Configuring BGP Neighbor Session Options

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:

- 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

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 uses 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 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 devices 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 The **neighbor fall-over** command is not supported in Cisco IOS Release 15.0(1)SY. The **route-map** and *map-name* keyword-argument pair in the **bgp nexthop** command are not supported in Cisco IOS Release 15.0(1)SY.

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

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.

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 software, 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 configuration mode or address family configuration mode. For more details, see the “Disabling TCP Path MTU Discovery Globally for All BGP Sessions” section or the “Disabling TCP Path MTU Discovery for a Single BGP Neighbor” section.

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. Enter one of the following commands:
   - **address-family ipv4** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
   - **address-family ipv6** [**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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode to create or configure a BGP routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 50000`	
Step 4	Enter one of the following commands: • **address-family ipv4** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] • **address-family ipv6** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast vrf blue`	Enters address family configuration mode and enables IPv4 or IPv6 addressing. Perform this step when configuring fast session deactivation for a VRF address-family. **Note** Step 4 is only required if you are configuring fast session deactivation on a VRF. If you are not configuring fast session deactivation on a VRF, skip this step and perform the following commands under router BGP mode (config-router) rather than address family configuration mode (config-router-af).
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.1 remote-as 50000`	Establishes a peering session with a BGP neighbor.
Step 6	**neighbor** *ip-address* **fall-over** **Example:** `Device(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:** `Device(config-router-af)# end`	Exits address family 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(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:** `Device(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:** `Device(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:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network* / *length* | **permit** *network* / *length*} [**ge** *ge-value*] [**le** *le-value*]	Creates a prefix list for BGP next-hop route filtering.

	Command or Action	Purpose
	Example: `Device(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28`	• 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:** `Device(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:** `Device(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:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template. For details about peer policy inheritance, see the "Configuring Peer Policy Template Inheritance with the inherit peer-policy Command" section or the "Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command" section.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables Cisco Express Forwarding for IPv6.
Step 5	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures an interface type and number.
Step 6	**ipv6 address** *ipv6-address* / *prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1:1::1/64`	Configures an IPv6 address and enables IPv6 processing on an interface.

	Command or Action	Purpose
Step 7	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *multiplier-value* **Example:** `Device(config-if)# bfd interval 500 min_rx 500 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 8	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts an interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 11	**no bgp default ipv4-unicast** **Example:** `Device(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.
Step 12	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Device(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:** `Device(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:** `Device(config-router-af)# neighbor 2001:DB8:2:1::4 fall-over bfd`	Enables BGP to monitor the peering session of an IPv6 neighbor using BFD.

	Command or Action	Purpose
Step 15	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

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.

Before you begin

- 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:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**trace** [*protocol*] *destination* **Example:** `Device# 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:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode, and creates a BGP routing process.
Step 5	**neighbor** *ip-address* **ttl-security hops** *hop-count* **Example:** `Device(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:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 7	**show running-config** **Example:**	(Optional) Displays the contents of the currently running configuration file.

	Command or Action	Purpose
	`Device# show running-config | begin bgp`	• 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:** `Device# 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

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.

Before you begin

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# 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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(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:** `Device(config-router)# no bgp transport path-mtu-discovery`	Disables TCP path MTU discovery for all BGP sessions.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# 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 mode or address family configuration mode.

Before you begin

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(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:** `Device(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:** `Device(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:** `Device(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:** `Device(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:**	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router-af)# end`	
Step 9	**show ip bgp neighbors** **Example:** `Device# 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.

Before you begin

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**bgp transport path-mtu-discovery** **Example:** `Device(config-router)# bgp transport path-mtu-discovery`	Enables TCP path MTU discovery for all BGP sessions.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	**show ip bgp neighbors** **Example:** `Device# 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 mode 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(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:** `Device(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:** `Device(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:** `Device(config-router-af)# neighbor 192.168.2.2 activate`	Activates the neighbor under the IPv4 address family.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**} **Example:** `Device(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:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# 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
```

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 device A and device B to monitor and use fast peering session deactivation for the neighbor session between the two devices. Although fast peering session deactivation is not required at both devices in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Device A

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

Device 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: 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
```

Where to Go Next

For information about advertising the bandwidth of an autonomous system exit link as an extended community, refer to the “BGP Link Bandwidth” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
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
Bidirectional Forwarding Detection configuration tasks	*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 http://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 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 15.2(3)T 15.2(4)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**. In Cisco IOS Release 15.2(4)S, support was added for the Cisco 7200 series router.