Configuring Advanced BGP Features

Last Updated: April 13, 2012

This module describes configuration tasks for various advanced Border Gateway Protocol (BGP) features. BGP is an interdomain routing protocol designed to provide loop-free routing between organizations. This module contains tasks to configure BGP next-hop address tracking, BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability, route dampening, Bidirectional Forwarding Detection (BFD) support for BGP, BGP MIB support and BGP support for Multi-Topology Routing (MTR).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Configuring Advanced BGP Features

Before configuring advanced BGP features you should be familiar with the "Cisco BGP Overview" module and the "Configuring a Basic BGP Network" module.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuring Advanced BGP Features

- A router that runs Cisco IOS software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.
- Multicast BGP peer support is not available in Cisco IOS software after Release 12.2(33)SRA.

Information About Configuring Advanced BGP Features

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco IOS software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS). For more details about configuring a basic BGP network, see the “Configuring a Basic BGP Network” module.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. For more details about connecting to external BGP peers, see the “Connecting to a Service Provider Using External BGP” chapter.

Although BGP is referred to as an exterior gateway protocol (EGP) many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions. For more details about internal BGP peers, see the “Configuring Internal BGP Features” chapter of the *Cisco IOS IP Routing Configuration Guide*.

Note BGP requires more configuration than other routing protocols and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

BGP Support for Next-Hop Address Tracking

To configure BGP next-hop address tracking you should understand the following concepts:

BGP Next-Hop Address Tracking

The BGP next-hop address tracking feature is enabled by default when a supporting Cisco software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best-path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

Default BGP Scanner Behavior

BGP monitors the next hop of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. By default, the BGP scanner is used to poll the RIB for this information every 60 seconds. During the 60 second time period between scan cycles, Interior Gateway Protocol (IGP) instability or other network failures can cause black holes and routing loops to temporarily form.

Selective BGP Next-Hop Route Filtering

In Cisco IOS Release 12.4(4)T, 12.2(33)SRB, and later releases, BGP selective next-hop route filtering was implemented as part of the BGP Selective Address Tracking feature to support BGP next-hop address tracking. Selective next-hop route filtering uses a route map to selectively define routes to help resolve the BGP next hop.

The ability to use a route map with the **bgp nexthop**command allows the configuration of the length of a prefix that applies to the BGP Next_Hop attribute. The route map is used during the BGP bestpath calculation and is applied to the route in the routing table that covers the next-hop attribute for BGP prefixes. If the next-hop route fails the route map evaluation, the next-hop route is marked as unreachable. This command is per address family, so different route maps can be applied for next-hop routes in different address families.

Note The **route-map** and *map-name* keyword-argument pair in the **bgp nexthop** command are not supported in Cisco IOS Release 15.0(1)SY.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

BGP Next_Hop Attribute

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

BGP Nonstop Forwarding Awareness

To configure BGP Nonstop Forwarding (NSF) awareness you should understand the following concepts:

Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, EIGRP, OSPF, and IS-IS have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

In this document, a networking device is said to be NSF-aware if it is running NSF-compatible software. A device is said to be NSF-capable if it has been configured to support NSF; therefore, it would rebuild routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the line cards with the new FIB information.

Note Currently, EIGRP supports only NSF awareness.

Cisco Express Forwarding for NSF

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by CEF. CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version (epoch) number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

Note For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

BGP Graceful Restart for NSF

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable or NSF-aware router has graceful restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peer(s) (NSF-aware peers) need to exchange the graceful restart capability in their OPEN messages, at the time of session establishment. If both the peers do not exchange the graceful restart capability, the session will not be graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the NSF-capable device. This functionality will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

BGP NSF Awareness

BGP support for NSF requires that neighbor routers are NSF-aware or NSF-capable. NSF awareness in BGP is also enabled by the graceful restart mechanism. A router that is NSF-aware functions like a router that is NSF-capable with one exception: an NSF-aware router is incapable of performing an SSO operation.

However, a router that is NSF-aware is capable of maintaining a peering relationship with a NSF-capable neighbor during a NSF SSO operation, as well as holding routes for this neighbor during the SSO operation.

The BGP Nonstop Forwarding Awareness feature provides an NSF-aware router with the capability to detect a neighbor that is undergoing an SSO operation, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes. The deployment of BGP NSF awareness can minimize the effects of Route Processor (RP) failure conditions and improve the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

NSF awareness for BGP is not enabled by default. The **bgp graceful-restart** command is used to globally enable NSF awareness on a router that is running BGP. NSF-aware operations are also transparent to the network operator and BGP peers that do not support NSF capabilities.

Note NSF awareness is enabled automatically in supported software images for Interior Gateway Protocols, such as EIGRP, IS-IS, and OSPF. In BGP, global NSF awareness is not enabled automatically and must be started by issuing the **bgp graceful-restart** command in router configuration mode.

BGP Graceful Restart per Neighbor

In Cisco IOS Releases 12.2(33)SRC, 12.2(33)SB (on platforms including the Cisco 10000 series routers), 15.0(1)M, and later releases, the ability to enable or disable BGP graceful restart for every individual BGP neighbor was introduced. Three new methods of configuring BGP graceful restart for BGP peers, in addition to the existing global BGP graceful restart configuration, are now available. Graceful restart can be enabled or disabled for a BGP peer or a BGP peer group using the **neighbor ha-mode graceful-restart** command, or a BGP peer can inherit a graceful restart configuration from a BGP peer-session template using the **ha-mode graceful-restart**command.

Although BGP graceful restart is disabled by default, the existing global command enables graceful restart for all BGP neighbors regardless of their capabilities. The ability to enable or disable BGP graceful restart for individual BGP neighbors provides a greater level of control for a network administrator.

When the BGP graceful restart capability is configured for an individual neighbor, each method of configuring graceful restart has the same priority, and the last configuration instance is applied to the neighbor. For example, if global graceful restart is enabled for all BGP neighbors but an individual neighbor is subsequently configured as a member of a peer group for which the graceful restart is disabled, graceful restart is disabled for that neighbor.

The configuration of the restart and stale-path timers is available only with the global **bgp graceful-restart** command, but the default values are set when the **neighbor ha-mode graceful-restart**or **ha-mode graceful-restart** commands are configured. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

BGP Peer Session Templates

Peer session templates are used to group and apply the configuration of general BGP session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template.

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A BGP neighbor can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template. A BGP neighbor can directly inherit only one session template and can indirectly inherit up to seven additional peer session templates.

Peer session templates support inheritance. A directly applied peer session template can directly or indirectly inherit configurations from up to seven peer session templates. So, a total of eight peer session templates can be applied to a neighbor or neighbor group.

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

For more details about BGP peer session templates, see the section “Configuring a Basic BGP Network.”

To use a BGP peer session template to enable or disable BGP graceful restart, see the section “Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates.”

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.

Minimizing 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 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.

Understanding 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.

BFD for BGP

Bidirectional Forwarding Detection (BFD) support for BGP was introduced in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.0(32)S, 12.2(33)SRA,12.2(33)SXH, 12.2(33)SB, and later releases. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a marked decrease in reconvergence time.

One caveat exists for BFD; BFD and BGP graceful restart capability cannot both be configured on a router running BGP. If an interface goes down, BFD detects the failure and indicates that the interface cannot be used for traffic forwarding and the BGP session goes down, but graceful restart still allows traffic forwarding on platforms that support NSF even though the BGP session is down, allowing traffic forwarding using the interface that is down. Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing.

For more details about BFD, see the "Bidirectional Forwarding Detection" module of the *Cisco IOS IP Routing: BFD Configuration Guide*.

BGP MIB Support

The Management Information Base (MIB) to support BGP is the CISCO-BGP4-MIB. In Cisco IOS Release 12.0(26)S, 12.3(7)T, 12.2(25)S, 12.2(33)SRA, 12.2(33)SXH, and later releases, the BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following sections describe the objects and notifications (traps) that are supported:

BGP FSM Transition Change Support

The *cbgpRouteTable* supports BGP Finite State Machine (FSM) transition state changes.

The *cbgpFsmStateChange* object allows you to configure SNMP notifications (traps) for all FSM transition state changes. This notification contains the following MIB objects:

- *bgpPeerLastError*
- *bgpPeerState*
- *cbgpPeerLastErrorTxt*
- *cbgpPeerPrevState*

The *cbgpBackwardTransition* object supports all BGP FSM transition state changes. This object is sent each time the FSM moves to either a higher or lower numbered state. This notification contains the following MIB objects:

- *bgpPeerLastError*
- *bgpPeerState*
- *cbgpPeerLastErrorTxt*
- *cbgpPeerPrevState*

The **snmp-server enable bgp traps** command allows you to enable the traps individually or together with the existing FSM backward transition and established state traps as defined in RFC 1657.

BGP Route Received Route Support

The *cbgpRouteTable* object supports the total number of routes received by a BGP neighbor. The following MIB object is used to query the CISCO-BGP4-MIB for routes that are learned from individual BGP peers:

- *cbgpPeerAddrFamilyPrefixTable*

Routes are indexed by the address-family identifier (AFI) or subaddress-family identifier (SAFI). The prefix information displayed in this table can also viewed in the output of the **show ip bgp** command.

BGP Prefix Threshold Notification Support

The *cbgpPrefixMaxThresholdExceed* and *cbgpPrfefixMaxThresholdClear* objects were introduced to allow you to poll for the total number of routes received by a BGP peer.

The *cbgpPrefixMaxThresholdExceed* object allows you to configure SNMP notifications to be sent when the prefix count for a BGP session has exceeded the configured value. This notification is configured on a per address family basis. The prefix threshold is configured with the **neighbor maximum-prefix** command. This notification contains the following MIB objects:

- *cbgpPeerPrefixAdminLimit*
- *cbgpPeerPrefixThreshold*

The cbgpPrfefixMaxThresholdClear object allows you to configure SNMP notifications to be sent when the prefix count drops below the clear trap limit. This notification is configured on a per address family basis. This notification contains the following objects:

- *cbgpPeerPrefixAdminLimit*
- *cbgpPeerPrefixClearThreshold*

Notifications are sent when the prefix count drops below the clear trap limit for an address family under a BGP session after the cbgpPrefixMaxThresholdExceed notification is generated. The clear trap limit is calculated by subtracting 5 percent from the maximum prefix limit value configured with the **neighbor**

maximum-prefix command. This notification will not be generated if the session goes down for any other reason after the cbgpPrefixMaxThresholdExceed is generated.

VPNv4 Unicast Address Family Route Support

The *cbgpRouteTable* object allows you to configure SNMP GET operations for VPNv4 unicast address-family routes.

The following MIB object allows you to query for multiple BGP capabilities (for example, route refresh, multiprotocol BGP extensions, and graceful restart):

- *cbgpPeerCapsTable*

The following MIB object allows you to query for IPv4 and VPNv4 address family routes:

- *cbgpPeerAddrFamilyTable*

Each route is indexed by peer address, prefix, and prefix length. This object indexes BGP routes by the AFI and then by the SAFI. The AFI table is the primary index, and the SAFI table is the secondary index. Each BGP speaker maintains a local Routing Information Base (RIB) for each supported AFI and SAFI combination.

cbgpPeerTable Support

The *cbgpPeerTable* has been modified to support the enhancements described in this document. The following new table objects are supported in the CISCO-BGP-MIB.my:

- *cbgpPeerLastErrorTxt*
- *cbgpPeerPrevState*

The following table objects are not supported. The status of theses objects is listed as deprecated, and these objects are not operational:

- *cbgpPeerPrefixAccepted*
- *cbgpPeerPrefixDenied*
- *cbgpPeerPrefixLimit*
- *cbgpPeerPrefixAdvertised*
- *cbgpPeerPrefixSuppressed*
- *cbgpPeerPrefixWithdrawn*

BGP Support for MTR

BGP support for MTR was introduced in Cisco IOS Release 12.2(33)SRB. For more details, see the "Multi-Topology Routing" documentation. Before using BGP to support MTR, you should be familiar with the following concepts:

- BGP Network Scope, page 10
- MTR CLI Hierarchy Under BGP, page 11
- BGP Sessions for Class-Specific Topologies, page 12
- Topology Translation Using BGP, page 12
- Topology Import Using BGP, page 12

BGP Network Scope

A new configuration hierarchy, named scope, has been introduced into the BGP protocol. To implement MTR for BGP, the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces some new configuration modes such as router scope configuration mode. Router scope configuration mode is entered by configuring the **scope** command in router configuration mode, and a collection of routing tables is created when this command is entered. BGP commands configured under the scope hierarchy are configured for a single network (globally), or on a per-VRF basis, and are referred to as scoped commands. The scope hierarchy can contain one or more address families.

MTR CLI Hierarchy Under BGP

The BGP CLI has been modified to provide backwards compatibility for pre-MTR BGP configuration and to provide a hierarchical implementation of MTR. Router configuration mode is backwards compatible with the pre-address family and pre-MTR configuration CLI. Global commands that affect all networks are configured in this configuration mode. For address-family and topology configuration, general session commands and peer templates can be configured to be used in the address-family or topology configuration modes.

After any global commands are configured, the scope is defined either globally or for a specific VRF. Address family configuration mode is entered by configuring the **address-family** command in router scope configuration mode or router configuration mode. Unicast is the default address family if no subaddress family (SAFI) is specified. MTR supports only the IPv4 address family with a SAFI of unicast or multicast. Entering address family configuration mode from router configuration mode configures BGP to use pre-MTR-based CLI. This configuration mode is backwards compatible with pre-existing address family configurations. Entering address family configuration mode from router scope configuration mode configures the router to use the hierarchical CLI that supports MTR. Address family configuration parameters that are not specific to a topology are entered in this address family configuration mode.

BGP topology configuration mode is entered by configuring the **topology**(BGP) command in address family configuration mode. Up to 32 topologies (including the base topology) can be configured on a router. The topology ID is configured by entering the **bgp tid** command. All address family and subaddress family configuration parameters for the topology are configured here.

Note Configuring a scope for a BGP routing process removes CLI support for pre-MTR-based configuration.

The following shows the hierarchy levels that are used when configuring BGP for MTR implementation:

```
router bgp <
autonomous-system-number
>
 ! global commands

 scope {global | vrf <
vrf-name
>}
  ! scoped commands

  address-family {<
afi
>} [<
safi
>]
   ! address family specific commands

   topology {<
topology-name
> | base}
    ! topology specific commands
```

BGP Sessions for Class-Specific Topologies

MTR is configured under BGP on a per-session basis. The base unicast and multicast topologies are carried in the global (default) session. A separate session is created for each class-specific topology that is configured under a BGP routing process. Each session is identified by its topology ID. BGP performs a best-path calculation individually for each class-specific topology. A separate RIB and FIB are maintained for each session.

Topology Translation Using BGP

Depending on the design and policy requirements for your network, you may need to install routes from a class-specific topology on one router in a class-specific topology on a neighboring router. Topology translation functionality using BGP provides support for this operation. Topology translation is BGP neighbor-session based. The **neighbor translate-topology** command is configured using the IP address and topology ID from the neighbor.

The topology ID identifies the class-specific topology of the neighbor. The routes in the class-specific topology of the neighbor are installed in the local class-specific RIB. BGP performs a best-path calculation on all installed routes and installs these routes into the local class-specific RIB. If a duplicate route is translated, BGP will select and install only one instance of the route per standard BGP best-path calculation behavior.

Topology Import Using BGP

Topology import functionality using BGP is similar to topology translation. The difference is that routes are moved between class-specific topologies on the same router using BGP. This function is configured by entering the **import topology** command. The name of the class-specific topology or base topology is specified when entering this command. Best-path calculations are run on the imported routes before they are installed into the topology RIB. This command also includes a **route-map** keyword to allow you to filter routes that are moved between class-specific topologies.

How to Configure Advanced BGP Features

Configuring BGP Next-Hop Address Tracking

The tasks in this section show how configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about configuring route dampening, see “Configuring BGP Route Dampening.”

- Adjusting the Delay Interval for BGP Next-Hop Address Tracking, page 14
- Configuring BGP Selective Next-Hop Route Filtering, page 15

Disabling BGP Next-Hop Address Tracking

Perform this task to disable BGP next-hop address tracking. BGP next-hop address tracking is enabled by default under the IPv4 and VPNv4 address families. Beginning with Cisco IOS Release 12.2(33)SB6, BGP next-hop address tracking is also enabled by default under the VPNv6 address family whenever the next hop is an IPv4 address mapped to an IPv6 next-hop address.

Disabling next hop address tracking may be useful if you the network has unstable IGP peers and route dampening is not resolving the stability issues. To reenable BGP next-hop address tracking, use the **bgp nexthop**command with the **trigger** and **enable** keywords.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**] | **vpnv6** [**unicast**]]
5. no **bgp nexthop trigger enable**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 64512`	Enters router configuration mod to create or configure a BGP routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**] | **vpnv6** [**unicast**]] **Example:** `Router(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	no **bgp nexthop trigger enable** **Example:** `Router(config-router-af)# no bgp nexthop trigger enable`	Disables BGP next-hop address tracking. • Next-hop address tracking is enabled by default for IPv4 and VPNv4 address family sessions. • The example disables next-hop address tracking.

Step 6	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address-family configuration mode, and enters Privileged EXEC mode.

Adjusting the Delay Interval for BGP Next-Hop Address Tracking

Perform this task to adjust the delay interval between routing table walks for BGP next-hop address tracking.

You can increase the performance of this feature by tuning the delay interval between full routing table walks to match the tuning parameters for the Interior Gateway protocol (IGP). The default delay interval is 5 seconds. This value is optimal for a fast-tuned IGP. In the case of an IGP that converges more slowly, you can change the delay interval to 20 seconds or more, depending on the IGP convergence time.

BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]]
5. **bgp nexthop trigger delay** *delay-timer*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 64512`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]] **Example:** `Router(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**bgp nexthop trigger delay** *delay-timer* **Example:** `Router(config-router-af)# bgp nexthop trigger delay 20`	Configures the delay interval between routing table walks for next-hop address tracking. • The time period determines how long BGP will wait before starting a full routing table walk after notification is received. • The value for the *delay-timer* argument is a number from 1 to 100 seconds. The default value is 5 seconds. • The example configures a delay interval of 20 seconds.

Step 6	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address-family configuration mode, and enters privileged EXEC mode.

Configuring BGP Selective Next-Hop Route Filtering

Perform this task to configure selective next-hop route filtering using a route map to filter potential next-hop routes. This task uses prefix lists and route maps to match IP addresses or source protocols and can be used to avoid aggregate addresses and BGP prefixes being considered as next-hop routes. Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

For more examples of how to use the **bgp nexthop** command, see the Examples: Configuring BGP Selective Next-Hop Route Filtering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
5. **bgp nexthop route-map** *map-name*
6. **exit**
7. **exit**
8. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network* / *length* | **permit** *network* / *length*}[**ge** *ge-value*] [**le** *le-value*]
9. **route-map** *map-name* [**permit**| **deny**][*sequence-number*]
10. **match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*]
11. **exit**
12. **route-map** *map-name* [**permit**| **deny**][*sequence-number*]
13. **end**
14. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • 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 *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**bgp nexthop route-map** *map-name* **Example:** `Router(config-router-af)# bgp nexthop route-map CHECK-NEXTHOP`	Permits a route map to selectively define routes to help resolve the BGP next hop. • In this example the route map named CHECK-NEXTHOP is created.
Step 6	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 7	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 8	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network* / *length* | **permit** *network* / *length*}[**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25`	Creates a prefix list for BGP next-hop route filtering. • Selective next-hop route filtering supports prefix length matching or source protocol matching on a per address-family basis. • The example creates a prefix list named FILTER25 that permits routes only if the mask length is more than 25; this will avoid aggregate routes being considered as the next-hop route.
Step 9	**route-map** *map-name* [**permit**| **deny**] [*sequence-number*] **Example:** `Router(config)# route-map CHECK-NEXTHOP deny 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named CHECK-NEXTHOP is created. If there is an IP address match in the following **match** command, the IP address will be denied.

	Command or Action	Purpose
Step 10	**match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*] **Example:** `Router(config-route-map)# match ip address prefix-list FILTER25`	Matches the IP addresses in the specified prefix list. • Use the *prefix-list-name* argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 11	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 12	**route-map** *map-name* [**permit**| **deny**] [*sequence-number*] **Example:** `Router(config)# route-map CHECK-NEXTHOP permit 20`	Configures a route map and enters route map configuration mode. • In this example, all other IP addresses are permitted by route map CHECK-NEXTHOP.

Step 13	**end**
	Example:
	`Router(config-route-map)# end`
		Exits route map configuration mode and enters privileged EXEC mode.

Step 14	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp`	Displays the entries in the BGP routing table. • Enter this command to view the next-hop addresses for each route. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Example

The following example from the **show ip bgp** command shows the next-hop addresses for each route:

```
BGP table version is 7, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*  10.1.1.0/24      192.168.1.2              0             0 40000 i
*  10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
*> 172.17.1.0/24    0.0.0.0                  0         32768
```

Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart

The tasks in this section show how configure BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability. The first task enables BGP NSF globally for all BGP neighbors and suggests a few troubleshooting options. The second task describes how to adjust the BGP graceful restart timers although the default settings are optimal for most network deployments. The next three tasks demonstrate how to enable or disable BGP graceful restart for individual BGP neighbors including peer session templates and peer groups. The final task verifies the local and peer router configuration of BGP NSF.

Enabling BGP Global NSF Awareness Using BGP Graceful Restart

Perform this task to enable BGP NSF awareness globally for all BGP neighbors. BGP NSF awareness is part of the graceful restart mechanism and BGP NSF awareness is enabled by issuing the **bgp graceful-restart** command in router configuration mode. BGP NSF awareness allows NSF-aware routers to support NSF-capable routers during an SSO operation. NSF-awareness is not enabled by default and should be configured on all neighbors that participate in BGP NSF.

Note The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

Note Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see the section “BFD for BGP.”

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart** [**restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bgp graceful-restart** [**restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:** `Router(config-router)# bgp graceful-restart`	Enables the BGP graceful restart capability and BGP NSF awareness. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. • Use this command on the restarting router and all of its peers (NSF-capable and NSF-aware).

Step 5	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and enters privileged EXEC mode.

- Troubleshooting Tips, page 20
- What to Do Next, page 21

Troubleshooting Tips

To troubleshoot the NSF feature, use the following commands in privileged EXEC mode, as needed:

- **debug ip bgp** Displays open messages that advertise the graceful restart capability.
- **debug ip bgp event** Displays graceful restart timer events, such as the restart timer and the stalepath timer.
- **debug ip bgp updates** Displays sent and received EOR messages. The EOR message is used by the NSF-aware router to start the stalepath timer, if configured.

- **show ip bgp** Displays entries in the BGP routing table. The output from this command will display routes that are marked as stale by displaying the letter “S” next to each stale route.
- **show ip bgp neighbor** Displays information about the TCP and BGP connections to neighbor devices. When enabled, the graceful restart capability is displayed in the output of this command.

What to Do Next

If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the “Configuring a Basic BGP Network” module.

Configuring BGP NSF Awareness Timers

Perform this task to adjust the BGP graceful restart timers. There are two BGP graceful restart timers that can be configured. The optional **restart-time** keyword and *seconds* argument determine how long peer routers will wait to delete stale routes before a BGP open message is received. The default value is 120 seconds. The optional **stalepath-time** keyword and *seconds* argument determine how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. The default value is 360 seconds.

Note The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart** [**restart-time** *seconds*]
5. **bgp graceful-restart** [**stalepath-time** *seconds*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bgp graceful-restart** [**restart-time** *seconds*] **Example:** `Router(config-router)# bgp graceful-restart restart-time 130`	Enables the BGP graceful restart capability and BGP NSF awareness. • The **restart-time** argument determines how long peer routers will wait to delete stale routes before a BGP open message is received. • The default value is 120 seconds. The configurable range is from 1 to 3600 seconds. **Note** Only the syntax applicable to this step is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 5	**bgp graceful-restart** [**stalepath-time** *seconds*] **Example:** `Router(config-router)# bgp graceful-restart stalepath-time 350`	Enables the BGP graceful restart capability and BGP NSF awareness. • The **stalepath-time** argument determines how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. • The default value is 360 seconds. The configurable range is from 1 to 3600 seconds. **Note** Only the syntax applicable to this step is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Step 6	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and enters privileged EXEC mode.

- What to Do Next, page 22

What to Do Next

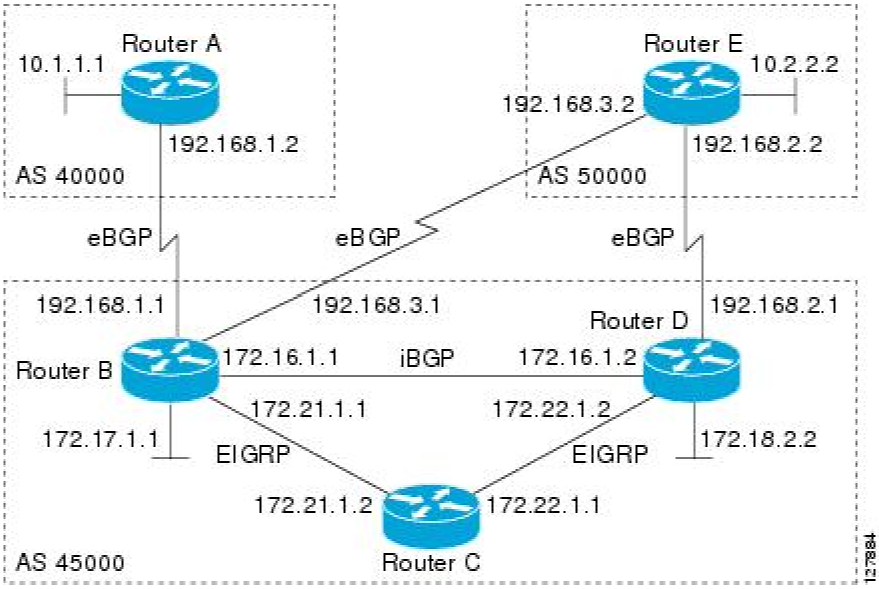
If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset the peer sessions by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the “Configuring a Basic BGP Network” module.

Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

Perform this task to enable and disable BGP graceful restart for BGP neighbors using peer session templates. In this task, a BGP peer session template is created, and BGP graceful restart is enabled. A second peer session template is created, and this template is configured to disable BGP graceful restart.

In this example, the configuration is performed at Router B in the figure below and two external BGP neighbors--at Router A and Router E in the figure below--are identified. The first BGP peer at Router A is configured to inherit the first peer session template that enables BGP graceful restart, whereas the second BGP peer at Router E inherits the second template that disables BGP graceful restart. Using the optional **show ip bgp neighbors** command, the status of the BGP graceful restart capability is verified for each BGP neighbor configured in this task.

Figure 1 Network Topology Showing BGP Neighbors



The restart and stale-path timers can be modified only using the global **bgp graceful-restart** command as shown in the Configuring BGP NSF Awareness Timers, page 21. The restart and stale-path timers are set to the default values when BGP graceful restart is enabled for BGP neighbors using peer session templates.

This task requires a Cisco IOS Release 12.2(33)SRC, or 12.2(33)SB.

Note A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode graceful-restart** [**disable**]
6. **exit-peer-session**
7. **template peer-session** *session-template-name*
8. **ha-mode graceful-restart** [**disable**]
9. **exit-peer-session**
10. **bgp log-neighbor-changes**
11. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
12. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
15. **end**
16. **show ip bgp template peer-session** [*session-template-number*]
17. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session S1`	Enters session-template configuration mode and creates a peer session template. • In this example, a peer session template named S1 is created.
Step 5	**ha-mode graceful-restart** [**disable**] **Example:** `Router(config-router-stmp)# ha-mode graceful-restart`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is enabled for the peer session template named S1.
Step 6	**exit-peer-session** **Example:** `Router(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 7	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session S2`	Enters session-template configuration mode and creates a peer session template. • In this example, a peer session template named S2 is created.
Step 8	**ha-mode graceful-restart** [**disable**] **Example:** `Router(config-router-stmp)# ha-mode graceful-restart disable`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is disabled for the peer session template named S2.
Step 9	**exit-peer-session** **Example:** `Router(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 10	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 11	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 192.168.1.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 12	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 inherit peer-session S1`	Inherits a peer session template. • In this example, the peer session template named S1 is inherited, and the neighbor inherits the enabling of BGP graceful restart.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 192.168.3.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 14	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Router(config-router)# neighbor 192.168.3.2 inherit peer-session S2`	Inherits a peer session-template. • In this example, the peer session template named S2 is inherited, and the neighbor inherits the disabling of BGP graceful restart.

Step 15	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 16	**show ip bgp template peer-session** [*session-template-number*] **Example:** `Router# show ip bgp template peer-session`	(Optional) Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the *session-template-name* argument. This command also supports all standard output modifiers.
Step 17	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]] **Example:** `Router# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 192.168.1.2.

Examples

The following example shows partial output from the show ip bgp neighbors command for the BGP peer at 192.168.1.2 (Router A in the figure above). Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can only be set using the global bgp graceful-restart command.

```
Router# show ip bgp neighbors 192.168.1.2
BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Inherits from template S1 for session parameters
  BGP version 4, remote router ID 192.168.1.2
  BGP state = Established, up for 00:02:11
  Last read 00:00:23, last write 00:00:27, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 192.168.3.2 (Router E in the figure above). Graceful restart is shown as disabled.

```
Router# show ip bgp neighbors 192.168.3.2
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Inherits from template S2 for session parameters
  BGP version 4, remote router ID 192.168.3.2
  BGP state = Established, up for 00:01:41
  Last read 00:00:45, last write 00:00:45, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.3.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Enabling BGP Graceful Restart for an Individual BGP Neighbor

Perform this task on Router B in the figure above to enable BGP graceful restart on the internal BGP peer at Router C in the figure above. Under address family IPv4, the neighbor at Router C is identified, and BGP graceful restart is enabled for the neighbor at Router C with the IP address 172.21.1.2. To verify that BGP graceful restart is enabled, the optional **show ip bgp neighbors** command is used.

This task requires a Cisco IOS Release 12.2(33)SRC, 12.2(33)SB, or 15.0(1)M.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **ha-mode graceful-restart** [**disable**]
8. **end**
9. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • 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 *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 172.21.1.2 remote-as 45000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 172.21.1.2 is an internal BGP peer because it has the same autonomous system number as the router where the BGP configuration is being entered (see Step 3).
Step 6	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 172.21.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local router. • In this example, the internal BGP peer at 172.21.1.2 is activated.
Step 7	**neighbor** *ip-address* **ha-mode graceful-restart** [**disable**] **Example:** `Router(config-router-af)# neighbor 172.21.1.2 ha-mode graceful-restart`	Enables the BGP graceful restart capability for a BGP neighbor. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is enabled for the neighbor at 172.21.1.2.

Step 8	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]] **Example:** `Router# show ip bgp neighbors 172.21.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 172.21.1.2.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.21.1.2. Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

```
Router# show ip bgp neighbors 172.21.1.2
BGP neighbor is 172.21.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.22.1.1
  BGP state = Established, up for 00:01:01
  Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
  Address tracking is enabled, the RIB does have a route to 172.21.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Disabling BGP Graceful Restart for a BGP Peer Group

Perform this task to disable BGP graceful restart for a BGP peer group. In this task, a BGP peer group is created and graceful restart is disabled for the peer group. A BGP neighbor, 172.16.1.2 at Router D in the figure above, is then identified and added as a peer group member and inherits the configuration associated with the peer group, which, in this example, disables BGP graceful restart.

This task requires a Cisco IOS Release 12.2(33)SRC, 12.2(33)SB, or 15.0(1)M.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number*
7. **neighbor** *peer-group-name* **ha-mode graceful-restart** [**disable**]
8. **neighbor** *ip-address* **peer-group** *peer-group-name*
9. **end**
10. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • 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 *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Router(config-router-af)# neighbor PG1 peer-group`	Creates a BGP peer group. - In this example, the peer group named PG1 is created.
Step 6	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor PG1 remote-as 45000`	Configures peering with a BGP peer group in the specified autonomous system. - In this example, the BGP peer group named PG1 is added to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 7	**neighbor** *peer-group-name* **ha-mode graceful-restart** [**disable**] **Example:** `Router(config-router-af)# neighbor PG1 ha-mode graceful-restart disable`	Enables the BGP graceful restart capability for a BGP neighbor. - Use the **disable** keyword to disable BGP graceful restart capability. - If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. - In this example, the BGP graceful restart capability is disabled for the BGP peer group named PG1.
Step 8	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Router(config-router-af)# neighbor 172.16.1.2 peer-group PG1`	Assigns the IP address of a BGP neighbor to a peer group. - In this example, the BGP neighbor peer at 172.16.1.2 is configured as a member of the peer group named PG1.

Step 9	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Step 10	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]] **Example:** `Router# show ip bgp neighbors 172.16.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. - In this example, the output is filtered to display information about the BGP peer at 172.16.1.2 and the "Graceful-Restart is disabled" line shows that the graceful restart capability is disabled for this neighbor.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.16.1.2. Graceful restart is shown as disabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

```
Router# show ip bgp neighbors 172.16.1.2
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
 Member of peer-group PG1 for session parameters
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Neighbor sessions:
    0 active, is multisession capable
!
Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Connections established 0; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
```

Verifying the Configuration of BGP Nonstop Forwarding Awareness

Use the following steps to verify the local configuration of BGP NSF awareness on a router and to verify the configuration of NSF awareness on peer routers in a BGP network.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]
3. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config** [*options*]

Displays the running configuration on the local router. The output will display the configuration of the **bgp graceful-restart** command in the BGP section. Repeat this command on all BGP neighbor routers to verify that all BGP peers are configured for BGP NSF awareness. In this example, BGP graceful restart is enabled globally and the external neighbor at 192.168.1.2 is configured to be a BGP peer and will have the BGP graceful restart capability enabled.

Example:

```
Router# show running-config
.
.
.
router bgp 45000
 bgp router-id 172.17.1.99
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 130
 bgp graceful-restart stalepath-time 350
 bgp graceful-restart
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 activate
.
.
.
```

Step 3 **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics**| **received prefix-filter**| **policy**[**detail**]]]

Displays information about TCP and BGP connections to neighbors. "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. In Cisco IOS Releases 12.2(33)SRC, 12.2(33)SB, or later releases, the ability to enable or disable the BGP graceful restart capability for an individual BGP neighbor, peer group or peer session template was introduced and output was added to this command to show the BGP graceful restart status.

The following partial output example using a Cisco IOS Release 12.2(33)SRC image, displays the graceful restart information for internal BGP neighbor 172.21.1.2 at Router C in the figure above. Note the "Graceful-Restart is enabled" message.

Example:

```
Router# show ip bgp neighbors 172.21.1.2
BGP neighbor is 172.21.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.22.1.1
  BGP state = Established, up for 00:01:01
  Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
  Address tracking is enabled, the RIB does have a route to 172.21.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
```

Configuring BGP Route Dampening

The tasks in this section configure and monitor BGP route dampening. Route dampening is designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

- Enabling and Configuring BGP Route Dampening, page 34
- Monitoring and Maintaining BGP Route Dampening, page 36

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** as-*number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • 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 *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 5	**bgp dampening** [*half-life reuse suppress max-suppress-time*] [**route-map** *map-name*] **Example:** `Router(config-router-af)# bgp dampening 30 1500 10000 120`	Enables BGP route dampening and changes the default values of route dampening factors. • The *half-life*, *reuse*, *suppress*, and *max-suppress-time* arguments are all position dependent; if one argument is entered then all the arguments must be entered. • Use the **route-map** keyword and *map-name* argument to control where BGP route dampening is enabled.

Step 6	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address family configuration mode and enters privileged EXEC mode.

Monitoring and Maintaining BGP Route Dampening

Perform the steps in this task as required to monitor and maintain BGP route dampening.

SUMMARY STEPS

1. **enable**
2. **show ip bgp flap-statistics** [**regexp** *regexp* | **filter-list** *access-list* | *ip-address mask* [**longer-prefix**]]
3. **clear ip bgp flap-statistics** [*neighbor-address* [*ipv4-mask*]] [**regexp** *regexp* | **filter-list** *extcom-number*]
4. **show ip bgp dampened-paths**
5. **clear ip bgp** [**ipv4** {**multicast** | **unicast**} | **ipv6**{**multicast** | **unicast**} | **vpnv4 unicast**] **dampening** [*neighbor-address*] [*ipv4-mask*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show ip bgp flap-statistics** [**regexp** *regexp* | **filter-list** *access-list* | *ip-address mask* [**longer-prefix**]]

Use this command to 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.

Example:

```
Router# show ip bgp flap-statistics
BGP table version is 10, local router ID is 172.17.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From            Flaps Duration Reuse    Path
*d 10.0.0.0         172.17.232.177  4     00:13:31 00:18:10 100
*d 10.2.0.0         172.17.232.177  4     00:02:45 00:28:20 100
```

Step 3 **clear ip bgp flap-statistics** [*neighbor-address* [*ipv4-mask*]] [**regexp** *regexp* | **filter-list** *extcom-number*]

Use this command to clear the accumulated penalty for routes that are received on a router that has BGP dampening enabled. If no arguments or keywords are specified, flap statistics are cleared for all routes. Flap statistics are also cleared when the peer is stable for the half-life time period. After the BGP flap statistics are cleared, the route is less likely to be dampened.

Example:

```
Router# clear ip bgp flap-statistics 172.17.232.177
```

Step 4 **show ip bgp dampened-paths**

Use this command to 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.

Example:

```
Router# show ip bgp dampened-paths
BGP table version is 10, local router ID is 172.29.232.182
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From             Reuse   Path
*d 10.0.0.0         172.16.232.177   00:18:4 100 ?
*d 10.2.0.0         172.16.232.177   00:28:5 100 ?
```

Step 5 **clear ip bgp** [**ipv4** {**multicast** | **unicast**} | **ipv6**{**multicast** | **unicast**} | **vpnv4 unicast**] **dampening** [*neighbor-address*] [*ipv4-mask*]

Use this command to clear stored route dampening information. If no keywords or arguments are entered, route dampening information for the entire routing table is cleared. The following example clears route dampening information for VPNv4 address family prefixes from network 192.168.10.0/24, and unsuppresses its suppressed routes.

Example:

```
Router# clear ip bgp vpnv4 unicast dampening 192.168.10.0 255.255.255.0
```

Decreasing BGP Convergence Time Using BFD

BFD support for BGP was introduced in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.2(33)SRA, 12.2(33)SXH, 12.2(33)SB, and later releases. You start a BFD process by configuring BFD on the interface. When the BFD process is started, no entries are created in the adjacency database, in other words, no BFD control packets are sent or received. The adjacency creation takes places once you have configured BFD support for the applicable routing protocols. The first two tasks must be configured to implement BFD support for BGP to reduce the BGP convergence time. The third task is an optional task to help monitor or troubleshoot BFD.

See also the "Configuring BGP Neighbor Session Options" chapter, the section "Configuring BFD for BGP IPv6 Neighbors."

Prerequisites

- Cisco Express Forwarding (CEF) and IP routing must be enabled on all participating routers.
- BGP must be configured on the routers before BFD is deployed. You should implement fast convergence for the routing protocol that you are using. See the IP routing documentation for your version of Cisco IOS software for information on configuring fast convergence.

Restrictions

- For the current Cisco implementation of BFD support for BGP in Cisco IOS Releases 12.0(31)S, 12.4(4)T, 12.2(33)SRA, 12.2(33)SXH, and 12.2(33)SB, BFD is supported only for IPv4 networks, and only asynchronous mode is supported. In asynchronous mode, either BFD peer can initiate a BFD session.
- BFD works only for directly-connected neighbors. BFD neighbors must be no more than one IP hop away. Multihop configurations are not supported.
- Configuring both BFD and BGP graceful restart for NSF on a router running BGP may result in suboptimal routing. For more details, see the BFD for BGP, page 8.

Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 6/0`	Enters interface configuration mode.
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Router(config-if)# bfd interval 50 min_rx 50 multiplier 5`	Enables BFD on the interface.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode.

Configuring BFD Support for BGP

Perform this task to configure BFD support for BGP, so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

- BGP must be running on all participating routers.
- The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See "Configuring BFD Session Parameters on the Interface" for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **fall-over bfd**
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp tag1`	Specifies a BGP process and enters router configuration mode.
Step 4	**neighbor** *ip-address* **fall-over bfd** **Example:** `Router(config-router)# neighbor 172.16.10.2 fall-over bfd`	Enables BFD support for fallover.

Step 5	**end**
	Example:
	`Router(config-router)# end`
		Returns the router to privileged EXEC mode.

Step 6	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors detail`	Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Router# show ip bgp neighbors`	Displays information about BGP and TCP connections to neighbors.

Monitoring and Troubleshooting BFD for Cisco 7600 Series Routers

To monitor or troubleshoot BFD on Cisco 7600 series routers, perform one or more of the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show bfd neighbors** [**details**]
3. **debug bfd** [**event** | **packet** | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Displays the BFD adjacency database. • The **details** keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	**debug bfd** [**event** | **packet** | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**] **Example:** `Router# debug bfd packet`	(Optional) Displays debugging information about BFD packets.

- What to Do Next, page 41

What to Do Next

For more information about configuring BFD support for another routing protocol see the "Bidirectional Forwarding Detection" configuration guide.

Enabling BGP MIB Support

SNMP notifications can be configured on the router and GET operations can be performed from an external management station only after BGP SNMP support is enabled. Perform this task on a router to configure SNMP notifications for the BGP MIB.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps bgp** [[**state-changes** [**all**] [**backward-trans**] [**limited**]] | [**threshold prefix**]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps bgp** [[**state-changes** [**all**] [**backward-trans**] [**limited**]] | [**threshold prefix**]] **Example:** `Router# snmp-server enable traps bgp`	Enables BGP support for SNMP operations. Entering this command with no keywords or arguments enables support for all BGP events. • The **state-changes** keyword is used to enable support for FSM transition events. • The **all** keyword enables support for FSM transitions events. • The **backward-trans** keyword enables support only for backward transition state change events. • The **limited** keyword enables support for backward transition state changes and established state events. • The **threshold** and **prefix** keywords are used to enable notifications when the configured maximum prefix limit is reached on the specified peer.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode, and enters privileged EXEC mode.

Configuring BGP Support for MTR

Before performing the following tasks, you must have configured MTR topologies. For more details, see the "Multi-Topology Routing" feature in Cisco IOS Release 12.2(33)SRB.

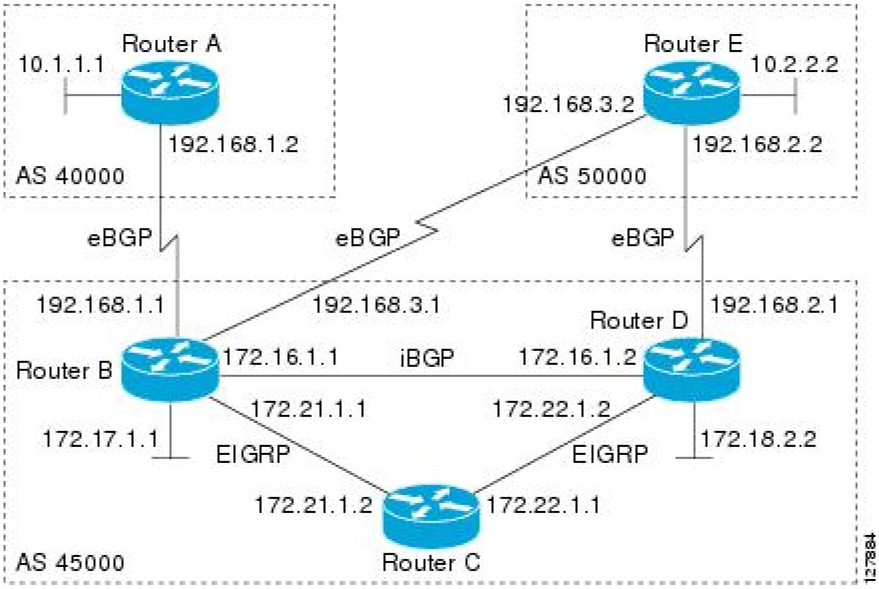
- Activating an MTR Topology Using BGP, page 43

- Importing Routes from an MTR Topology Using BGP, page 48

Activating an MTR Topology Using BGP

Perform this task to activate an MTR topology inside an address family using BGP. This task is configured on Router B in the figure below and must also be configured on Router D and Router E. In this task, a scope hierarchy is configured to apply globally and a neighbor is configured under router scope configuration mode. Under the IPv4 unicast address family, an MTR topology that applies to video traffic is activated for the specified neighbor. There is no interface configuration mode for BGP topologies.

Figure 2 BGP Network Diagram



The BGP CLI has been modified to provide backwards compatibility for pre-MTR BGP configuration and to provide a hierarchical implementation of MTR. A new configuration hierarchy, named scope, has been introduced into the BGP protocol. To implement MTR for BGP, the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces some new configuration modes such as router scope configuration mode. Router scope configuration mode is entered by configuring the **scope** command in router configuration mode, and a collection of routing tables is created when this command is entered. The following shows the hierarchy levels that are used when configuring BGP for MTR implementation:

```
router bgp <
autonomous-system-number
>
 ! global commands

 scope {global | vrf <
vrf-name
>}
  ! scoped commands

  address-family {<
afi
>} [<
safi
>]
   ! address family specific commands
   topology {<
topology-name
> | base}
    ! topology specific commands
```

Before using BGP to support MTR, you should be familiar with all the concepts documented in the BGP Support for MTR, page 10.

- You must be running a Cisco IOS Release 12.2(33)SRB, or later release, on any routers configured for MTR.
- A global MTR topology configuration has been configured and activated.
- IP routing and CEF are enabled.

Note

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only the IPv4 address family (multicast and unicast) is supported.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **scope** {**global** | **vrf** *vrf-name*}
5. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**}
7. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
8. **topology** {**base**| *topology-name*}
9. **bgp tid** *number*
10. **neighbor** *ip-address* **activate**
11. **neighbor** {*ip-address*| *peer-group-name*} **translate-topology** *number*
12. **end**
13. **clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out**| **soft** [**in** [**prefix-filter**] | **out**]]
14. **show ip bgp topology** {***** | *topology*} **summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.

Step 4	**scope** {**global** | **vrf** *vrf-name*}
	Example:
	`Router(config-router)# scope global`
		Defines the scope to the BGP routing process and enters router scope configuration mode.
		• BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode.
		• Use the **global** keyword to specify that BGP uses the global routing table.
		• Use the **vrf** keyword and *vrf-name* argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.

Step 5	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-scope)# neighbor 172.16.1.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose

Step 6	**neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**}
	Example:
	`Router(config-router-scope)# neighbor 172.16.1.2 transport multi-session`
		Enables a TCP transport session option for a BGP session.
		• Use the **connection-mode** keyword to specify the type of connection, either active or passive.
		• Use the **path-mtu-discovery** keyword to enable TCP transport path maximum transmission unit (MTU) discovery.
		• Use the **multi-session** keyword to specify a separate TCP transport session for each address family.
		• Use the **single-session** keyword to specify that all address families use a single TCP transport session.

Step 7	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:** `Router(config-router-scope)# address-family ipv4`	Specifies the IPv4 address family and enters router scope address family configuration mode. • Use the **mdt** keyword to specify IPv4 MDT address prefixes. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **unicast** keyword to specify 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. • Non-topology-specific configuration parameters are configured in this configuration mode.

Step 8	**topology** {**base**| *topology-name*}
	Example:
	`Router(config-router-scope-af)# topology VIDEO`
		Configures the topology instance in which BGP will route class-specific or base topology traffic, and enters router scope address family topology configuration mode.

Step 9	**bgp tid** *number* **Example:** `Router(config-router-scope-af-topo)# bgp tid 100`	Associates a BGP routing process with the specified topology ID. • Each topology must be configured with a unique topology ID.

Step 10	**neighbor** *ip-address* **activate**
	Example:
	`Router(config-router-scope-af-topo)#`
	`neighbor 172.16.1.2 activate`
		Enables the BGP neighbor to exchange prefixes for the NSAP address family with the local router.
		Note If you have configured a peer group as a BGP neighbor, you do not use this command because peer groups are automatically activated when any peer group parameter is configured.

	Command or Action	Purpose
Step 11	**neighbor** {*ip-address*| *peer-group-name*} **translate-topology** *number* **Example:** `Router(config-router-scope-af-topo)# neighbor 172.16.1.2 translate-topology 200`	(Optional) Configures BGP to install routes from a topology on another router to a topology on the local router. • The topology ID is entered for the *number* argument to identify the topology on the router.

Step 12	**end**
	Example:
	`Router(config-router-scope-af-topo)# end`
		(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.

Step 13	**clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out**| **soft** [**in** [**prefix-filter**] | **out**]] **Example:** `Router# clear ip bgp topology VIDEO 45000`	Resets BGP neighbor sessions under a specified topology or all topologies.
Step 14	**show ip bgp topology** {***** | *topology*} **summary** **Example:** `Router# show ip bgp topology VIDEO summary`	(Optional) Displays BGP information about a topology. • Most standard BGP keywords and arguments can be entered following the topology keyword. **Note** Only the syntax required for this task is shown. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following example shows summary output for the **show ip bgp topology** command and the VIDEO topology:

```
Router# show ip bgp topology VIDEO summary
BGP router identifier 192.168.3.1, local AS number 45000
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.1.2      4 45000     289     289        1    0    0 04:48:44        0
192.168.3.2     4 50000       3       3        1    0    0 00:00:27        0
```

- What to Do Next, page 47

What to Do Next

Repeat this task for every topology that you want to enable, and repeat this configuration on all neighbor routers that are to use the topologies. If you want to import routes from one MTR topology to another on the same router, proceed to the next task.

Importing Routes from an MTR Topology Using BGP

Perform this task to import routes from one MTR topology to another on the same router, when multiple topologies are configured on the same router. In this task, a prefix list is defined to permit prefixes from the 10.2.2.0 network, and this prefix list is used with a route map to filter routes moved from the imported topology. A global scope is configured, address family IPv4 is entered, the VIDEO topology is specified, the VOICE topology is imported, and the routes are filtered using the route map named 10NET.

- You must be running a Cisco IOS Release 12.2(33)SRB, or later release, on any routers configured for MTR.
- A global topology configuration has been configured and activated.
- IP routing and CEF are enabled.

Note

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops from occurring. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only the IPv4 address family (multicast and unicast) is supported.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network* / *length*| **permit** *network* / *length*} [**ge** *ge-value*] [**le** *le-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **scope** {**global** | **vrf** *vrf-name*}
9. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
10. **topology** {**base**| *topology-name*}
11. **import topology** {**base**| *topology-name*}[**route-map** *map-name*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network* / *length*| **permit** *network* / *length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list TEN permit 10.2.2.0/24`	Configures an IP prefix list. - In this example, prefix list TEN permits advertising of the 10.2.2.0/24 prefix depending on a match set by the **match ip address** command.
Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map 10NET`	Creates a route map and enters route map configuration mode. - In this example, the route map named 10NET is created.

Step 5	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
	Example:
	`Router(config-route-map)# match ip address prefix-list TEN`
		Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list.
		- In this example, the route map is configured to match prefixes permitted by prefix list TEN.

Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.

Step 8	**scope** {**global** | **vrf** *vrf-name*}
	Example:
	`Router(config-router)# scope global`
		Defines the scope to the BGP routing process and enters router scope configuration mode.
		• BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode.
		• Use the **global** keyword to specify that BGP uses the global routing table.
		• Use the **vrf** keyword and *vrf-name* argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.

Step 9	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:** `Router(config-router-scope)# address-family ipv4`	Enters router scope address family configuration mode to configure an address family session under BGP. • Non-topology-specific configuration parameters are configured in this configuration mode.

Step 10	**topology** {**base**| *topology-name*}
	Example:
	`Router(config-router-scope-af)# topology VIDEO`
		Configures the topology instance in which BGP will route class-specific or base topology traffic, and enters router scope address family topology configuration mode.

Step 11	**import topology** {**base**| *topology-name*}[**route-map** *map-name*]
	Example:
	`Router(config-router-scope-af-topo)# import topology VOICE route-map 10NET`
		(Optional) Configures BGP to move routes from one topology to another on the same router.
		• The **route-map** keyword can be used to filter routes that moved between topologies.

Step 12	**end**
	Example:
	`Router(config-router-scope-af-topo)# end`
		(Optional) Exits router scope address family topology configuration mode, and returns to privileged EXEC mode.

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the“Connecting to a Service Provider Using External BGP” module.
- If you want to configure some internal BGP features, see the “Configuring Internal BGP Features” chapter of the BGP section of the *Cisco IOS IP Routing Protocols Configuration Guide*.
- If you want to configure BGP neighbor session options, see the “Configuring BGP Neighbor Session Options”module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	“Cisco BGP Overview” module of the *Cisco IOS IP Routing Protocols Configuration Guide*.
Conceptual and configuration details for basic BGP tasks.	“Configuring a Basic BGP Network” module of the *Cisco IOS IP Routing Protocols Configuration Guide*.
Information about SNMP and SNMP operations.	“Configuring SNMP Support” section of the *Cisco IOS Network Management Configuration Guide*.

Standards

Standard	Title
MDT SAFI	*MDT SAFI*

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1657	*Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4724	*Graceful Restart Mechanism for BGP*

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 Advanced 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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Configuring Advanced BGP Features

Feature Name	Releases	Feature Configuration Information
BGP Graceful Restart per Neighbor	12.2(33)SRC 12.2(33)SB 15.0(1)M 15.0(1)S Cisco IOS XE 3.1.0SG	The BGP Graceful Restart per Neighbor feature enables or disables the BGP graceful restart capability for an individual BGP neighbor, including using peer session templates and BGP peer groups. In Cisco IOS Release 12.2(33)SB, platform support includes the Cisco 10000 series routers. The following commands were introduced or modified by this feature: **ha-mode graceful-restart**, **neighbor ha-mode graceful-restart**, **show ip bgp neighbors**.
BGP MIB Support Enhancements	12.0(26)S 12.2(25)S 12.3(7)T 12.2(33)SRA 12.2(33)SXH	The BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following command was introduced in this feature: **snmp-server enable traps bgp**.

Feature Name	Releases	Feature Configuration Information
BGP Nonstop Forwarding (NSF) Awareness	12.2(15)T 15.0(1)S	Nonstop Forwarding (NSF) awareness allows a router to assist NSF-capable neighbors to continue forwarding packets during a Stateful Switchover (SSO) operation. The BGP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running BGP to forward packets along routes that are already known for a router that is performing an SSO operation. This capability allows the BGP peers of the failing router to retain the routing information that is advertised by the failing router and continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation. The following commands were introduced or modified by this feature: **bgp graceful-restart**, **show ip bgp**, **show ip bgp neighbors**.
BGP Selective Address Tracking	12.4(4)T 12.2(33)SRB	The BGP Selective Address Tracking feature introduces the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The following commands were modified by this feature: **bgp nexthop**, **neighbor fall-over**.

Feature Name	Releases	Feature Configuration Information
BGP Support for BFD	12.0(31)S 12.4(4)T 12.2(33)SRA 12.2(33)SXH 12.2(33)SB 15.0(1)S	Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster reconvergence time. The following commands were introduced or modified by this feature: **bfd**, **neighbor fall-over**, **show bfd neighbors**, **show ip bgp neighbors**.

Feature Name	Releases	Feature Configuration Information
BGP Support for MTR	12.2(33)SRB	BGP support for MTR introduces a new configuration hierarchy and command-line interface (CLI) commands to support multi-topology routing (MTR) topologies. The new configuration hierarchy, or scope, can be implemented by BGP independently of MTR. MTR allows the configuration of service differentiation through class-based forwarding. MTR supports multiple unicast topologies and a separate multicast topology. A topology is a subset of the underlying network (or base topology) characterized by an independent set of Network Layer Reachability Information (NLRI). In 12.2(33)SRB, this feature was introduced on the Cisco 7600. The following commands were introduced or modified by this feature: **address-family ipv4 (BGP)**, **bgp tid**, **clear ip bgp topology**, **import topology**, **neighbor translate-topology**, **neighbor transport**, **scope**, **show ip bgp topology**, **topology (BGP).**

Feature Name	Releases	Feature Configuration Information
BGP Support for Next-Hop Address Tracking	12.0(29)S 12.3(14)T 12.2(33)SXH 15.0(1)S	The BGP Support for Next-Hop Address Tracking feature is enabled by default when a supporting Cisco IOS software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a bestpath calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed. The following command was introduced in this feature: **bgp nexthop**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.