IP Routing: BGP Features Bookmap Cisco IOS XE Release 3.3SG

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

BGP Support for 4-byte ASN

The Cisco implementation of 4-byte autonomous system numbers uses asplain (65538, for example) as the default regular expression match and the output display format for AS numbers. However, you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396.

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.

Information About BGP Support for 4-byte ASN

Cisco Implementation of 4-Byte Autonomous System Numbers

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 15.1(1)SG, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain (65538, for example) as the default regular expression match and the output display format for AS numbers. However, you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396.

To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the **bgp asnotation dot** command followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions.

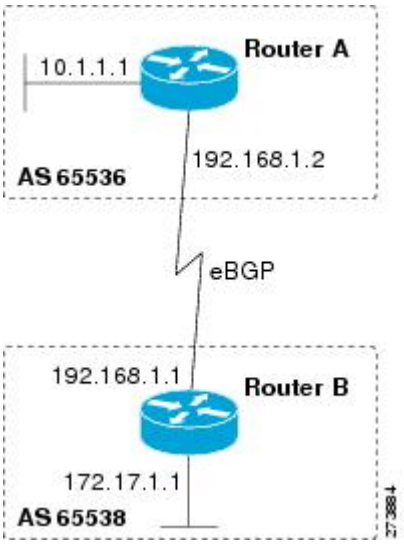
In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses asdot (1.2, for example) as the only configuration format, regular expression match, and output display, with no asplain support.

For an example of BGP peers in two autonomous systems using 4-byte numbers, see the figure below. To view a configuration example of the configuration between three neighbor peers in separate 4-byte autonomous systems configured using asdot notation, see the Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers.

Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number be upgraded to support 4-byte autonomous system numbers.

Note A new private autonomous system number, 23456, was created by RFC 4893, and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

Figure 1 BGP Peers in Two Autonomous Systems Using 4-Byte Numbers



How to Configure BGP Support for 4-byte ASN

Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Perform this task to configure a BGP routing process and BGP peers when the BGP peers are located in an AS that uses 4-byte autonomous system numbers. The address family configured here is the default IPv4 unicast address family, and the configuration is done at Router B in the figure above (in the "Cisco Implementation of 4-Byte Autonomous System Numbers" section). The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

This task requires Cisco IOS Release 12.0(32)SY8, 12.2(33)SXI1, or a later release to be running on the router.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Repeat Step 4 to define other BGP neighbors, as required.
6. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
7. **neighbor** *ip-address* **activate**
8. Repeat Step 7 to activate other BGP neighbors, as required.
9. **network** *network-number* [**mask** *network-mask*][**route-map** *route-map-name*]
10. **end**
11. **show ip bgp** [*network*] [*network-mask*]
12. **show ip bgp summary**

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 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 65536`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the 4-byte autonomous system number, 65536, is defined in asplain notation.
Step 5	Repeat Step 4 to define other BGP neighbors, as required.	--
Step 6	**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 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 virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 8	Repeat Step 7 to activate other BGP neighbors, as required.	--

	Command or Action	Purpose
Step 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 10	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp 10.1.1.0`	(Optional) Displays the entries in the BGP routing table. **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 12	**show ip bgp summary** **Example:** `Router# show ip bgp summary`	(Optional) Displays the status of all BGP connections.

Examples

The following output from the **show ip bgp** command at Router B shows the BGP routing table entry for network 10.1.1.0 learned from the BGP neighbor at 192.168.1.2 in Router A in the figure above with its 4-byte autonomous system number of 65536 displayed in the default asplain format.

```
RouterB# show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
  Advertised to update-groups:
     2
  65536
    192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The following output from the **show ip bgp summary** command shows the 4-byte autonomous system number 65536 for the BGP neighbor 192.168.1.2 of Router A in the figure above after this task has been configured on Router B:

```
RouterB# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
```

```
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  Stated
192.168.1.2     4       65536       6       6        3    0    0 00:01:33       1
```

Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers

Perform this task to modify the default output format for 4-byte autonomous system numbers from asplain format to asdot notation format. The **show ip bgp summary** command is used to display the changes in output format for the 4-byte autonomous system numbers.

This example requires Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, or a later release, to be running on the router.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **bgp asnotation dot**
6. **end**
7. **clear ip bgp ***
8. **show ip bgp summary**
9. **show ip bgp regexp** *regexp*
10. **configure terminal**
11. **router bgp** *autonomous-system-number*
12. **no bgp asnotation dot**
13. **end**
14. **clear ip bgp ***

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp summary** **Example:** `Router# show ip bgp summary`	Displays the status of all BGP connections.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.
Step 5	**bgp asnotation dot** **Example:** `Router(config-router)# bgp asnotation dot`	Changes the default output format of BGP 4-byte autonomous system numbers from asplain (decimal values) to dot notation. **Note** 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	**clear ip bgp *** **Example:** `Router# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte autonomous system number format change is reflected in all BGP sessions. **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 8	**show ip bgp summary** **Example:** `Router# show ip bgp summary`	Displays the status of all BGP connections.
Step 9	**show ip bgp regexp** *regexp* **Example:** `Router# show ip bgp regexp ^1\.0$`	Displays routes that match the autonomous system path regular expression. • In this example, a regular expression to match a 4-byte autonomous system path is configured using asdot format.

Command or Action	Purpose
Step 10 **configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 11 **router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte autonomous system number, 65538, is defined in asplain notation.
Step 12 **no bgp asnotation dot** **Example:** `Router(config-router)# no bgp asnotation dot`	Resets the default output format of BGP 4-byte autonomous system numbers back to asplain (decimal values). **Note** 4-byte autonomous system numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 13 **end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 14 **clear ip bgp *** **Example:** `Router# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte autonomous system number format change is reflected in all BGP sessions. **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*.

Examples

The following output from the **show ip bgp summary** command shows the default asplain format of the 4-byte autonomous system numbers. Note the asplain format of the 4-byte autonomous system numbers, 65536 and 65550.

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4        65536       7       7        1    0    0 00:03:04      0
192.168.3.2     4        65550       4       4        1    0    0 00:00:15      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the output is converted to asdot notation format as shown in the following output from the **show ip bgp summary** command. Note the asdot format of the 4-byte

autonomous system numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 autonomous system numbers.

```
Router# show ip bgp summary
BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4          1.0       9       9        1    0    0 00:04:13      0
192.168.3.2     4         1.14       6       6        1    0    0 00:01:24      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the regular expression match format for 4-byte autonomous system paths is changed to asdot notation format. Although a 4-byte autonomous system number can be configured in a regular expression using either asplain format or asdot format, only 4-byte autonomous system numbers configured using the current default format are matched. In the first example below, the **show ip bgp regexp** command is configured with a 4-byte autonomous system number in asplain format. The match fails because the default format is currently asdot format and there is no output. In the second example using asdot format, the match passes and the information about the 4-byte autonomous system path is shown using the asdot notation.

Note The asdot notation uses a period, which is a special character in Cisco regular expressions. To remove the special meaning, use a backslash before the period.

```
Router# show ip bgp regexp ^65536$
Router# show ip bgp regexp ^1\.0$
BGP table version is 2, 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 1.0 i
```

Configuration Examples for BGP Support for 4-byte ASN

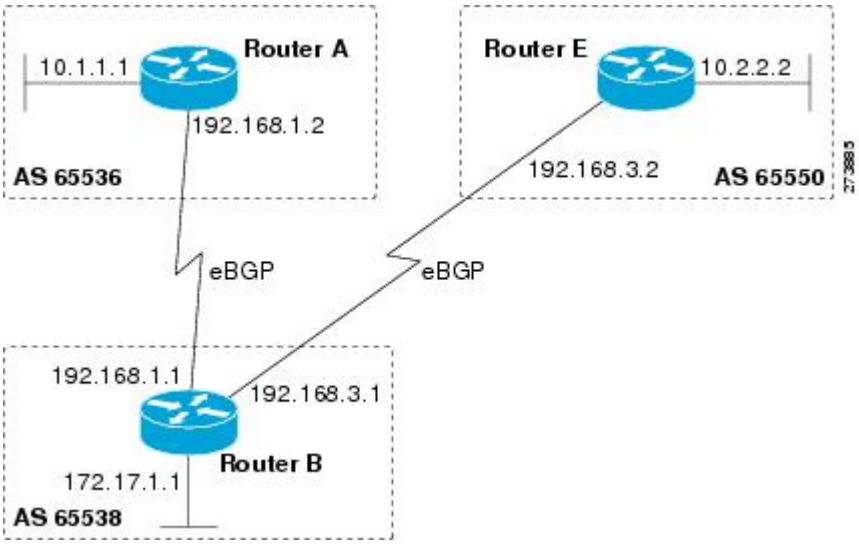
Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, 15.1(1)SG, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 15.1(1)SG, and later releases and shows the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers

(at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using asplain notation. IPv4 unicast routes are exchanged with all peers.

Figure 2 BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format



Router A

```
router bgp 65536
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 65538
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 65536
 neighbor 192.168.3.2 remote-as 65550
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

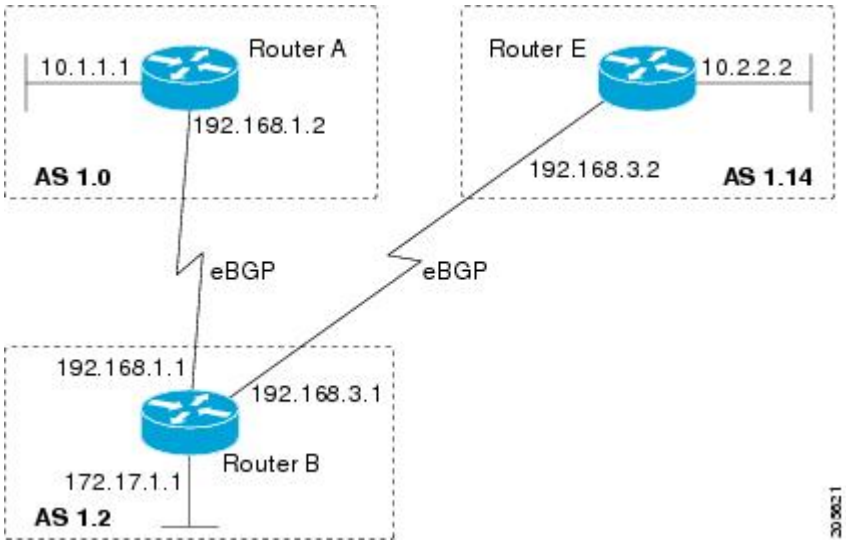
Router E

```
router bgp 65550
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T and shows how to create the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

Figure 3 BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format



Router A

```
router bgp 1.0
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
```

```
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 1.2
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 1.0
 neighbor 192.168.3.2 remote-as 1.14
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

Router E

```
router bgp 1.14
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases and shows how to create a VRF with a route-target that uses a 4-byte autonomous system number, 65537, and how to set the route target to extended community value 65537:100 for routes that are permitted by the route map.

```
ip vrf vpn_red
 rd 64500:100
 route-target both 65537:100
 exit
route-map red_map permit 10
 set extcommunity rt 65537:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 65537.

```
RouterB# show route-map red_map
```

```
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:65537:100
  Policy routing matches: 0 packets, 0 bytes
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T and shows how to create a VRF with a route-target that uses a 4-byte autonomous system number, 1.1, and how to set the route target to extended community value 1.1:100 for routes that are permitted by the route map.

Note In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXI1, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
ip vrf vpn_red
 rd 64500:100
 route-target both 1.1:100
 exit
route-map red_map permit 10
 set extcommunity rt 1.1:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 1.1.

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:1.1:100
  Policy routing matches: 0 packets, 0 bytes
```

Additional References

Related Documents

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

Standards and RFCs

Standard/RFC	Title
RFC 4893	*BGP Support for Four-octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous System (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

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 BGP Support for 4-byte ASN

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 BGP Support for 4-byte ASN

Feature Name	Releases	Feature Information
BGP Support for 4-byte ASN	12.0(32)S12 12.0(32)SY8 12.0(33)S3 12.2(33)SRE 12.2(33)XNE 12.2(33)SXI1 12.4(24)T 15.0(1)S Cisco IOS XE Release 3.1.0SG 15.1(1)SG Cisco IOS XE Release 3.3SG	The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 15.1(1)SG, Cisco IOS XE Release 3.3SG, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the **bgp asnotation dot** command. In Cisco IOS Releases 12.0(32)S12 and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses asdot as the only configuration format, regular expression match, and output display, with no asplain support. The following commands were introduced or modified by this feature: **bgp asnotation dot**, **bgp confederation identifier**, **bgp confederation peers**, all **clear ip bgp** commands that configure an autonomous system number, **ip as-path access-list**, **ip extcommunity-list**, **match source-protocol**, **neighbor local-as**, **neighbor remote-as**, **neighbor soo**, **redistribute (**IP **)**, **router bgp**, **route-target**, **set as-**

Feature Name	Releases	Feature Information
		path, **set extcommunity**, **set origin**, **soo**, all **show ip bgp** commands that display an autonomous system number, and **show ip extcommunity-list**.

BGP Support for Dual AS Configuration for Network AS Migrations

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.

Information About BGP Support for Dual AS Configuration for Network AS Migrations

BGP Network Autonomous System Migration

Autonomous System Migration for BGP Networks

Autonomous system migration can be necessary when a telecommunications or Internet service provider purchases another network. It is desirable for the provider to be able to integrate the second autonomous system without disrupting existing customer peering arrangements. The amount of configuration required in the customer networks can make this a cumbersome task that is difficult to complete without disrupting service.

Dual Autonomous System Support for BGP Network Autonomous System Migration

In Cisco IOS Release 12.0(29)S, 12.3(14)T, 12.2(33)SXH, and later releases, support was added for dual BGP autonomous system configuration to allow a secondary autonomous system to merge under a primary autonomous system, without disrupting customer peering sessions. The configuration of this feature is transparent to customer networks. Dual BGP autonomous system configuration allows a router to appear, to external peers, as a member of secondary autonomous system during the autonomous system migration. This feature allows the network operator to merge the autonomous systems and then later migrate customers to new configurations during normal service windows without disrupting existing peering arrangements.

The **neighbor local-as** command is used to customize the AS_PATH attribute by adding and removing autonomous system numbers for routes received from eBGP neighbors. This feature allows a router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. This feature simplifies this process of changing the autonomous system number in a BGP network by allowing the network operator to merge a secondary autonomous system into a primary autonomous system and then later update the customer configurations during normal service windows without disrupting existing peering arrangements.

BGP Autonomous System Migration Support for Confederations, Individual Peering Sessions, and Peer Groupings

This feature supports confederations, individual peering sessions, and configurations applied through peer groups and peer templates. If this feature is applied to a group peers, the individual peers cannot be customized.

Ingress Filtering During BGP Autonomous System Migration

Autonomous system path customization increases the possibility that routing loops can be created if such customization is misconfigured. The larger the number of customer peerings, the greater the risk. You can minimize this possibility by applying policies on the ingress interfaces to block the autonomous system number that is in transition or routes that have no **local-as** configuration.

Caution BGP prepends the autonomous system number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This feature should be configured only for autonomous system migration and should be deconfigured after the transition has been completed. This procedure should be attempted only by an experienced network operator, as routing loops can be created with improper configuration.

BGP Network Migration to 4-Byte Autonomous System Numbers

The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA will start to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.

The Cisco implementation of 4-byte autonomous system numbers supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

Migrating your BGP network to 4-byte autonomous system numbers requires some planning. If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number be upgraded to support 4-byte autonomous system numbers.

For details about steps to perform to upgrade a BGP network to full 4-byte autonomous system support, see the Migration Guide for Explaining 4-Byte Autonomous System white paper.

How to Configure Dual AS Peering for Network Migration

- Configuring Dual AS Peering for Network Migration, page 19

Configuring Dual AS Peering for Network Migration

Perform this task to configure a BGP peer router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. When the BGP peer is configured with dual autonomous system numbers then the network operator can merge a secondary autonomous system into a primary autonomous system and update the customer configuration during a future service window without disrupting existing peering arrangements.

The **show ip bgp** and **show ip bgp neighbors** commands can be used to verify autonomous system number for entries in the routing table and the status of this feature.

Note

- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for only true eBGP peering sessions. This feature cannot be configured for two peers in different subautonomous systems of a confederation.
- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for individual peering sessions and configurations applied through peer groups and peer templates. If this command is applied to a peer group, the peers cannot be individually customized.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **local-as** [*autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]]
6. **neighbor** *ip-address* **remove-private-as**
7. **end**
8. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter-prefixes** *mask-length*]
9. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes**| **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*]

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 40000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 45000`	Establishes a peering session with a BGP neighbor.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **local-as** [*autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]] **Example:** `Router(config-router)# neighbor 10.0.0.1 local-as 50000 no-prepend replace-as dual-as`	Customizes the AS_PATH attribute for routes received from an eBGP neighbor. • The **replace-as** keyword is used to prepend only the local autonomous system number (as configured with the *ip-address* argument) to the AS_PATH attribute. The autonomous system number from the local BGP routing process is not prepended. • The **dual-as** keyword is used to configure the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous system number configured with the *ip-address* argument (local-as). • The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the local-as number.
Step 6	**neighbor** *ip-address* **remove-private-as** **Example:** `Router(config-router)# neighbor 10.0.0.1 remove-private-as`	(Optional) Removes private autonomous system numbers from outbound routing updates. • This command can be used with the **replace-as** functionality to remove the private autonomous system number and replace it with an external autonomous system number. • Private autonomous system numbers (64512 to 65535) are automatically removed from the AS_PATH attribute when this command is configured.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits configuration mode and enters privileged EXEC mode.
Step 8	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter-prefixes** *mask-length*] **Example:** `Router# show ip bgp`	Displays entries in the BGP routing table. • The output can be used to verify if the real autonomous system number or local-as number is configured.

	Command or Action	Purpose
Step 9	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes**| **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*] **Example:** `Router# show ip bgp neighbors`	Displays information about TCP and BGP connections to neighbors. • The output will display **local AS**, **no-prepend**, **replace-as**, and **dual-as** with the corresponding autonomous system number when these options are configured.

Configuration Examples for Dual AS Peering for Network Migration

Example: Dual AS Configuration

The following examples shows how this feature is used to merge two autonomous systems without interrupting peering arrangements with the customer network. The **neighbor local-as** command is configured to allow Router 1 to maintain peering sessions through autonomous system 40000 and autonomous system 45000. Router 2 is a customer router that runs a BGP routing process in autonomous system 50000 and is configured to peer with autonomous-system 45000.

Router 1 in Autonomous System 40000 (Provider Network)

```
interface Serial3/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 40000
 no synchronization
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
 neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```

Router 1 in Autonomous System 45000 (Provider Network)

```
interface Serial3/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 45000
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
```

Router 2 in Autonomous System 50000 (Customer Network)

```
interface Serial3/0
```

```
 ip address 10.3.3.33 255.255.255.0
!
router bgp 50000
 bgp router-id 10.0.0.3
 neighbor 10.3.3.11 remote-as 45000
```

After the transition is complete, the configuration on router 50000 can be updated to peer with autonomous system 40000 during a normal maintenance window or during other scheduled downtime:

```
 neighbor 10.3.3.11 remote-as 100
```

Example: Dual AS Confederation Configuration

The following example can be used in place of the Router 1 configuration in the "Example: Dual AS Configuration" example. The only difference between these configurations is that Router 1 is configured to be part of a confederation.

```
interface Serial3/0/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 65534
 no synchronization
 bgp confederation identifier 100
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
 neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```

Example: Replace an AS with Another AS in Routing Updates

The following example strips private autonomous system 64512 from outbound routing updates for the 10.3.3.33 neighbor and replaces it with autonomous system 50000:

```
router bgp 64512
 neighbor 10.3.3.33 local-as 50000 no-prepend replace-as
```

Additional References

Related Documents

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

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 BGP Support for Dual AS Configuration for Network AS Migrations

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 2 Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations

Feature Name	Releases	Feature Information
BGP Support for Dual AS Configuration for Network AS Migrations	12.0(27)S 12.2(25)S 12.3(11)T 12.2(33)SRA 12.2(33)SXH 15.0(1)S 15.1(1)SG Cisco IOS XE Release 3.3SG	The BGP Support for Dual AS Configuration for Network AS Migrations feature extended the functionality of the BGP Local-AS feature by providing additional autonomous system path customization configuration options. The configuration of this feature is transparent to customer peering sessions, allowing the provider to merge two autonomous systems without interrupting customer peering arrangements. Customer peering sessions can later be updated during a maintenance window or during other scheduled downtime. The following command was modified by this feature: **neighbor local-as**.

BGP Dynamic Neighbors

BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups.

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.

Information About BGP Dynamic Neighbors

BGP Dynamic Neighbors

Support for the BGP Dynamic Neighbors feature was introduced in Cisco IOS Release 12.2(33)SXH on the Cisco Catalyst 6500 series switches. BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups.

After a subnet range is configured for a BGP peer group and a TCP session is initiated by another router for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. After the initial configuration of subnet ranges and activation of the peer group (referred to as a *listen range group*), dynamic BGP neighbor creation does not require any further CLI configuration on the initial router. Other routers can establish a BGP session with the initial router, but the initial router need not

establish a BGP session to other routers if the IP address of the remote peer used for the BGP session is not within the configured range.

To support the BGP Dynamic Neighbors feature, the output for the **show ip bgp neighbors**, **show ip bgp peer-group**, and **show ip bgp summary** commands was updated to display information about dynamic neighbors.

A dynamic BGP neighbor will inherit any configuration for the peer group. In larger BGP networks, implementing BGP dynamic neighbors can reduce the amount and complexity of CLI configuration and save CPU and memory usage. Only IPv4 peering is supported.

How to Configure BGP Dynamic Neighbors

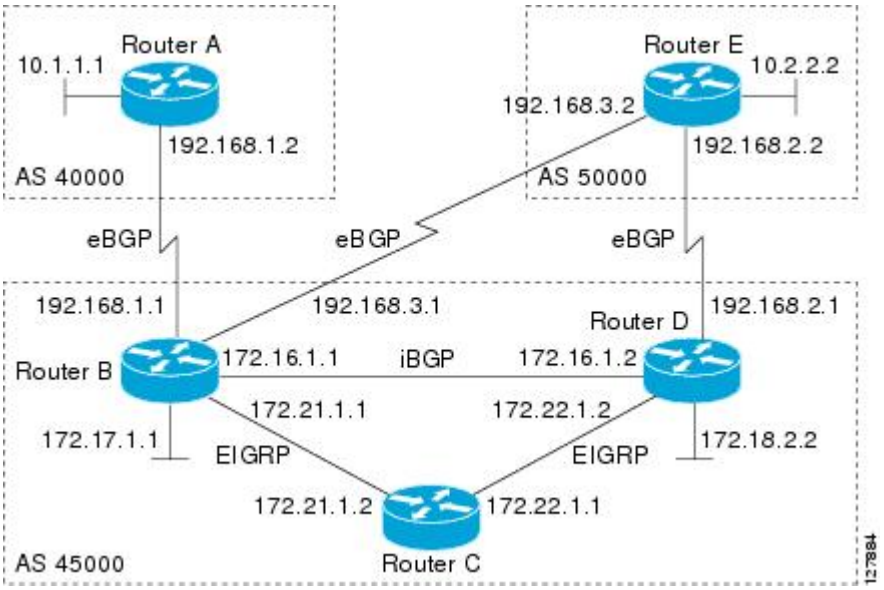
Implementing BGP Dynamic Neighbors Using Subnet Ranges

In Cisco IOS Release 12.2(33)SXH, support for BGP dynamic neighbors was introduced. Perform this task to implement the dynamic creation of BGP neighbors using subnet ranges.

In this task, a BGP peer group is created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer group is added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured. The peer group is activated under the IPv4 address family.

The next step is to move to another router--Router E in the figure below--where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.3.2) is within the configured subnet range for dynamic BGP peers. The task moves back to the first router, Router B, to run three **show** commands that have been modified to display dynamic BGP peer information.

Figure 4 BGP Dynamic Neighbor Topology



This task requires Cisco IOS Release 12.2(33)SXH, or a later release, to be running.

Note This task supports only IPv4 BGP peering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *peer-group-name* **peer-group**
6. **bgp listen** [**limit** *max-number*]
7. **bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*]
8. **neighbor** {*ip-address* **|** *ipv6-address* **|** *peer-group-name*} **ebgp-multihop [** *ttl*]
9. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
10. **address-family ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]]
11. **neighbor** {*ip-address*| *peer-group-name*} **activate**
12. **end**
13. Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.
14. **enable**
15. **configure terminal**
16. **router bgp** *autonomous-system-number*
17. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*[**alternate-as** *autonomous-system-number...*]
18. Return to the first router.
19. **show ip bgp summary**
20. **show ip bgp peer-group** [*peer-group-name*] [**summary**]
21. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `RouterB> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on router B.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `RouterB# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `RouterB(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:** `RouterB(config-router)# bgp log-neighbor-changes`	(Optional) 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 5	**neighbor** *peer-group-name* **peer-group** **Example:** `RouterB(config-router)# neighbor group192 peer-group`	Creates a BGP peer group. • In this example, a peer group named group192 is created. This group will be used as a listen range group.
Step 6	**bgp listen** [**limit** *max-number*] **Example:** `RouterB(config-router)# bgp listen limit 200`	Sets a global limit of BGP dynamic subnet range neighbors. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic subnet range neighbors that can be created. **Note** Only the syntax applicable to this task is used in this example. For the complete syntax, see Step 7.
Step 7	**bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*] **Example:** `RouterB(config-router)# bgp listen range 192.168.0.0/16 peer-group group192`	Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic neighbors that can be created. • Use the optional **range** keyword and *network* / *length* argument to define a prefix range to be associated with the specified peer group. • In this example, the prefix range 192.168.0.0/16 is associated with the listen range group named group192.

	Command or Action	Purpose
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop [** *ttl*] **Example:** `RouterB(config-router)# neighbor group192 ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 9	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:** `RouterB(config-router)# neighbor group192 remote-as 40000 alternate-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • Use the optional **alternate-as** keyword and *autonomous-system-number* argument to identify up to five alternate autonomous system numbers for listen range neighbors. • In this example, the peer group named group192 is configured with two possible autonomous system numbers. **Note** The **alternate-as** keyword is used only with the listen range peer groups, not with individual BGP neighbors.
Step 10	**address-family ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]] **Example:** `RouterB(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 11	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `RouterB(config-router-af)# neighbor group192 activate`	Activates the neighbor or listen range peer group for the configured address family. • In this example, the neighbor 172.16.1.1 is activated for the IPv4 address family. **Note** Usually BGP peer groups cannot be activated using this command, but the listen range peer groups are a special case.
Step 12	**end** **Example:** `RouterB(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 13	Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.	--

	Command or Action	Purpose
Step 14	**enable** **Example:** `RouterE> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on Router E.
Step 15	**configure terminal** **Example:** `RouterE# configure terminal`	Enters global configuration mode.
Step 16	**router bgp** *autonomous-system-number* **Example:** `RouterE(config)# router bgp 50000`	Enters router configuration mode for the specified routing process.
Step 17	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*[**alternate-as** *autonomous-system-number*...] **Example:** `RouterE(config-router)# neighbor 192.168.3.1 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the interface (192.168.3.2 in the figure above) at Router E is with the subnet range set for the BGP listen range group, group192. When TCP opens a session to peer to Router B, Router B creates this peer dynamically.
Step 18	Return to the first router.	--
Step 19	**show ip bgp summary** **Example:** `RouterB# show ip bgp summary`	(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors. • In this step, the configuration has returned to Router B.
Step 20	**show ip bgp peer-group** [*peer-group-name*] [**summary**] **Example:** `RouterB# show ip bgp peer-group group192`	(Optional) Displays information about BGP peer groups.

Command or Action	Purpose
Step 21 **show ip bgp neighbors** [*ip-address*] **Example:** `RouterB# show ip bgp neighbors 192.168.3.2`	(Optional) Displays information about BGP and TCP connections to neighbors. • In this example, information is displayed about the dynamically created neighbor at 192.168.3.2. The IP address of this BGP neighbor can be found in the output of either the **show ip bgp summary** or the **show ip bgp peer-group** command. **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*.

Examples

The following output examples were taken from Router B in the figure above after the appropriate configuration steps in this task were completed on both Router B and Router E.

The following output from the **show ip bgp summary** command shows that the BGP neighbor 192.168.3.2 was dynamically created and is a member of the listen range group, group192. The output also shows that the IP prefix range of 192.168.0.0/16 is defined for the listen range named group192.

```
Router# show ip bgp 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
*192.168.3.2    4 50000       2       2        0    0    0 00:00:37        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1/(200 max), Subnet ranges: 1
BGP peergroup group192 listen range group members:
  192.168.0.0/16
```

The following output from the **show ip bgp peer-group** command shows information about the listen range group, group192 that was configured in this task:

```
Router# show ip bgp peer-group group192
BGP peer-group is group192,  remote AS 40000
  BGP peergroup group192 listen range group members:
  192.168.0.0/16
  BGP version 4
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP neighbor is group192, peer-group external, members:
  *192.168.3.2
  Index 0, Offset 0, Mask 0x0
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
```

The following sample output from the **show ip bgp neighbors** command shows that the neighbor 192.168.3.2 is a member of the peer group, group192, and belongs to the subnet range group 192.168.0.0/16, which shows that this peer was dynamically created:

```
Router# show ip bgp neighbors 192.168.3.2
BGP neighbor is *192.168.3.2,  remote AS 50000, external link
 Member of peer-group group192 for session parameters
 Belongs to the subnet range group: 192.168.0.0/16
  BGP version 4, remote router ID 192.168.3.2
  BGP state = Established, up for 00:06:35
  Last read 00:00:33, last write 00:00:25, hold time is 180, keepalive intervals
  Neighbor capabilities:
    Route refresh: advertised and received(new)
```

```
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             7          7
    Route Refresh:          0          0
    Total:                  8          8
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
  group192 peer-group member
.
.
.
```

Configuration Examples for BGP Dynamic Neighbors

Example Implementing BGP Dynamic Neighbors Using Subnet Ranges

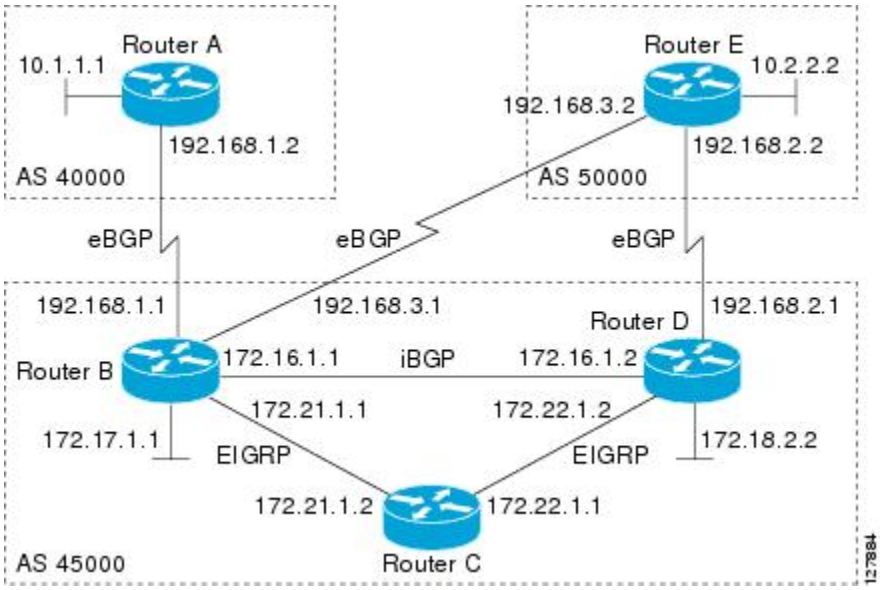
In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured for one of the peer groups, group192. The subnet range peer groups and a standard BGP peer are then activated under the IPv4 address family.

The configuration moves to another router--Router A in the figure below--where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.1.2) is within the configured subnet range for dynamic BGP peers.

A third router--Router E in the figure below--also starts a BGP peering session with Router B. Router E is in the autonomous system 50000, which is the configured alternate autonomous system. Router B responds to the resulting TCP session by creating another dynamic BGP peer.

This example concludes with the output of the **show ip bgp summary** command entered on Router B.

Figure 5 BGP Dynamic Neighbor Topology



Router B

```
enable
configure terminal
router bgp 45000
 bgp log-neighbor-changes
 bgp listen limit 200
 bgp listen range 172.21.0.0/16 peer-group group172
 bgp listen range 192.168.0.0/16 peer-group group192
 neighbor group172 peer-group
 neighbor group172 remote-as 45000
 neighbor group192 peer-group
 neighbor group192 remote-as 40000 alternate-as 50000
 neighbor 172.16.1.2 remote-as 45000
 address-family ipv4 unicast
 neighbor group172 activate
 neighbor group192 activate
 neighbor 172.16.1.2 activate
 end
```

Router A

```
enable
configure terminal
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
 exit
```

Router E

```
enable
configure terminal
router bgp 50000
 neighbor 192.168.3.1 remote-as 45000
 exit
```

After both Router A and Router E are configured, the **show ip bgp summary** command is run on Router B. The output displays the regular BGP neighbor, 172.16.1.2, and the two BGP neighbors that were created dynamically when Router A and Router E initiated TCP sessions for BGP peering to Router B. The output also shows information about the configured listen range subnet groups.

```
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      15      15        1    0    0 00:12:20        0
*192.168.1.2    4 40000       3       3        1    0    0 00:00:37        0
*192.168.3.2    4 50000       6       6        1    0    0 00:04:36        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 2/(200 max), Subnet ranges: 2
BGP peergroup group172 listen range group members:
  172.21.0.0/16
BGP peergroup group192 listen range group members:
  192.168.0.0/16
```

Additional References

Related Documents

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

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 BGP Dynamic Neighbors

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 3 Feature Information for BGP Dynamic Neighbors

Feature Name	Releases	Feature Information
BGP Dynamic Neighbors	12.2(33)SXH 15.1(2)T 15.0(1)S 15.1(1)SG Cisco IOS XE Release 3.3SG	BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups. After a subnet range is configured for a BGP peer group and a TCP session is initiated for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. The new BGP neighbor will inherit any configuration for the peer group. The following commands were introduced or modified by this feature: **bgp listen**, **debug ip bgp range**, **neighbor remote-as**, **show ip bgp neighbors**, **show ip bgp peer-group**, and **show ip bgp summary**.

BGP Graceful Restart per Neighbor

The BGP graceful restart feature is already available on a global basis. The BGP Graceful Restart per Neighbor feature allows BGP graceful restart to be enabled or disable for an individual neighbor, providing greater network flexibility and service.

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.

Information About BGP Graceful Restart per Neighbor

BGP Graceful Restart per Neighbor

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.

How to Configure BGP Graceful Restart per Neighbor

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.

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.

	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	**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.

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

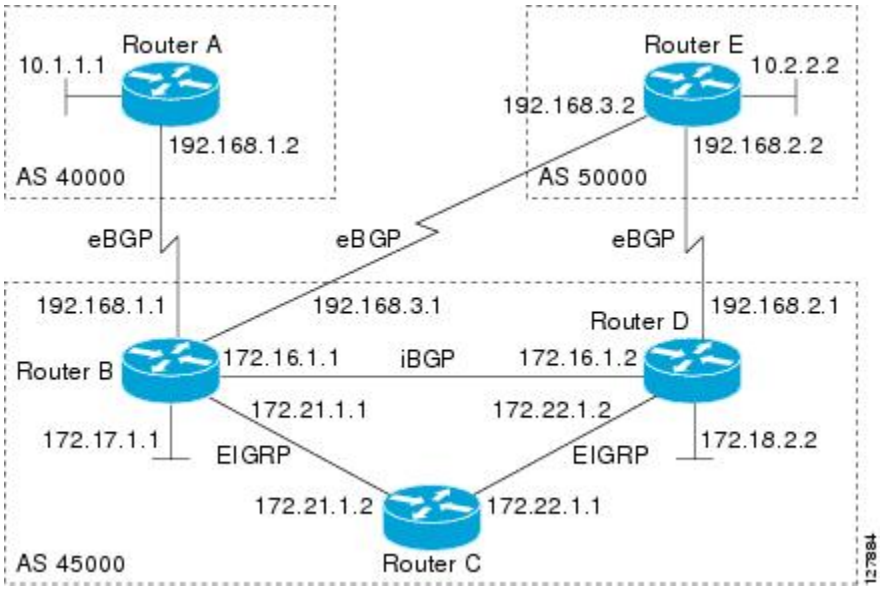
Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

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 6 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 figure. 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.

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 be set only by using the **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
```

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.

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.

	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** *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.

	Command or Action	Purpose
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
```

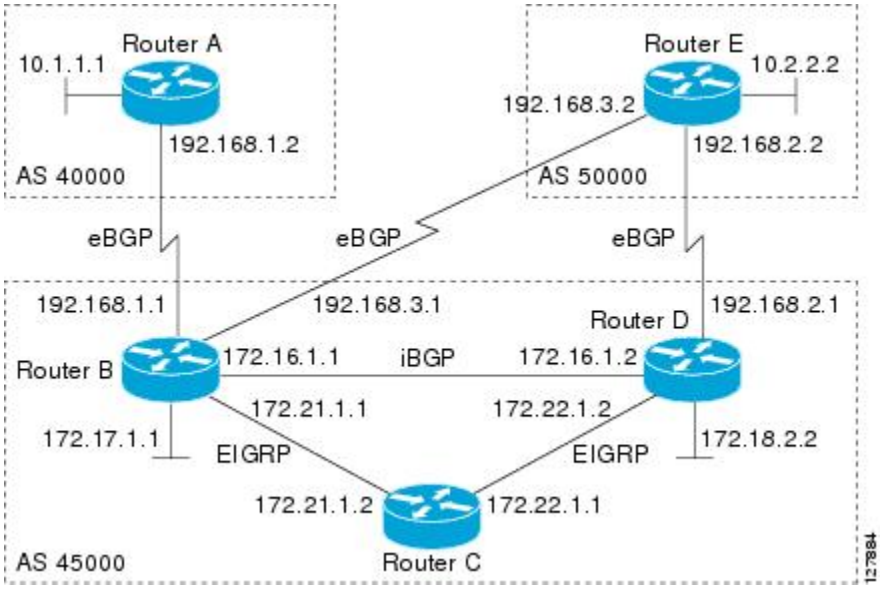
Configuration Examples for BGP Graceful Restart per Neighbor

• Examples: Enabling and Disabling BGP Graceful Restart per Neighbor, page 51

Examples: Enabling and Disabling BGP Graceful Restart per Neighbor

The ability to enable or disable the BGP graceful restart capability for an individual BGP neighbor, peer group, or peer session template was introduced. The following example is configured on Router B in the figure below and enables the BGP graceful restart capability for the BGP peer session template named S1 and disables the BGP graceful restart capability for the BGP peer session template named S2. The external BGP neighbor at Router A in the figure below (192.168.1.2) inherits peer session template S1, and the BGP graceful restart capability is enabled for this neighbor. Another external BGP neighbor at Router E in the figure below (192.168.3.2) is configured with the BGP graceful restart capability disabled after inheriting peer session template S2.

Figure 7 Network Topology Showing BGP Neighbors for BGP Graceful Restart



The BGP graceful restart capability is enabled for an individual internal BGP neighbor, 172.21.1.2 at Router C in the figure above, whereas the BGP graceful restart is disabled for the BGP neighbor 172.16.1.2 at Router D in the figure above because it is a member of the peer group PG1. The disabling of BGP graceful restart is configured for all members of the peer group, PG1. The restart and stale-path timers are modified and the BGP sessions are reset.

```
router bgp 45000
 template peer-session S1
 remote-as 40000
 ha-mode graceful-restart
 exit-peer-session
 template peer-session S2
 remote-as 50000
 ha-mode graceful-restart disable
 exit-peer-session
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 150
 bgp graceful-restart stalepath-time 400
 address-family ipv4 unicast
 neighbor PG1 peer-group
 neighbor PG1 remote-as 45000
 neighbor PG1 ha-mode graceful-restart disable
 neighbor 172.16.1.2 peer-group PG1
 neighbor 172.21.1.2 remote-as 45000
```

```
 neighbor 172.21.1.2 activate
 neighbor 172.21.1.2 ha-mode graceful-restart
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 inherit peer-session S1
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 inherit peer-session S2
 end
clear ip bgp *
```

To demonstrate how the last configuration instance of the BGP graceful restart capability is applied, the following example initially enables the BGP graceful restart capability globally for all BGP neighbors. A BGP peer group, PG2, is configured with the BGP graceful restart capability disabled. An individual external BGP neighbor, 192.168.1.2 at Router A in the figure above, is then configured to be a member of the peer group, PG2. The last graceful restart configuration instance is applied, and, in this case, the neighbor, 192.168.1.2, inherits the configuration instance from the peer group PG2 and the BGP graceful restart capability is disabled for this neighbor.

```
router bgp 45000
 bgp log-neighbor-changes
 bgp graceful-restart
 address-family ipv4 unicast
 neighbor PG2 peer-group
 neighbor PG2 remote-as 40000
 neighbor PG2 ha-mode graceful-restart disable
 neighbor 192.168.1.2 peer-group PG2
 end
clear ip bgp *
```

Additional References

Related Documents

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

Standards and RFCs

Standard/RFC	Title
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 BGP Graceful Restart per Neighbor

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 4 Feature Information for BGP Graceful Restart per Neighbor

Feature Name	Releases	Feature 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 3.1SG 15.1(1)SG Cisco IOS XE 3.3SG	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 by this feature: **ha-mode graceful-restart**, and**neighbor ha-mode graceful-restart**. The following command was modified by this feature: **show ip bgp neighbors**.

BGP Support for Next-Hop Address Tracking

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.

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.

Information About BGP Support for Next-Hop Address Tracking

BGP Support for Next-Hop Address Tracking

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

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.

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.

How to Configure BGP Support for Next-Hop Address Tracking

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.”

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.
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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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.

Configuration Examples for BGP Support for Next-Hop Address Tracking

Example: Enabling and Disabling BGP Next-Hop Address Tracking

In the following example, next-hop address tracking is disabled under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
 no bgp nexthop trigger enable
```

Example: Adjusting the Delay Interval for BGP Next-Hop Address Tracking

In the following example, the delay interval for next-hop tracking is configured to occur every 20 seconds under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
 bgp nexthop trigger delay 20
```

Additional References

Related Documents

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

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 BGP Support for Next-Hop Address Tracking

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 5 Feature Information for BGP Support for Next-Hop Address Tracking

Feature Name	Releases	Feature Information
BGP Support for Next-Hop Address Tracking	12.0(29)S 12.3(14)T 12.2(33)SXH 15.0(1)S 15.1(1)SG Cisco IOS XE Release 3.3SG	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**.

BGP Support for the L2VPN Address Family

BGP support for the Layer 2 Virtual Private Network (L2VPN) address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

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 BGP Support for the L2VPN Address Family

The BGP Support for L2VPN Address Family feature assumes prior knowledge of Virtual Private Network (VPN), Virtual Private LAN Service (VPLS), and Multiprotocol Layer Switching (MPLS) technologies.

Restrictions for BGP Support for the L2VPN Address Family

- For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.
- BGP multipaths and confederations are not supported under the L2VPN address family.

Information About BGP Support for the L2VPN Address Family

L2VPN Address Family

In Cisco IOS XE Release 2.6 and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or Generic Routing Encapsulation (GRE). The L2VPN address family is configured under BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS Virtual Private LAN Service (VPLS) feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the "VPLS Autodiscovery: BGP Based" feature.

In L2VPN address family, the following BGP commands are supported:

- **bgp nexthop**
- **bgp scan-time**
- **neighbor activate**
- **neighbor advertisement-interval**
- **neighbor allowas-in**
- **neighbor capability**
- **neighbor inherit**
- **neighbor maximum-prefix**
- **neighbor next-hop-self**
- **neighbor next-hop-unchanged**
- **neighbor peer-group**
- **neighbor remove-private-as**
- **neighbor route-map**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor soft-reconfiguration**

- **neighbor soo**
- **neighbor weight**

Note For route reflectors using L2VPNs, the **neighbor next-hop-self** and **neighbor next-hop-unchanged**commands are not supported.

For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

VPLS ID

A VPLS ID is a BGP extended community value that identifies the VPLS domain. Manual configuration of this ID is optional because a default VPLS ID is generated using the BGP autonomous system number and the configured VPN ID. A VPLS ID can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter a VPLS ID in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

45000:3

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

192.168.10.15:1

How to Configure BGP Support for the L2VPN Address Family

- Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family, page 65

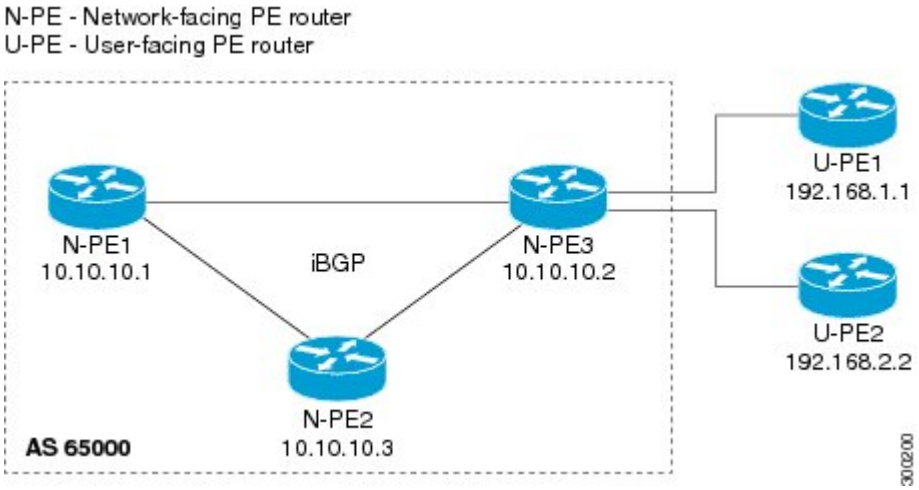
Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family

Perform this task to implement VPLS autodiscovery of each provider edge (PE) router that is a member of a specific VPLS. In Cisco IOS XE Release 2.6, the BGP L2VPN address family was introduced with a separate L2VPN RIB that contains endpoint provisioning information. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time any Layer 2 (L2) virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

BGP-based VPLS autodiscovery eliminates the need to manually provision a VPLS neighbor. After a PE router configures itself to be a member of a particular VPLS, information needed to set up connections to remote routers in the same VPLS is distributed by a discovery process. When the discovery process is complete, each member of the VPLS will have the information needed to set up VPLS pseudowires to form the full mesh of pseudowires needed for the VPLS.

This task is configured at router N-PE3 in the figure below and must be repeated at routers N-PE1 and N-PE2 with the appropriate changes such as different IP addresses. For a full configuration of these routers, see the figure below.

Figure 8 Network Diagram for BGP Autodiscovery Using the L2VPN Address Family



In this task, the PE router N-PE3 in the figure above is configured with a Layer 2 router ID, a VPN ID, a VPLS ID, and is enabled to automatically discover other PE routers that are part of the same VPLS domain. A BGP session is created to activate BGP neighbors under the L2VPN address family. Finally, two optional **show** commands are entered to verify the steps in the task.

This task assumes that MPLS is configured with VPLS options. For more details, see the "VPLS Autodiscovery: BGP Based" feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 router-id** *ip-address*
4. **l2 vfi** *vfi-name* **autodiscovery**
5. **vpn id** *vpn-id*
6. **vpls-id** *vpls-id*
7. **exit**
8. Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.
9. **router bgp** *autonomous-system-number*
10. **no bgp default ipv4-unicast**
11. **bgp log-neighbor-changes**
12. **bgp update-delay** *seconds*
13. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
14. **neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number*
15. Repeat Step 13 and Step 14 to configure other BGP neighbors.
16. **address-family l2vpn** [**vpls**]
17. **neighbor** *ip-address* **activate**
18. **neighbor** {*ip-address*| *peer-group-name*} **send-community**[**both**| **standard**| **extended**]
19. Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.
20. **end**
21. **show vfi**
22. **show ip bgp l2vpn vpls** {**all** | **rd** *vpn-rd*}

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	**l2 router-id** *ip-address* **Example:** `Router(config)# l2 router-id 10.1.1.3`	Specifies a router ID (in IP address format) for the PE router to use with VPLS autodiscovery pseudowires. • In this example, the L2 router ID is defined as 10.1.1.3.
Step 4	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Router(config)# l2 vfi customerA autodiscovery`	Creates an L2 VFI, enables the VPLS PE router to automatically discover other PE routers that are part of the same VPLS domain, and enters L2 VFI autodiscovery configuration mode. • In this example, the L2 VFI named customerA is created.
Step 5	**vpn id** *vpn-id* **Example:** `Router(config-vfi)# vpn id 100`	Specifies a VPN ID. • Use the same VPN ID for the PE routers that belong to the same VPN. Make sure that the VPN ID is unique for each VPN in the service provider network. • Use the *vpn-id* argument to specify a number in the range from 1 to 4294967295. • In this example, a VPN ID of 100 is specified.
Step 6	**vpls-id** *vpls-id* **Example:** `Router(config-vfi)# vpls-id 65000:100`	(Optional) Specifies a VPLS ID. • The VPLS ID is an identifier that is used to identify the VPLS domain. This command is optional because a default VPLS ID is automatically generated using the BGP autonomous system number and the VPN ID configured for the VFI. Only one VPLS ID can be configured per VFI, and the same VPLS ID cannot be configured in multiple VFIs on the same router. • In this example, a VPLS ID of 65000:100 is specified.
Step 7	**exit** **Example:** `Router(config-vfi)# exit`	Exits L2 VFI autodiscovery configuration mode and returns to global configuration mode.
Step 8	Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.	--
Step 9	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 10	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast**router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 11	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 12	**bgp update-delay** *seconds* **Example:** `Router(config-router)# bgp update-delay 1`	Sets the maximum initial delay period before a BGP-speaking networking device sends its first updates. • Use the *seconds* argument to set the delay period.
Step 13	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.10.10.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 14	**neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.10.10.1 update-source loopback 1`	(Optional) Configures a router to select a specific source or interface to receive routing table updates. • This example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface.
Step 15	Repeat Step 13 and Step 14 to configure other BGP neighbors.	--

	Command or Action	Purpose
Step 16	**address-family l2vpn** [**vpls**] **Example:** `Router(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers. • In this example, an L2VPN VPLS address family session is created.
Step 17	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.10.10.1 activate`	Enables the neighbor to exchange information for the L2VPN VPLS address family with the local router. **Note** If you have configured a BGP peer group as a neighbor, you do not use this step. BGP peer groups are activated when a BGP parameter is configured. For example, the **neighbor send-community** command in the next step will automatically activate a peer group.
Step 18	**neighbor** {*ip-address*| *peer-group-name*} **send-community**[**both**| **standard**| **extended**] **Example:** `Router(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 19	Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.	--
Step 20	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 21	**show vfi** **Example:** `Router# show vfi`	(Optional) Displays information about the configured VFI instances.
Step 22	**show ip bgp l2vpn vpls** {**all** | **rd** *vpn-rd*} **Example:** `Router# show ip bgp l2vpn vpls all`	(Optional) Displays information about the L2 VPN VPLS address family.

Examples

The following is sample output from the **show vfi** command that shows two VFIs, CustomerA and CustomerB, with their associated VPN and VPLS IDs:

```
Router# show vfi
Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No
VFI name: customerA, state: down, type: multipoint
  VPN ID: 100, VPLS-ID: 65000:100
  RD: 65000:100, RT: 65000:100
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address     VC ID        Discovered Router ID    S
  10.10.10.1         100        10.10.10.99             Y
VFI name: customerB, state: down, type: multipoint
  VPN ID: 200, VPLS-ID: 65000:200
  RD: 65000:200, RT: 65000:200
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address     VC ID        Discovered Router ID    S
  10.10.10.3         200                10.10.10.98    Y
```

The following is sample output from the **show ip bgp l2vpn vpls all** command that shows two VFIs identified by their VPN route distinguisher:

```
Router# show ip bgp l2vpn vpls all
BGP table version is 5, local router ID is 10.10.10.2
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
Route Distinguisher: 65000:100
*> 65000:100:10.10.10.1/96
                    0.0.0.0                              32768 ?
*>i65000:100:192.168.1.1/96
                    10.10.10.2               0    100      0 ?
Route Distinguisher: 65000:200
*> 65000:200:10.10.10.3/96
                    0.0.0.0                              32768 ?
*>i65000:200:192.168.2.2/96
                    10.10.10.2               0    100      0 ?
```

- What to Do Next, page 71

What to Do Next

To configure more VPLS features, see the main VPLS documentation in the "VPLS Autodiscovery: BGP Based" feature.

Configuration Examples for BGP Support for the L2VPN Address Family

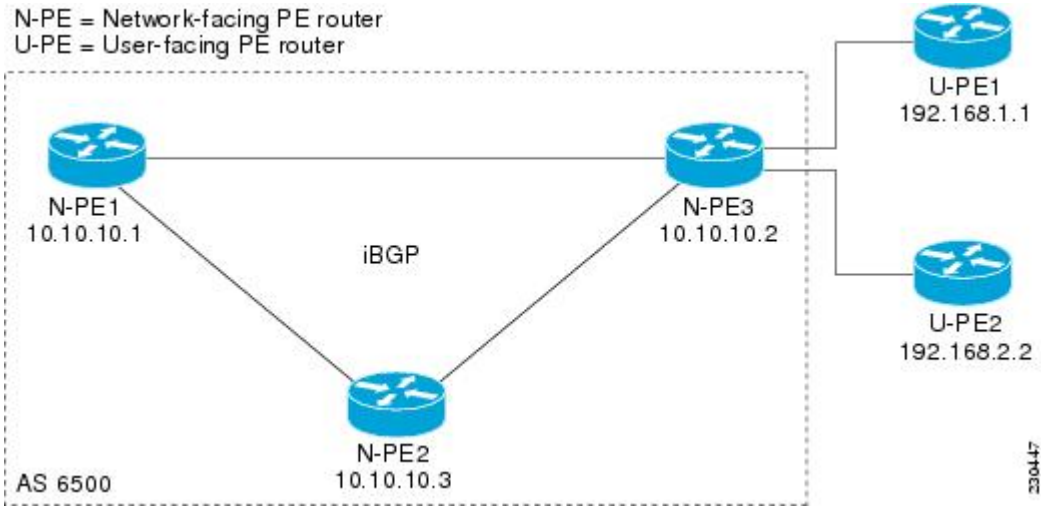
- Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example, page 71

Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family Example

In this configuration example, all the routers in autonomous system 65000 in the figure below are configured to provide BGP support for the L2VPN address family. VPLS autodiscovery is enabled and L2

VFI and VPN IDs are configured. BGP neighbors are configured and activated under L2VPN address family to ensure that the VPLS endpoint provisioning information is saved to a separate L2VPN RIB and then distributed to the other BGP peers in BGP update messages. When the endpoint information is received by the BGP peers, a pseudowire mesh is set up to support L2VPN-based services.

Figure 9 Network Diagram for VPLS Autodiscovery Using BGP and the L2VPN Address Family



Router N-PE1

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 1000 2000
mpls label protocol ldp
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
 ip address 10.0.0.1 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.2 update-source Loopback 1
 neighbor 10.10.10.3 remote-as 65000
 neighbor 10.10.10.3 update-source Loopback 1
!
 address-family l2vpn vpls
```

```
 neighbor 10.10.10.2 activate
 neighbor 10.10.10.2 send-community extended
 neighbor 10.10.10.3 activate
 neighbor 10.10.10.3 send-community extended
 exit-address-family
!
ip classless
```

Router N-PE2

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.2 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
 ip address 10.0.0.2 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.1 update-source Loopback1
 neighbor 10.10.10.3 remote-as 65000
 neighbor 10.10.10.3 update-source Loopback1
!
 address-family l2vpn vpls
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-community extended
 neighbor 10.10.10.3 activate
 neighbor 10.10.10.3 send-community extended
 exit-address-family
!
ip classless
```

Router N-PE3

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
```

```
!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
 ip address 10.0.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.1 update-source Loopback1
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.2 update-source Loopback1
!
 address-family l2vpn vpls
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-community extended
 neighbor 10.10.10.2 activate
 neighbor 10.10.10.2 send-community extended
 exit-address-family
!
ip classless
```

Where to Go Next

For more details about configuring VPLS autodiscovery, see the "VPLS Autodiscovery: BGP Based" feature.

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*
BGP overview	“Cisco BGP Overview” module
Configuring basic BGP tasks	“Configuring a Basic BGP Network” module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index

Feature Information for BGP Support for the L2VPN Address Family

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 6 Feature Information for BGP Support for the L2VPN Address Family

Feature Name	Releases	Feature Information
BGP Support for the L2VPN Address Family	Cisco IOS XE Release 2.6 Cisco IOS XE Release 3.3SG	BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services. The following commands were introduced or modified by this feature: **address-family l2vpn**, **clear ip bgp l2vpn**, and **show ip bgp l2vpn**.

BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced.

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 BGP Event-Based VPN Import

Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Information About BGP Event-Based VPN Import

BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing BGP path import process. BGP Virtual Private Network (VPN) import provides importing functionality for BGP paths where BGP paths are imported from the BGP VPN table into a BGP virtual routing and forwarding (VRF) topology. In the existing path import process, when path updates occur, the import updates are processed during the next scan time which is a configurable interval of 5 to 15 seconds. The scan time adds a delay in the propagation of routes. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available.

Using the BGP Event-Based VPN Import feature, convergence times are significantly reduced because provider edge (PE) routers can propagate VPN paths to customer edge (CE) routers without the scan time delay. Configuration changes such as adding imported route-targets to a VRF are not processed immediately, and are still handled during the 60-second periodic scanner pass.

Import Path Selection Policy

Event-based VPN import introduces three path selection policies:

- All--Import all available paths from the exporting net that match any route target (RT) associated with the importing VRF instance.
- Best path--Import the best available path that matches the RT of the VRF instance. If the best path in the exporting net does not match the RT of the VRF instance, a best available path that matches the RT of the VRF instance is imported.
- Multipath--Import the best path and all paths marked as multipaths that match the RT of the VRF instance. If there are no best path or multipath matches, then the best available path is selected.

Multipath and best path options can be restricted using an optional keyword to ensure that the selection is made only on the configured option. If the **strict** keyword is configured in the **import path selection** command, the software disables the fall back safety option of choosing the best available path. If no paths appropriate to the configured option (best path or multipath) in the exporting net match the RT of the VRF instance, then no paths are imported. This behavior matches the behavior of the software before the BGP Event-Based VPN Import feature was introduced.

When the restriction is not set, paths that are imported as the best available path are tagged. In **show** command output these paths are identified with the wording, "imported safety path."

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the **show** command output. Any path that is marked as "not-in-vrf" is not considered as a best path because paths not in the VRF appear less attractive than paths in the VRF.

Import Path Limit

To control the memory utilization, a maximum limit of the number of paths imported from an exporting net can be specified per importing net. When a selection is made of paths to be imported from one or more exporting net, the first selection priority is a best path, the next selection priority is for multipaths, and the lowest selection priority is for nonmultipaths.

How to Configure BGP Event-Based VPN Import

Configuring a Multiprotocol VRF

Perform this task to configure a multiprotocol VRF that allows you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. In this task, only the IPv4 address family is configured, but we recommend using the multiprotocol VRF configuration for all new VRF configurations.

Note This task is not specific to the BGP Event-Based VPN Import feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **address-family ipv4** [**unicast**]
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **no shutdown**
13. **exit**
14. Repeat Step 3 through Step 13 to bind other VRF instances with an interface.
15. **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	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf-A`	Configures a VRF routing table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 45000:1`	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target both 45000:100`	Creates a route target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to both import routing information from, and export routing information to, the target VPN extended community. • Use the *route-target-ext-community* argument to add the route target extended community attributes to the VRF's list of import, export, or both (import and export) route target extended communities.
Step 6	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters VRF address family configuration mode. • This step is required here to specify an address family for the VRF defined in the previous steps.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1/1`	Enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding vrf-A`	Associates a VRF instance with the interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.4.8.149 255.255.255.0`	Configures an IP address for the interface.
Step 12	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts a disabled interface.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	Repeat Step 3 through Step 13 to bind other VRF instances with an interface.	--
Step 15	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Event-Based VPN Import Processing for BGP Paths

Perform this task to reduce convergence times when BGP paths change by configuring event-based processing for importing BGP paths into a VRF table. Two new CLI commands allow the configuration of a maximum number of import paths per importing net and the configuration of a path selection policy.

This task assumes that you have previously configured the VRF to be used with the VRF address family syntax. To configure a VRF, see the Configuring a Multiprotocol VRF, page 79.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the Configuring Event-Based VPN Import Processing for BGP Paths Example, page 85.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **import path selection** {**all** | **bestpath** [**strict**] | **multipath** [**strict**]}
6. **import path limit** *number-of-import-paths*
7. **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 for the specified routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf vrf-A`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**import path selection** {**all** | **bestpath** [**strict**] | **multipath** [**strict**]} **Example:** `Router(config-router-af)# import path selection all`	Specifies the BGP path selection policy for importing routes into a VRF table. • In this example, all paths that match any RT of the VRF instance are imported.
Step 6	**import path limit** *number-of-import-paths* **Example:** `Router(config-router-af)# import path limit 3`	Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.
Step 7	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Monitoring and Troubleshooting BGP Event-Based VPN Import Processing

Perform the steps in this task as required to monitor and troubleshoot the BGP event-based VPN import processing.

Only partial command syntax for the **show** commands used in this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Referenc*e.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*network-address* [*mask*]]
3. **show ip route** [**vrf** *vrf-name*] [*ip-address* [*mask*]]
4. **debug ip bgp vpnv4 unicast import** {**events** | **updates** [*access-list*]}

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*network-address* [*mask*]]

In this example output, a safe import path selection policy is in effect because the **strict** keyword is not configured using the **import path selection** command. When a path is imported as the best available path (when the bestpath or multipaths are not eligible for import), the path is marked with "imported safety path," as shown in the output.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0

BGP routing table entry for 45000:1:172.17.0.0/16, version 10
Paths: (1 available, best #1, table vrf-A)
Flag: 0x820
   Not advertised to any peer
   2, imported safety path from 50000:2:172.17.0.0/16
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 200, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
```

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the **show** command output.

In the following example output, a path was received from another peer router and was not subject to the VPN importing rules. This path, 10.0.101.2, was added to the VPNv4 table and associated with the vrf-A net because it contains a match of the RD information although the RD information was from the original router. This path is not, however, an RT match for vrf-A and is marked as "not-in-vrf." Note that on the net for vrf-A, this path is not the bestpath because any paths that are not in the VRF appear less attractive than paths in the VRF.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0

BBGP routing table entry for 45000:1:172.17.0.0/16, version 11
Paths: (2 available, best #2, table vrf-A)
Flag: 0x820
   Not advertised to any peer
   2
     10.0.101.2 from 10.0.101.2 (10.0.101.2)
       Origin IGP, metric 100, localpref 100, valid, internal, not-in-vrf
       Extended Community: RT:45000:200
       mpls labels in/out nolabel/16
   2
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 50, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
       mpls labels in/out nolabel/16
```

Step 3 **show ip route** [**vrf** *vrf-name*] [*ip-address* [*mask*]]

In this example output, information about the routing table for VRF vrf-A is displayed:

Example:

```
Router# show ip route vrf vrf-A 172.17.0.0

Routing Table: vrf-A
```

```
Routing entry for 172.17.0.0/16
  Known via "bgp 1", distance 200, metric 50
  Tag 2, type internal
  Last update from 10.0.101.33 00:00:32 ago
  Routing Descriptor Blocks:
  * 10.0.101.33 (default), from 10.0.101.33, 00:00:32 ago
      Route metric is 50, traffic share count is 1
      AS Hops 1
      Route tag 2
      MPLS label: 16
      MPLS Flags: MPLS Required
```

Step 4 **debug ip bgp vpnv4 unicast import** {**events** | **updates** [*access-list*]}

Use this command to display debugging information related to the importing of BGP paths into a VRF instance table. The actual output depends on the commands that are subsequently entered.

Note If no access list to filter prefixes is specified when using the updates keyword, all updates for all prefixes are displayed and this may slow down your network.

Example:

```
Router# debug ip bgp vpnv4 unicast import events

BGP import events debugging is on
```

Configuration Examples for BGP Event-Based VPN Import

Configuring Event-Based VPN Import Processing for BGP Paths Example

In this example configuration, a VRF (vrf-A) is configured and VRF forwarding is applied to Fast Ethernet interface 1/1. In address family mode the import path selection is set to all and the number of import paths is set to 3. Two BGP neighbors are configured under the IPv4 address family and activated under the VPNv4 address family.

```
vrf definition vrf-A
 rd 45000:1
 route-target import 45000:100
 address-family ipv4
  exit-address-family
!
interface FastEthernet1/1
 no ip address
 vrf forwarding vrf-A
 ip address 10.4.8.149 255.255.255.0
 no shut
 exit
!
router bgp 45000
 network 172.17.1.0 mask 255.255.255.0
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 vrf vrf-A
  import path selection all
  import path limit 3
```

```
 exit-address-family
address-family vpnv4
 neighbor 192.168.1.2 activate
 neighbor 192.168.3.2 activate
 end
```

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" module.
- If you want to configure BGP neighbor session options, see the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some advanced BGP features, see the "Configuring Advanced BGP Features" module.

Additional References

The following sections provide references related to the BGP Event-Based VPN Import feature.

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: BGP 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*.
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup
Cisco IOS Master Command List	http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for BGP Event-Based VPN Import

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 7 Feature Information for BGP Event-Based VPN Import

Feature Name	Releases	Feature Information
BGP Event-Based VPN Import	Cisco IOS XE Release 2.6 Cisco IOS XE Release 3.3SG	The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced. The following commands were introduced or modified: • **bgp scan-time** • **import path limit** • **import path selection** • **maximum-path ebgp** • **maximum-path ibgp** • **show ip bgp vpnv4** • **show ip bgp vpnv6**

BGP VPLS Auto Discovery Support on Route Reflector

BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.

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.

Information About BGP VPLS Auto Discovery Support on Route Reflector

BGP VPLS Autodiscovery Support on Route Reflector

In Cisco IOS Release 12.2(33)SRE, BGP VPLS Autodiscovery Support on Route Reflector was introduced. On the Cisco 7600 and Cisco 7200 series routers, BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector. The route reflector reflects the VPLS prefixes to other provider edge (PE) routers so that the PEs do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP VPLS address family on the route reflector.

For an example of a route reflector configuration that can reflect VPLS prefixes, see the Example: BGP VPLS Autodiscovery Support on Route Reflector. For more information about VPLS Autodiscovery, see the VPLS Autodiscovery:BGP Based chapter in the *Cisco IOS MPLS Configuration Guide* .

Configuration Example for BGP VPLS Auto Discovery Support on Route Reflector

Example BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge Route Reflector) is configured as a route reflector capable of reflecting VPLS prefixes. The VPLS address family is configured by **address-family l2vpn vpls** below.

```
hostname PE-RR
!
router bgp 1
 bgp router-id 1.1.1.3
 no bgp default route-target filter
 bgp log-neighbor-changes
neighbor iBGP_PEERS peer-group
neighbor iBGP_PEERS remote-as 1
neighbor iBGP_PEERS update-source Loopback1
neighbor 1.1.1.1 peer-group iBGP_PEERS
neighbor 1.1.1.2 peer-group iBGP_PEERS
!
address-family l2vpn vpls
  neighbor iBGP_PEERS send-community extended
  neighbor iBGP_PEERS route-reflector-client
  neighbor 1.1.1.1 peer-group iBGP_PEERS
  neighbor 1.1.1.2 peer-group iBGP_PEERS
exit-address-family
!
```

Additional References

Related Documents

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

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 BGP VPLS Auto Discovery Support on Route Reflector

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 8 Feature Information for BGP VPLS Auto Discovery Support on Route Reflector

Feature Name	Releases	Feature Information
BGP VPLS Auto Discovery Support on Route Reflector	12.2(33)SRE 15.1(1)SG Cisco IOS XE 3.3SG 15.2(4)S	BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.