Configuring BGP Neighbor Session Options

Last Updated: September 8, 2011

This module describes configuration tasks to configure various options involving Border Gateway Protocol (BGP) neighbor peer sessions. BGP is an interdomain routing protocol designed to provide loop-free routing between organizations. This module contains tasks that use BGP neighbor session commands to configure:

- Fast session deactivation
- Bidirectional Forwarding Detection (BFD) for BGP IPv6 neighbors
- A router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down
- Options to help an autonomous system migration
- TTL Security Check, a lightweight security mechanism to protect External BGP (eBGP) peering sessions from CPU-utilization-based attacks

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.

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

Prerequisites for Configuring BGP Neighbor Session Options

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

Restrictions for Configuring BGP Neighbor Session Options

A router that runs Cisco IOS software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.

Information About Configuring BGP Neighbor Session Options

BGP Neighbor Sessions

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. A BGP-speaking router does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking routers.

A BGP neighbor device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a peer instead of neighbor because a neighbor may imply the idea that the BGP devices are directly connected with no other router in between. Configuring BGP neighbor or peer sessions uses BGP neighbor session commands so this module will prefer the use of the term "neighbor" over "peer."

BGP Support for Fast Peering Session Deactivation

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco IOS software. This timer value is set as the default to protect the BGP routing process from instability that can be caused by peering sessions with other routing protocols. BGP routers typically carry large routing tables, so frequent session resets are not desirable.

BGP Fast Peering Session Deactivation

BGP fast peering session deactivation improves BGP convergence and response time to adjacency changes with BGP neighbors. This feature is event driven and configured on a per-neighbor basis. When this feature is enabled, BGP will monitor the peering session with the specified neighbor. Adjacency changes are detected and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

Selective Address Tracking for BGP Fast Session Deactivation

In Cisco IOS Release 12.4(4)T, 12.2(31)SB, 12.2(33)SRB, and later releases, the BGP Selective Address Tracking feature introduced the use of a route map with BGP fast session deactivation. The **route-map** keyword and *map-name* argument are used with the **neighbor fall-over** BGP neighbor session command to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset. The route map is not used for session establishment.

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

BFD Support of BGP IPv6 Neighbors

In Cisco IOS Release 15.1(2)S and later releases, Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that have an IPv6 address. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. BFD provides faster reconvergence time for BGP after a forwarding path failure.

BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached

Prefix Limits and BGP Peering Sessions

There is a configurable limit on the maximum number of prefixes that a router that is running BGP can receive from a peer router. This limit is configured with the **neighbor maximum-prefix** command. When the router receives too many prefixes from a peer router and the maximum-prefix limit is exceeded, the peering session is disabled or brought down. The session stays down until the network operator manually brings the session back up by entering the **clear ip bgp** command. Entering the **clear ip bgp** command clears stored prefixes.

BGP Neighbor Session Restart with the Maximum Prefix Limit

In Cisco IOS Release 12.0(22)S, 12.2(15)T, 12.2(18)S, and later releases, the **restart** keyword was added to enhance the capabilities of the **neighbor maximum-prefix** command. This enhancement allows the network operator to configure a router to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down. There is configurable time interval at which peering can be reestablished automatically. The configurable timer argument for the **restart** keyword is specified in minutes. The time range is from 1 to 65,535 minutes.

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.

TTL Security Check for BGP Neighbor Sessions

BGP Support for the TTL Security Check

When implemented for BGP, the TTL Security Check feature introduces a lightweight security mechanism to protect eBGP neighbor sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses.

The TTL Security Check feature protects the eBGP neighbor session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP neighbor session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is unnecessary.

Although it is possible to forge the TTL field in an IP packet header, accurately forging the TTL count to match the TTL count from a trusted peer is impossible unless the network to which the trusted peer belongs has been compromised.

The TTL Security Check feature supports both directly connected neighbor sessions and multihop eBGP neighbor sessions. The BGP neighbor session is not affected by incoming packets that contain invalid TTL values. The BGP neighbor session will remain open, and the router will silently discard the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

TTL Security Check for BGP Neighbor Sessions

The BGP Support for TTL Security Check feature is configured with the **neighbor ttl-security** command in router configuration mode or address family configuration mode. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. The *hop-count* argument is used to configure the maximum number of hops that separate the two peers. The TTL value is determined by the router from the configured hop count. The value for this argument is a number from 1 to 254.

TTL Security Check Support for Multihop BGP Neighbor Sessions

The BGP Support for TTL Security Check feature supports both directly connected neighbor sessions and multihop neighbor sessions. When this feature is configured for a multihop neighbor session, the **neighbor ebgp-multihop** router configuration command cannot be configured and is not needed to establish the neighbor session. These commands are mutually exclusive, and only one command is required to establish a multihop neighbor session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.

To configure this feature for an existing multihop session, you must first disable the existing neighbor session with the **no neighbor ebgp-multihop** command. The multihop neighbor session will be restored when you enable this feature with the **neighbor ttl-security** command.

This feature should be configured on each participating router. To maximize the effectiveness of this feature, the *hop-count* argument should be strictly configured to match the number of hops between the local and external network. However, you should also consider path variation when configuring this feature for a multihop neighbor session.

Benefits of the BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect eBGP neighbor sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP autonomous system.

BGP Support for TCP Path MTU Discovery per Session

Path MTU Discovery

The IP protocol family was designed to use a wide variety of transmission links. The maximum IP packet length is 65000 bytes. Most transmission links enforce a smaller maximum packet length limit, called the maximum transmission unit (MTU), which varies with the type of the transmission link. The design of IP accommodates link packet length limits by allowing intermediate routers to fragment IP packets as necessary for their outgoing links. The final destination of an IP packet is responsible for reassembling its fragments as necessary.

All TCP sessions are bounded by a limit on the number of bytes that can be transported in a single packet, and this limit is known as the maximum segment size (MSS). TCP breaks up packets into chunks in a transmit queue before passing packets down to the IP layer. A smaller MSS may not be fragmented at an IP device along the path to the destination device, but smaller packets increase the amount of bandwidth needed to transport the packets. The maximum TCP packet length is determined by both the MTU of the outbound interface on the source device and the MSS announced by the destination device during the TCP setup process.

Path MTU discovery (PMTUD) was developed as a solution to the problem of finding the optimal TCP packet length. PMTUD is an optimization (detailed in RFC 1191) wherein a TCP connection attempts to send the longest packets that will not be fragmented along the path from source to destination. It does this by using a flag, don’t fragment (DF), in the IP packet. This flag is supposed to alter the behavior of an intermediate router that cannot send the packet across a link because it is too long. Normally the flag is off, and the router should fragment the packet and send the fragments. If a router tries to forward an IP datagram, with the DF bit set, to a link that has a lower MTU than the size of the packet, the router will drop the packet and return an ICMP Destination Unreachable message to the source of this IP datagram, with the code indicating "fragmentation needed and DF set." When the source device receives the ICMP message, it will lower the send MSS, and when TCP retransmits the segment, it will use the smaller segment size.

BGP Neighbor Session TCP PMTUD

TCP path MTU discovery is enabled by default for all BGP neighbor sessions, but there are situations when you may want to disable TCP path MTU discovery for one or all BGP neighbor sessions. Although PMTUD works well for larger transmission links (for example, Packet over Sonet links), a badly configured TCP implementation or a firewall may slow or stop the TCP connections from forwarding any packets. In this type of situation, you may need to disable TCP path MTU discovery. In Cisco IOS Release 12.2(33)SRA, 12.2(31)SB, 12.2(33)SXH, 12.4(20)T, and later releases, configuration options were introduced to permit TCP path MTU discovery to be disabled, or subsequently reenabled, either for a single BGP neighbor session or for all BGP sessions. To disable the TCP path MTU discovery globally for all BGP neighbors, use the **no bgp transport path-mtu-discovery** command in router configuration mode. To disable the TCP path MTU discovery for a single neighbor, use the **no neighbor transport path-mtu-discovery** command in router or address family configuration modes. For more details, see the Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 25 or the Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 27.

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 Neighbor Session Options

Configuring Fast Session Deactivation

The tasks in this section show how to 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 route dampening, see the "Configuring Internal BGP Features" module.

Configuring Fast Session Deactivation for a BGP Neighbor

Perform this task to establish a peering session with a BGP neighbor and then configure the peering session for fast session deactivation to improve the network convergence time if the peering session is deactivated.

Enabling fast session deactivation for a BGP neighbor can significantly improve BGP convergence time. However, unstable IGP peers can still 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*]
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **fall-over**
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 50000`	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*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters 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	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 50000`	Establishes a peering session with a BGP neighbor.

	Command or Action	Purpose
Step 6	**neighbor** *ip-address* **fall-over** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 fall-over`	Configures the BGP peering to use fast session deactivation. • BGP will remove all routes learned through this peer if the session is deactivated.
Step 7	**end** **Example:** `Router(config-router-af)# end`	Exits configuration mode and enters privileged EXEC mode.

Configuring Selective Address Tracking for Fast Session Deactivation

Perform this task to configure selective address tracking for fast session deactivation. The optional **route-map** keyword and *map-name* argument of the **neighbor fall-over** command are used to determine if a peering session with a BGP neighbor should be deactivated (reset) when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **fall-over** [**route-map** *map-name*]
6. **exit**
7. **ip prefix-list** *list-name* [**seq** *seq-value* **]**{**deny** *network* / *length* | **permit** *network* / *length*}[**ge** *ge-value*] [**le** *le-value*]
8. **route-map** *map-name* [**permit** | **deny**][*sequence-number*]
9. **match ip address prefix-list** *prefix-list-name* [*prefix-list-name*...]
10. **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.
Step 4	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	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.
Step 5	**neighbor** *ip-address* **fall-over** [**route-map** *map-name*] **Example:** `Router(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR`	Applies a route map when a route to the BGP changes. • In this example, the route map named CHECK-NBR is applied when the route to neighbor 192.168.1.2 changes.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network / length* | **permit** *network / length*}[**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28`	Creates a prefix list for BGP next-hop route filtering. • Selective next-hop route filtering supports prefix- length matching or source-protocol matching on a per-address family basis. • The example creates a prefix list named FILTER28 that permits routes only if the mask length is greater than or equal to 28.
Step 8	**route-map** *map-name* [**permit** | **deny**][*sequence-number*] **Example:** `Router(config)# route-map CHECK-NBR permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named CHECK-NBR is created. If there is an IP address match in the following **match** command, the IP address will be permitted.
Step 9	**match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*] **Example:** `Router(config-route-map)# match ip address prefix-list FILTER28`	Matches the IP addresses in the specified prefix list. • Use the *prefix-list-name* argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 10	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.

• What to Do Next, page 12

What to Do Next

The BGP Support for Next-Hop Address Tracking feature improves the response time of BGP to next-hop changes for routes installed in the RIB, which can also improve overall BGP convergence. For information about BGP next-hop address tracking, see the "Configuring Advanced BGP Features" module.

Configuring BFD for BGP IPv6 Neighbors

In Cisco IOS Release 15.1(2)S and later releases, Bidirectional Forwarding Detection (BFD) can be used for BGP neighbors that have an IPv6 address.

Once it has been verified that BFD neighbors are up, the **show bgp ipv6 unicast neighbors**command will indicate that BFD is being used to detect fast fallover on the specified neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 cef**
5. **interface** *type number*
6. **ipv6 address** *ipv6-address* / *prefix-length*
7. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *multiplier-value*
8. **no shutdown**
9. **exit**
10. **router bgp** *autonomous-system-number*
11. **no bgp default ipv4-unicast**
12. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
13. **neighbor** *ipv6-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ipv6-address* **fall-over bfd**
15. **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	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**ipv6 cef** **Example:** `Router(config)# ipv6 cef`	Enables Cisco Express Forwarding for IPv6.

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/1`	Configures an interface type and number.
Step 6	**ipv6 address** *ipv6-address* / *prefix-length* **Example:** `Router(config-if)# ipv6 address 2001:DB8:1:1::1/64`	Configures an IPv6 address and enables IPv6 processing on an interface.
Step 7	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *multiplier-value* **Example:** `Router(config-if)# bfd interval 500 min_rx 500 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 8	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts an interface.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 11	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default IPv4 unicast address family for establishing peering sessions. • We recommend configuring this command in the global scope.

	Command or Action	Purpose
Step 12	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Router(config-router)# address-family ipv6`	Enters address family configuration mode and enables IPv6 addressing.
Step 13	**neighbor** *ipv6-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:DB8:2:1::4 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table of the local router.
Step 14	**neighbor** *ipv6-address* **fall-over bfd** **Example:** `Router(config-router)# neighbor 2001:DB8:2:1::4 fall-over bfd`	Enables BGP to monitor the peering session of an IPv6 neighbor using BFD.
Step 15	**end** **Example:** `Router(config-router)# end`	Exits configuration mode and enters privileged EXEC mode.

Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded

Perform this task to configure the time interval at which a BGP neighbor session is reestablished by a router when the number of prefixes that have been received from a BGP peer has exceeded the maximum prefix limit.

The network operator can configure a router that is running BGP to automatically reestablish a neighbor session that has been brought down because the configured maximum-prefix limit has been exceeded. No intervention from the network operator is required when this feature is enabled.

Note This task attempts to reestablish a disabled BGP neighbor session at the configured time interval that is specified by the network operator. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. The network operator will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the **warning-only** keyword of the **neighbor maximum-prefix** command can be configured to disable the restart capability, while the network operator corrects the underlying problem.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *minutes*] [**warning-only**]
5. **end**
6. **show ip bgp neighbors** *ip-address*

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 101`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *minutes*] [**warning-only**] **Example:** `Router(config-router)# neighbor 10.4.9.5 maximum-prefix 1000 90 restart 60`	Configures the maximum-prefix limit on a router that is running BGP. • Use the **restart** keyword and *minutes* argument to configure the router to automatically reestablish a neighbor session that has been disabled because the maximum-prefix limit has been exceeded. The configurable range of *minutes*is from 1 to 65535 minutes. • Use the **warning-only** keyword to configure the router to disable the restart capability to allow you to fix a peer that is sending too many prefixes. **Note** If the *minutes*argument is not configured, the disabled session will stay down after the maximum-prefix limit is exceeded. This is the default behavior.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits configuration mode and enters privilaged EXEC mode.
Step 6	**show ip bgp neighbors** *ip-address* **Example:** `Router# show ip bgp neighbors 10.4.9.5`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will display the maximum prefix limit for the specified neighbor and the configured restart timer value. **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 example output from the **show ip bgp neighbors** command verifies that a router has been configured to automatically reestablish disabled neighbor sessions. The output shows that the maximum prefix limit for neighbor 10.4.9.5 is set to 1000 prefixes, the restart threshold is set to 90 percent, and the restart interval is set at 60 minutes.

```
Router# show ip bgp neighbors 10.4.9.5

BGP neighbor is 10.4.9.5,  remote AS 101, internal link
  BGP version 4, remote router ID 10.4.9.5
  BGP state = Established, up for 2w2d
  Last read 00:00:14, hold time is 180, keepalive interval is 60 seconds
  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:         23095      23095
    Route Refresh:          0          0
    Total:              23096      23096
```

```
  Default minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor versions 1/0 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 2, Offset 0, Mask 0x4
  Member of update-group 2
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             0          0
    Prefixes Total:               0          0
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          0
    Used as multipath:          n/a          0
                                 Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Total:                              0          0
!Configured maximum number of prefixes and restart interval information!
  Maximum prefixes allowed 1000
  Threshold for warning message 90%, restart interval 60 min
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.4.9.21, Local port: 179
Foreign host: 10.4.9.5, Foreign port: 11871
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x5296BD2C):
Timer          Starts    Wakeups            Next
Retrans         23098          0             0x0
TimeWait            0          0             0x0
AckHold         23096      22692             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss: 1900546793  snduna: 1900985663  sndnxt: 1900985663     sndwnd:  14959
irs: 2894590641  rcvnxt: 2895029492  rcvwnd:      14978  delrcvwnd:   1406
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 316 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
Datagrams (max data segment is 1460 bytes):
Rcvd: 46021 (out of order: 0), with data: 23096, total data bytes: 438850
Sent: 46095 (retransmit: 0, fastretransmit: 0), with data: 23097, total data by9
```

- Troubleshooting Tips, page 18

Troubleshooting Tips

Use the **clear ip bgp** command to resets a BGP connection using BGP soft reconfiguration. This command can be used to clear stored prefixes to prevent a router that is running BGP from exceeding the maximum-prefix limit. For more details about using BGP soft reconfiguration, see the "Monitoring and Maintaining Basic BGP" task in the "Configuring a Basic BGP Network" module.

Display of the following error messages can indicate an underlying problem that is causing the neighbor session to become disabled. The network operator should check the values that are configured for the maximum-prefix limit and the configuration of any peers that are sending an excessive number of prefixes. The following sample error messages are similar to the error messages that may be displayed:

```
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Up
00:01:14:%BGP-4-MAXPFX:No. of unicast prefix received from 10.10.10.2 reaches 5, max 6
00:01:14:%BGP-3-MAXPFXEXCEED:No.of unicast prefix received from 10.10.10.2:7 exceed limit6
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Down - BGP Notification sent
00:01:14:%BGP-3-NOTIFICATION:sent to neighbor 10.10.10.2 3/1 (update malformed) 0 byte
```

The **bgp dampening** command can be used to configure the dampening of a flapping route or interface when a peer is sending too many prefixes and causing network instability. Use this command only when troubleshooting or tuning a router that is sending an excessive number of prefixes. For more details about BGP route dampening, see the "Configuring Advanced BGP Features" module.

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.

	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 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.
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.

	Command or Action	Purpose
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.
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.

Configuring the TTL Security Check for BGP Neighbor Sessions

Perform this task to allow BGP to establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the BGP neighbor session.

- To maximize the effectiveness of the BGP Support for TTL Security Check feature, we recommend that you configure it on each participating router. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router.

Note

- The **neighbor ebgp-multihop** command is not needed when the BGP Support for TTL Security Check feature is configured for a multihop neighbor session and should be disabled before configuring this feature.
- The effectiveness of the BGP Support for TTL Security Check feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected neighbor sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of the local and remote network. This restriction also includes peers that are on the network segment between the local and remote network.

SUMMARY STEPS

1. **enable**
2. **trace** [*protocol*] *destination*
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *ip-address* **ttl-security hops** *hop-count*
6. **end**
7. **show running-config**
8. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace** [*protocol*] *destination* **Example:** `Router# trace ip 10.1.1.1`	Discovers the routes of the specified protocol that packets will actually take when traveling to their destination. • Enter the **trace** command to determine the number of hops to the specified peer.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65000`	Enters router configuration mode, and creates a BGP routing process.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **ttl-security hops** *hop-count* **Example:** `Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2`	Configures the maximum number of hops that separate two peers. • The *hop-count* argument is set to the number of hops that separate the local and remote peer. If the expected TTL value in the IP packet header is 254, then the number 1 should be configured for the *hop-count* argument. The range of values is a number from 1 to 254. • When the BGP Support for TTL Security Check feature is enabled, BGP will accept incoming IP packets with a TTL value that is equal to or greater than the expected TTL value. Packets that are not accepted are discarded. • The example configuration sets the expected incoming TTL value to at least 253, which is 255 minus the TTL value of 2, and this is the minimum TTL value expected from the BGP peer. The local router will accept the peering session from the 10.1.1.1 neighbor only if it is one or two hops away.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits configuration mode and enters privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config | begin bgp`	(Optional) Displays the contents of the currently running configuration file. • The output of this command displays the configuration of the **neighbor ttl-security** command for each peer under the BGP configuration section of output. That section includes the neighbor address and the configured hop count. **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 neighbors** [*ip-address*] **Example:** `Router# show ip bgp neighbors 10.4.9.5`	(Optional) Displays information about the TCP and BGP connections to neighbors. • This command displays "External BGP neighbor may be up to *number* hops away" when the BGP Support for TTL Security Check feature is enabled. The *number* value represents the hop count. It is a number from 1 to 254. **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 configuration of the BGP Support for TTL Security Check feature can be verified with the **show running-config** and **show ip bgp neighbors**commands. This feature is configured locally on each peer, so there is no remote configuration to verify.

The following is sample output from the **show running-config** command. The output shows that neighbor 10.1.1.1 is configured to establish or maintain the neighbor session only if the expected TTL count in the incoming IP packet is 253 or 254.

```
Router# show running-config
| begin bgp
```

```
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 55000
 neighbor 10.1.1.1 ttl-security hops 2
 no auto-summary
.
.
.
```

The following is sample output from the **show ip bgp neighbors** command. The output shows that the local router will accept packets from the 10.1.1.1 neighbor if it is no more than 2 hops away. The configuration of this feature is displayed in the address family section of the output. The relevant line is shown in bold in the output.

```
Router# show ip bgp neighbors 10.1.1.1
BGP neighbor is 10.1.1.1,  remote AS 55000, external link
  BGP version 4, remote router ID 10.2.2.22
  BGP state = Established, up for 00:59:21
  Last read 00:00:21, hold time is 180, keepalive interval is 60 seconds
  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:                  2          2
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           226        227
    Route Refresh:          0          0
    Total:                228        229
  Default minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  Member of update-group 1
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               0          0
    Prefixes Total:                 0          0
    Implicit Withdraw:              0          0
    Explicit Withdraw:              0          0
    Used as bestpath:             n/a          0
    Used as multipath:            n/a          0
                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Total:                                0          0
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 2; dropped 1
  Last reset 00:59:50, due to User reset
  External BGP neighbor may be up to 2 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.2.2.22, Local port: 179
Foreign host: 10.1.1.1, Foreign port: 11001
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0xCC28EC):
Timer          Starts    Wakeups            Next
Retrans            63          0             0x0
TimeWait            0          0             0x0
AckHold            62         50             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss:  712702676  snduna:  712703881  sndnxt:  712703881     sndwnd:  15180
irs: 2255946817  rcvnxt: 2255948041  rcvwnd:      15161  delrcvwnd:   1223
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
```

```
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):
Rcvd: 76 (out of order: 0), with data: 63, total data bytes: 1223
Sent: 113 (retransmit: 0, fastretransmit: 0), with data: 62, total data bytes: 4
```

Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following tasks:

- Disabling TCP Path MTU Discovery Globally for All BGP Sessions, page 25
- Disabling TCP Path MTU Discovery for a Single BGP Neighbor, page 27
- Enabling TCP Path MTU Discovery Globally for All BGP Sessions, page 30
- Enabling TCP Path MTU Discovery for a Single BGP Neighbor, page 32

Disabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to disable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but we recommend that you enter the **show ip bgp neighbors** command to ensure that TCP path MTU discovery is enabled.

This task assumes that you have previously configured BGP neighbors with active TCP connections.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address*]
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **no bgp transport path-mtu-discovery**
6. **end**
7. **show ip bgp neighbors** [*ip-address*]

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	**show ip bgp neighbors** [*ip-address*] **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • Use this command to determine whether BGP neighbors have TCP path MTU discovery enabled. **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 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.
Step 5	**no bgp transport path-mtu-discovery** **Example:** `Router(config-router)# no bgp transport path-mtu-discovery`	Disables TCP path MTU discovery for all BGP sessions.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip bgp neighbors** [*ip-address*] **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will not display that any neighbors have TCP path MTU enabled. **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 sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

The following is sample output from the **show ip bgp neighbors** command after the **no bgp transport path-mtu-discovery** command has been entered. Note that the path mtu entries are missing.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle
```

Disabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an internal BGP (iBGP) neighbor and then disable TCP path MTU discovery for the BGP neighbor session. The **neighbor transport** command can be used in router configuration or address family configuration mode.

This task assumes that you know that TCP path MTU discovery is enabled by default for all your BGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
5. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address*| *peer-group-name*} **activate**
7. **no neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**}
8. **end**
9. **show ip bgp neighbors**

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.
Step 4	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example creates an IPv4 unicast address family session.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 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.
Step 6	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.1 activate`	Activates the neighbor under the IPv4 address family.
Step 7	**no neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**} **Example:** `Router(config-router-af)# no neighbor 172.16.1.1 transport path-mtu-discovery`	Disables TCP path MTU discovery for a single BGP neighbor. • In this example, TCP path MTU discovery is disabled for the neighbor at 172.16.1.1.
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** **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will not display that the neighbor has TCP path MTU discovery enabled. **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 sample output shows that TCP path MTU discovery has been disabled for BGP neighbor 172.16.1.1 but that it is still enabled for BGP neighbor 192.168.2.2. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.1,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.17.1.99
```

```
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.1
  Address tracking requires at least a /24 route to the peer
  Connections established 1; dropped 0
  Last reset never
.
.
.
SRTT: 165 ms, RTTO: 1172 ms, RTV: 1007 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle
.
.
.
BGP neighbor is 192.168.2.2,  remote AS 50000, external link
  BGP version 4, remote router ID 10.2.2.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 192.168.2.2
  Address tracking requires at least a /24 route to the peer
  Connections established 2; dropped 1
  Last reset 00:05:11, due to User reset
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 210 ms, RTTO: 904 ms, RTV: 694 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

Enabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to enable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but if the BGP Support for TCP Path MTU Discovery per Session feature has been disabled, you can use this task to reenable it. To verify that TCP path MTU discovery is enabled, use the **show ip bgp neighbors** command.

This task assumes that you have previously configured BGP neighbors with active TCP connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp transport path-mtu-discovery**
5. **end**
6. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**bgp transport path-mtu-discovery** **Example:** `Router(config-router)# bgp transport path-mtu-discovery`	Enables TCP path MTU discovery for all BGP sessions.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	**show ip bgp neighbors** **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will show that all neighbors have TCP path MTU discovery enabled. **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 sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
```

```
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

Enabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an eBGP neighbor and then enable TCP path MTU discovery for the BGP neighbor session. The **neighbor transport** command can be used in router configuration or address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
5. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address*| *peer-group-name*} **activate**
7. **neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**}
8. **end**
9. **show ip bgp neighbors** [*ip-address*]

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 for the specified routing process.
Step 4	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters 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	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.2.2 remote-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.
Step 6	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.2.2 activate`	Activates the neighbor under the IPv4 address family.
Step 7	**neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**} **Example:** `Router(config-router-af)# neighbor 192.168.2.2 transport path-mtu-discovery`	Enables TCP path MTU discovery for a single BGP neighbor.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip bgp neighbors** [*ip-address*] **Example:** `Router# show ip bgp neighbors 192.168.2.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **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 sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for the BGP neighbor at 192.168.2.2. Two entries in the output--Transport(tcp) path-mtu-discovery is enabled and path-mtu capable--show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors 192.168.2.2
BGP neighbor is 192.168.2.2,  remote AS 50000, external link
  BGP version 4, remote router ID 10.2.2.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 192.168.2.2
  Address tracking requires at least a /24 route to the peer
  Connections established 2; dropped 1
  Last reset 00:05:11, due to User reset
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 210 ms, RTTO: 904 ms, RTV: 694 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

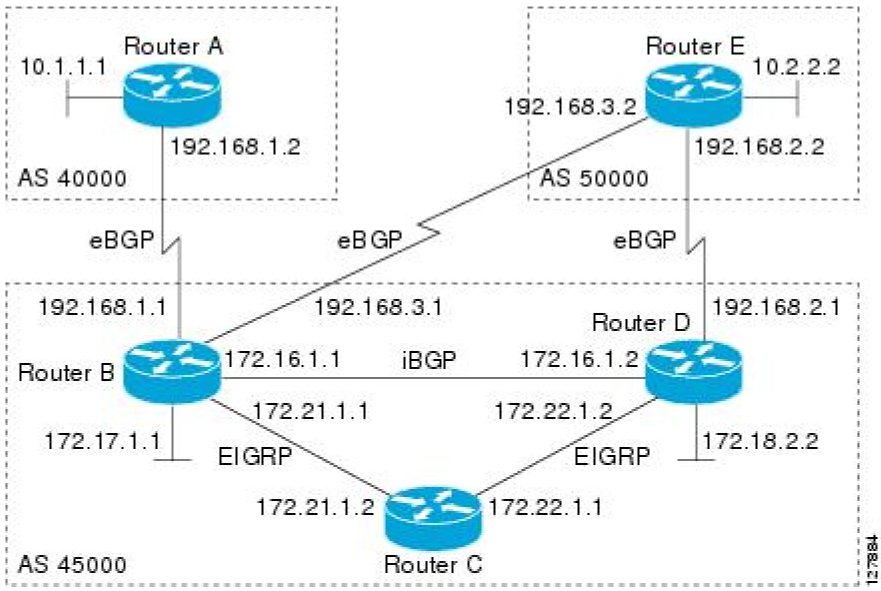
Implementing BGP Dynamic Neighbors Using Subnet Ranges

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 1 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.
Step 2	**configure terminal** **Example:** `RouterB# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.	--
Step 14	**enable** **Example:** `RouterE> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on Router E.

	Command or Action	Purpose
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 Neighbor Session Options

Example Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on Router A and Router B to monitor and use fast peering session deactivation for the neighbor session between the two routers. Although fast peering session deactivation is not required at both routers in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Router A

```
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
 neighbor 192.168.1.1 fall-over
 end
```

Router B

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over
 end
```

Example Configuring Selective Address Tracking for Fast Session Deactivation

The following example shows how to configure the BGP peering session to be reset if a route with a prefix of /28 or a more specific route to a peer destination is no longer available:

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over route-map CHECK-NBR
 exit
ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28
route-map CHECK-NBR permit 10
 match ip address prefix-list FILTER28
 end
```

Example Configuring BFD for a BGP IPv6 Neighbor

The following example configures FastEthernet interface 0/1 with the IPv6 address 2001:DB8:4:1::1. Bidirectional Forwarding Detection (BFD) is configured for the BGP neighbor at 2001:DB8:5:1::2. BFD will track forwarding path failure of the BGP neighbor and provide faster reconvergence time for BGP after a forwarding path failure.

```
ipv6 unicast-routing
ipv6 cef
interface fastethernet 0/1
 ipv6 address 2001:DB8:4:1::1/64
 bfd interval 500 min_rx 500 multiplier 3
 no shutdown
 exit
router bgp 65000
 no bgp default ipv4-unicast
 address-family ipv6 unicast
 neighbor 2001:DB8:5:1::2 remote-as 65001
 neighbor 2001:DB8:5:1::2 fall-over bfd
 end
```

Example Restart Session After Maximum Number of Prefixes From Neighbor Reached

The following example sets the maximum number of prefixes allowed from the neighbor at 192.168.6.6 to 2000 and configures the router to reestablish a peering session after 30 minutes if one has been disabled:

```
router bgp 101
 network 172.16.0.0
 neighbor 192.168.6.6 maximum-prefix 2000 restart 30
```

Examples Configuring Dual-AS Peering for Network Migration

The following examples show how to configure and verify this feature:

- Example Dual-AS Configuration, page 43
- Example Dual-AS Confederation Configuration, page 43
- Example Replace-AS Configuration, page 44

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-AS Configuration

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
```

Example Configuring the TTL-Security Check

The example configurations in this section show how to configure the BGP Support for TTL Security Check feature.

The following example uses the **trace** command to determine the hop count to an eBGP peer. The hop count number is displayed in the output for each networking device that IP packets traverse to reach the specified neighbor. In the following example, the hop count for the 10.1.1.1 neighbor is 1.

```
Router# trace ip 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.1.1
  1 10.1.1.1 0 msec *  0 msec
```

The following example sets the hop count to 2 for the 10.1.1.1 neighbor. Because the hop-count argument is set to 2, BGP will accept only IP packets with a TTL count in the header that is equal to or greater than 253.

```
Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
```

Examples Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following configuration examples:

Example Disabling TCP Path MTU Discovery Globally for All BGP Sessions

The following example shows how to disable TCP path MTU discovery for all BGP neighbor sessions. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been disabled.

```
enable
configure terminal
 router bgp 45000
  no bgp transport path-mtu-discovery
  end
show ip bgp neighbors
```

Example Disabling TCP Path MTU Discovery for a Single BGP Neighbor

The following example shows how to disable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2:

```
enable
configure terminal
 router bgp 45000
  neighbor 192.168.2.2 remote-as 50000
  neighbor 192.168.2.2 activate
  no neighbor 192.168.2.2 transport path-mtu-discovery
  end
show ip bgp neighbors 192.168.2.2
```

Example Enabling TCP Path MTU Discovery Globally for All BGP Sessions

The following example shows how to enable TCP path MTU discovery for all BGP neighbor sessions. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been enabled.

```
enable
configure terminal
 router bgp 45000
  bgp transport path-mtu-discovery
  end
show ip bgp neighbors
```

Example Enabling TCP Path MTU Discovery for a Single BGP Neighbor

The following example shows how to enable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been enabled.

```
enable
configure terminal
 router bgp 45000
  neighbor 192.168.2.2 remote-as 50000
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 transport path-mtu-discovery
  end
show ip bgp neighbors 192.168.2.2
```

Example Implementing BGP Dynamic Neighbors Using Subnet Ranges

In Cisco IOS Release 12.2(33)SXH, support for BGP dynamic neighbors was introduced. The following example configurations show how to implement BGP dynamic neighbors using subnet ranges.

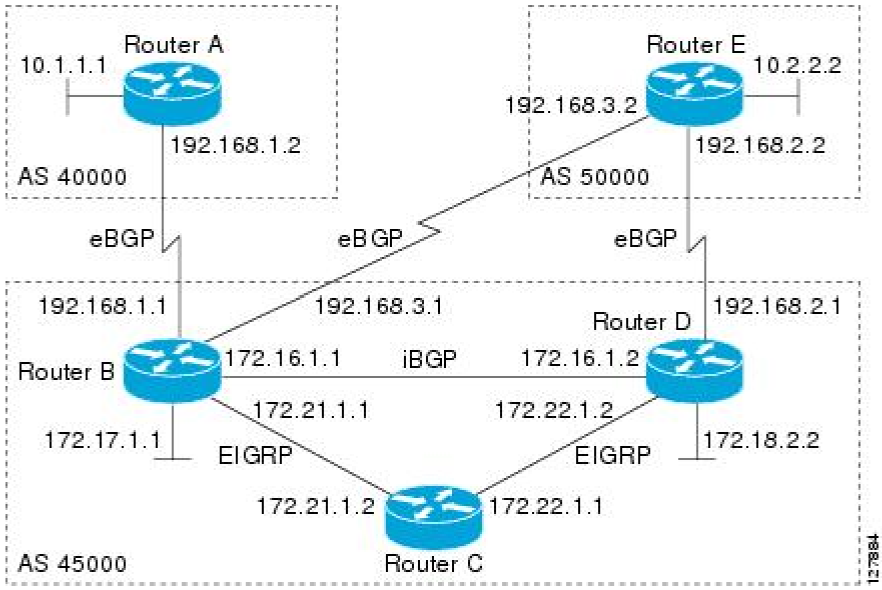
In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added 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 2 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
```

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 some advanced BGP features including BGP next-hop address tracking and route dampening, see the "Configuring Advanced BGP Features" module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	"Cisco BGP Overview" module
Conceptual and configuration details for basic BGP tasks	"Configuring a Basic BGP Network" module
Conceptual and configuration details for advanced BGP tasks	"Configuring Advanced BGP Features" module
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases
Bidirectional Forwarding Detection configuration tasks	*Cisco IOS XE IP Routing: BFD Configuration Guide*

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

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

RFCs

RFC	Title
RFC 1191	Path MTU Discovery
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

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

Feature Information for Configuring BGP Neighbor Session Options

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

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

Table 1 Feature Information for Configuring BGP Neighbor Session Options Features

Feature Name	Releases	Feature Information
BGP Dynamic Neighbors	12.2(33)SXH 15.1(2)T 15.0(1)S	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 output for three **show** commands has been updated to display information about dynamic neighbors. 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**, **show ip bgp summary**.

Feature Name	Releases	Feature Information
BGP Restart Session After Max-Prefix Limit	12.0(22)S 12.2(15)T 12.2(18)S 15.0(1)S	The BGP Restart Session After Max-Prefix Limit feature enhanced the capabilities of the **neighbor maximum-prefix** command with the introduction of the **restart** keyword. This enhancement allows the network operator to configure the time interval at which a peering session is reestablished by a router when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit. The following commands were modified by this release: **neighbor maximum-prefix**, **show ip bgp neighbors**.
BGP Selective Address Tracking	12.4(4)T 12.2(31)SB 12.2(33)SRB	The BGP Selective Address Tracking feature introduced the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The following commands were modified by this feature: **bgp nexthop**, **neighbor fall-over**.

Feature Name	Releases	Feature 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	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, and 12.2(33)SXI1, 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 Release 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-path**, **set extcommunity**, **set origin**, **soo**, all **show ip bgp** commands that display an

Feature Name	Releases	Feature Information
		autonomous system number, and **show ip extcommunity-list**.
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	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 Support for Fast Peering Session Deactivation	12.0(29)S 12.3(14)T 12.2(33)SRA 12.2(31)SB 12.2(33)SXH 15.0(1)S	The BGP Support for Fast Peering Session Deactivation feature introduced an event-driven notification system that allows a Border Gateway Protocol (BGP) process to monitor BGP peering sessions on a per-neighbor basis. This feature improves the response time of BGP to adjacency changes by allowing BGP to detect an adjacency change and deactivate the terminated session in between standard BGP scanning intervals. Enabling this feature improves overall BGP convergence. The following command was modified by this feature: **neighbor fall-over**.

Feature Name	Releases	Feature Information
BGP Support for TCP Path MTU Discovery per Session	12.2(33)SRA 12.2(31)SB 12.2(33)SXH 12.4(20)T 15.0(1)S	BGP support for TCP path maximum transmission unit (MTU) discovery introduced the ability for BGP to automatically discover the best TCP path MTU for each BGP session. The TCP path MTU is enabled by default for all BGP neighbor sessions, but you can disable, and subsequently enable, the TCP path MTU globally for all BGP sessions or for an individual BGP neighbor session. The following commands were introduced or modified by this feature: **bgp transport**, **neighbor transport**, **show ip bgp neighbors**.
BGP Support for TTL Security Check	12.0(27)S 12.3(7)T 12.2(25)S 12.2(18)SXE 15.0(1)S	The BGP Support for TTL Security Check feature introduced a lightweight security mechanism to protect external Border Gateway Protocol (eBGP) peering sessions from CPU utilization-based attacks using forged IP packets. Enabling this feature prevents attempts to hijack the eBGP peering session by a host on a network segment that is not part of either BGP network or by a host on a network segment that is not between the eBGP peers. The following commands were introduced or modified by this feature: **neighbor ttl-security**, **show ip bgp neighbors**.
BGP IPv6 Client for Single-Hop BFD	15.1(2)S	Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that use an IPv6 address. The following command was modified by this feature: **neighbor fall-over**.