP Route Map next-hop self

This section contains an example of how to configure BGP Route Map next-hop self.

In this example, a route map is configured that matches the networks where you wish to override settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath`. Subsequently, `next-hop self` is configured. After this, the `bgp route map priority` is configured for the specified address family so that the previously specified route map takes priority over the settings for `bgp next-hop unchanged` and `bgp next-hop unchanged allpath`. This configuration results in static routes being redistributed with a next hop of self, but connected routes and routes learned via IBGP or EBGp continue to be redistributed with an unchanged next hop.

```

route-map static-nexthop-rewrite permit 10
match source-protocol static
set ip next-hop self
route-map static-nexthop-rewrite permit 20
match route-type internal
match route-type external
match source-protocol connected
!
router bgp 65000
neighbor 172.16.232.50 remote-as 65001
address-family vpnv4
neighbor 172.16.232.50 activate
neighbor 172.16.232.50 next-hop unchanged allpaths
neighbor 172.16.232.50 route-map static-nexthop-rewrite out
exit-address-family
address-family ipv4 unicast vrf inside
bgp route-map priority
redistribute static
redistribute connected
exit-address-family
end

```

Additional References for Internal BGP Features

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Related Topic	Document Title
BGP overview	“Cisco BGP Overview” module
Basic BGP configuration tasks	“Configuring a Basic BGP Network” module
iBGP multipath load sharing	“iBGP Multipath Load Sharing” module
Connecting to a service provider	“Connecting to a Service Provider Using External BGP” module
Configuring features that apply to multiple IP routing protocols	<i>IP Routing: Protocol-Independent Configuration Guide</i>

RFCs

RFC	Title
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1773	<i>Experience with the BGP Protocol</i>
RFC 1774	<i>BGP-4 Protocol Analysis</i>
RFC 1930	<i>Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5396	<i>Textual Representation of Autonomous system (AS) Numbers</i>
RFC 5398	<i>Autonomous System (AS) Number Reservation for Documentation Use</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Internal BGP Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Table 1: Feature Information for Configuring Internal BGP Features

Feature Name	Releases	Feature Configuration Information
Configuring internal BGP features	10.3 12.0(7)T 12.0(32)S12 12.2(33)SRA 12.2(33)SXH	<p>All the features contained in this module are considered to be legacy features and will work in all trains release images.</p> <p>The following commands were introduced or modified by these features:</p> <ul style="list-style-type: none"> • bgp always-compare-med • bgp bestpath med confed • bgp bestpath med missing-as-worst • bgp client-to-client reflection • bgp cluster-id • bgp confederation identifier • bgp confederation peers • bgp dampening • bgp deterministic med • clear ip bgp dampening • clear ip bgp flap-statistics • neighbor route-reflector-client • neighbor timers • show ip bgp • show ip bgp dampening dampened-paths • show ip bgp dampening flap-statistics • timers bgp

Feature Name	Releases	Feature Configuration Information
BGP Outbound Route Map on Route Reflector to Set IP Next Hop	12.0(16)ST 12.0(22)S 12.2 12.2(14)S 15.0(1)S	The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.