MPLS VPNs over IP Tunnels

The MPLS VPNs over IP Tunnels feature introduces the capability to deploy layer 3 Virtual Private Network (VPN) services, as proposed in RFC 2547, *BGP/MPLS VPNs*, over an IP core network using L2TPv3 multipoint tunnelling instead of Multiprotocol Label Switching (MPLS). This feature allows L2TPv3 tunnels to be configured as multipoint tunnels to transport IP VPN services across the core IP network. Because multipoint tunnels support multiple end points, only a single tunnel needs to be configured on each Provider Edge (PE) router. This feature also introduces a simple packet validation mechanism to enforce VPN integrity.

Feature History for MPLS VPNs over IP Tunnels

Release	Modification
12.0(28)S	This feature was introduced.
12.0(30)S	Support for the Cisco 12000 series Internet router, the Route Processor (RP), and Performance Route Processor (PRP) was integrated into Cisco IOS Release 12.0(30)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

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

Prerequisites for MPLS VPNs over IP Tunnels

- CEF or dCEF (for distributed platforms) must be enabled on all participating routers.

Supported Line Cards for the Cisco 12000 Series Internet Router

Table 1 shows the status of Cisco 12000 Series Internet Router line card support in Cisco IOS Release 12.0(30)S. Support is shown for backbone-facing interfaces (BFIs) in the network core and customer-facing interfaces (CFI) on the network edge for the Layer 2 Tunnel Protocol Version 3 (L2TPv3) feature and the MPLS VPNs over IP Tunnels feature.

Table 1 Line Card Support in Cisco IOS Release 12.0(30)S

Cisco 12000 Series Internet Router Line Card	L2TPv3 CFI (Edge)	L2TPv3 BFI (Core)	2547bis CFI (Edge)	2547bis BFI (Core)
4 Port Channelized OC12 B	yes	yes	yes	yes
1 port channelized OC-48 to DS3	no	no	no	no
4 port-GE line card for Cisco 12000 Series	no	yes	yes	yes
Channelized OC3/STM1	yes	no	yes	no
Cisco 12000 Series 4-Port OC-12/STM-4 ATM ISE Multimode Line Card	yes	no	yes	no
Cisco 12000 Series 4-Port OC-3c/STM-1c ATM ISE Line Card	yes	no	yes	no

For more information about supported line cards that run on the Cisco 12000 Series Internet router in Cisco IOS Release 12.0(30)S, see the *Cross-Platform Release Notes for Cisco IOS Release 12.0 S*.

Information About MPLS VPNs over IP Tunnels

Deploying Layer 3 VPNs Over Multipoint L2TPv3 Tunnels

VPN services have been traditionally deployed over IP core networks by configuring MPLS or through L2TPv3 tunnels using point-to-point links. This feature introduces the capability to deploy layer 3 VPN services by configuring multipoint L2TPv3 tunnels over an existing IP core network. This feature is configured on only the PE routers and requires no configuration on the core routers. The L2TPv3 multipoint tunnel network allows layer 3 VPN services to be carried through the core without the configuration of MPLS. L2TPv3 multipoint tunnelling supports multiple tunnel end points, which creates a full mesh topology that requires only a single tunnel to be configured on each PE router. This feature provides the capability for VPN traffic to be carried from enterprise networks across cooperating service provider core networks to remote sites.

BGP Advertises Tunnel Type and Tunnel Capabilities Between PE Routers

Border Gateway Protocol (BGP) is used to advertise the tunnel endpoints and the subaddress family indentifier (SAFI) specific attributes (which contains the tunnel type, and tunnel capabilities). This feature introduces the tunnel SAFI and the BGP SAFI-Specific Attribute (SSA) attribute. The tunnel SAFI defines the tunnel endpoint and carries the endpoint IPv4 address and next hop. The tunnel SAFI is identified by the SAFI number 64. The BGP SSA carries the BGP preference and BGP flags. It also carries the tunnel cookie, tunnel cookie length, and session ID. The BGP SSA is identified by attribute number 19.

These attributes allow BGP to distribute tunnel encapsulation information between PE routers. VPNv4 traffic is routed through these tunnels. The next hop, advertised in BGP VPNv4 updates, determines which tunnel to use for routing tunnel traffic.

Configuring the PE Routers and Managing Address Space

A single multipoint L2TPv3 tunnel is configured on each PE router. The VPN is created by configuring a unique Virtual Routing and Forwarding (VRF) instance. The tunnel that transports the VPN traffic across the core network resides in its own address space. A special purpose VRF called a Resolve in VRF (RiV) is created to manage the tunnel address space. The address space configured under the RiV is associated with the tunnel, and a static route is configured in the RiV to route outgoing traffic through the tunnel.

Packet Validation Mechanism

This feature provides a simple mechanism to validate received packets from appropriate peers. The multipoint L2TPv3 tunnel header is automatically configured with a 64-bit cookie and L2TPv3 session ID. This packet validation mechanism is intended to protect the VPN from illegitimate traffic sources, such as injecting a rogue packet into the tunnel to gain access to the VPN. The cookie and session ID are not user configurable; however, they are visible in the packet as it's routed between the two tunnel end-points. This packet validation mechanism will not protect the VPN from hackers who have the ability to monitor legitimate traffic between PE routers.

How to Configure MPLS VPNs over IP Tunnels

To deploy layer 3 VPN services over multipoint L2TPv3 tunnels, you will create a VRF instance, create the multipoint L2TPv3 tunnel, redirect the VPN IP traffic to the tunnel, and configure the BGP VPNv4 exchange so that BGP updates are filtered through a route-map and prefixes are resolved in the VRF table. The configuration steps are described in the following sections:

- Configuring the VRF for the L2TPv3 Tunnel, page 4
- Configuring the Multipoint L2TPv3 Tunnel, page 6
- Configuring a Route Map for the Layer 3 VPN, page 8
- Defining the Address Space and Configuring BGP, page 9
- Verifying the VRF and RiV, page 11
- Verifying the Multipoint L2TPv3 Tunnel, page 13

Configuring the VRF for the L2TPv3 Tunnel

The VPN is created by configuring a unique Virtual Routing and Forwarding (VRF) instance. The tunnel that transports the VPN traffic across the core network resides in its own address space. A special purpose VRF called a Resolve in VRF (RiV) is created to manage the tunnel address space.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *as-number:network-number* | *ip-address:network number*
5. **route-target import** *as-number:network-number* | *ip-address:network number*
6. **route-target export** *as-number:network-number* | *ip-address:network number*
7. **exit**
8. **ip vrf** *vrf-name*
9. **rd** *as-number:network-number* | *ip-address:network number*
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	**ip vrf vrf-name** **Example:** `Router(config)# ip vrf CUSTOMER_A`	Creates a VRF routing table and specifies the VRF name (or tag). • The **ip vrf** command creates a VRF routing table and a CEF table, which are both named using the *vrf-name* argument. Associated with these tables is the default route-distinguisher value.
Step 4	**rd as-number:network-number | ip-address:network-number** **Example:** `Router(config-vrf)# rd 100:110`	Creates routing and forwarding tables for the VRF instance created in step 3. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).
Step 5	**route-target [export | import | both] as-number:network-number | ip-address:network-number** **Example:** `Router(config-vrf)# route-target import 100:1000`	Imports routing information from the target VPN extended community.
Step 6	**route-target [export | import | both] as-number:network-number | ip-address:network-number** **Example:** `Router(config-vrf)# route-target export 100:1000`	Exports routing information to the target VPN extended community.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 8	**ip vrf vrf-name** **Example:** `Router(config)# ip vrf MY_RIV`	Creates the special Resolve in VRF (RiV) VRF instance and table that will be used for the tunnel and redirection of the IP address. • Creates a VRF routing table and specifies the VRF name (or tag). The **ip vrf** command creates a VRF routing table and a CEF table; both are named using the vrf-name argument. Associated with these tables is the default route-distinguisher value.
Step 9	**rd as-number:network-number | ip-address:network-number** **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for the VRF instance created in step 8. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).

Step 10	**end**
	Example:
	`Router(config-vrf)# end`
		Exits VRF configuration mode and enters privileged EXEC mode.

What to Do Next

Proceed to the next task "Configuring the Multipoint L2TPv3 Tunnel."

Configuring the Multipoint L2TPv3 Tunnel

Border Gateway Protocol (BGP) is used to advertise the tunnel type, tunnel capabilities, and tunnel-specific attributes. BGP is also used to distribute VPNv4 routing information between PE routers on the edge of the network, which maintains peering relationships between the VPN service and VPN sites. The next hop advertised in BGP VPNv4 updates triggers tunnel endpoint discovery.

Prerequisites

The IP address of the interface, specified as the tunnel source, should match the IP address used by BGP as the next hop for the VPNv4 update. The BGP configuration will include the **neighbor** *ip-address* **update-source loopback 0** command.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **interface tunnel** *interface-number*
4. **ip vrf forwarding** *RiV-name*
5. **ip address** *ip-address subnet-mask*
6. **tunnel source loopback** *interface-number*

7. **tunnel mode l3vpn l2tpv3 multipoint**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel interface-number** **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode, and creates the tunnel.
Step 4	**ip vrf forwarding RiV-name** **Example:** `Router(config-if)# ip vrf forwarding MY_RIV`	Associates the VRF with an interface or the subinterface. • The RiV name is configured for the VRF argument in this step.
Step 5	**ip address ip-address subnet-mask** **Example:** `Router(config-if)# ip-address 172.16.1.3 255.255.255.255`	Specifies the IP address for the tunnel.
Step 6	**tunnel source loopback interface-number** **Example:** `Router(config-if)# tunnel source loopback 0`	Associates the tunnel source IP address with the loopback interface.
Step 7	**tunnel mode l3vpn l2tpv3 multipoint** **Example:** `Router(config-if)# tunnel mode l3vpn l2tpv3 multipoint`	Sets the mode for the layer 3 VPN tunnel as “l2tpv3 multipoint”.

Step 8	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Troubleshooting

The **clear tunnel l3vpn l2tpv3** command can be used to generate and distribute a new L2TPv3 session for a layer 3 VPN. This command is issued on the PE router. The *hold-time* argument is used to configure the amount of time that the existing session will remain valid, while the new session is propagated to peers. The default value for the *hold-time* argument is 15 seconds. This should be enough time for most networks. However, this value can be increased if it takes longer for the new session to propagate to all other PE routers.

What to Do Next

Proceed to the next task "Configuring a Route Map for the Layer 3 VPN."

Configuring a Route Map for the Layer 3 VPN

A route map must be configured in order to set the next hop to be resolved within the VRF table.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **route-map** *map-name*
4. **set ip next-hop in-vrf** *RiV-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* [**permit**|**deny**] [*sequence-number*] **Example:** `Router(config)# route-map SELECT_UPDATE_FOR_L3VPN permit 10`	Enters route-map configuration mode, and creates a route-map.

	Command or Action	Purpose
Step 4	**set ip next-hop in-vrf** *RiV-name* **Example:** `Router(config-route-map)# set ip next-hop in-vrf MY_RIV`	Specifies that the next hop is to be resolved in the VRF table for the specified VRF. - The RiV is configured for the VRF argument in this step.

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

What to Do Next

Proceed to the next task “Defining the Address Space and Specifying Address Resolution.”

Defining the Address Space and Configuring BGP

The configuration task described in this section sets up the BGP VPNv4 exchange so that the updates are filtered through a route-map and interesting prefixes are resolved in the VRF table. The tunnel that transports the VPN traffic across the BGP core network resides in its own address space. The RiV is specified in this configuration to direct packet forwarding and next hop resolution.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip route vrf** *riv-vrf-name o.o.o.o o.o.o.o* **tunnel** *interface-number*
4. **router bgp** *as-number*
5. **neighbor** *ip-address* | *peer-group-name* **remote-as** *as-number*
6. **neighbor** *ip-address* | *peer-group-name* **update-source** *interface-type*
7. **address-family vpnv4** [**unicast**]
8. **neighbor** *ip-address* | *peer-group-name* **activate**
9. **neighbor** *ip-address* | *peer-group-name* **route-map** *map-name* {**in** | **out**}
10. **exit-address-family**
11. **address-family ipv4** [**tunnel**]
12. **neighbor** *ip-address* | *peer-group-name* **activate**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.

Step 2	**configure** {**terminal** | **memory** | **network**}
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip route vrf riv-vrf-name 0.0.0.0 0.0.0.0 tunnel n** **Example:** `Router(config)# ip route vrf MY_RIV 0.0.0.0 0.0.0.0 tunnel 1`	Sets the packet forwarding to the Resolve-in-VRF (RiV). **Note** A 0.0.0.0 0.0.0.0 default route must be configured for the RiV. If the default route is not configured, the next hop may not be resolvable.
Step 4	**router bgp as-number** **Example:** `Router (config)# router bgp 100`	Specifies the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 5	**neighbor {ip-address | peer-group-name} remote-as as-number** **Example:** `Router(config-router)# neighbor 172.16.1.2 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 6	**neighbor {ip-address | peer-group-name} update-source interface-type** **Example:** `Router(config-router)# neighbor 172.16.1.2 update-source Loopback 0`	Specifies a specific operational interface that BGP sessions use for TCP connections.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4 unicast`	Specifies address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 8	**neighbor {ip-address | peer-group-name} activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.2 activate`	Enables the exchange of information with a neighboring router. Use the **neighbor activate** command in address family configuration or router configuration mode

	Command or Action	Purpose

Step 9	**`neighbor {ip-address | peer-group-name} route-map map-name {in | out}`**
	Example:
	`Router(config-router-af)# neighbor 172.16.1.2 route-map SELECT_UPDATE_FOR_L3VPN in`
		Applies a route map to incoming or outgoing routes. Use once for each inbound route.

Step 10	**`exit-address-family`**
	Example:
	`Router(config-router-af)# exit-address-family`
		Exits address family configuration mode, and enters router configuration mode.

Step 11	**`address-family ipv4`** `[`**`tunnel`**`]`
	Example:
	`Router(config-router)# address-family ipv4 tunnel`
		Enter address family configuration mode for the IPv4 tunnel SAFI.
		• The configuration of this SAFI allows BGP to advertise the tunnel endpoints and SAFI specific attribute (which contains the tunnel type and the tunnel capabilities) between the PE routers.
		Note Redistribution is enabled automatically within this SAFI.

Step 12	**`neighbor {ip-address | peer-group-name} activate`**
	Example:
	`Router(config-router-af)# neighbor 172.16.1.2 activate`
		Enables the exchange of information with a neighboring router. Use the **neighbor activate** command in address family configuration or router configuration mode

Step 13	**`end`**
	Example:
	`Router(config-router-af)# end`
		Exits address-family configuration mode and enters privileged EXEC mode.

Verifying the VRF and RiV

Use the following steps to verify the configuration of the VRF and RiV.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 vrf** *vrf-name*
3. **show ip route vrf** *vrf-name*
4. **show ip cef vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [ip-prefix/length [longer-prefixes] [output-modifiers]] [network-address [mask] [longer-prefixes] [output-modifiers]] [cidr-only] [community] [community-list] [dampened-paths] [filter-list] [flap-statistics] [inconsistent-as] [neighbors] [paths [line]] [peer-group] [quote-regexp] [regexp] [summary] [tags]** **Example:** `Router# show ip bgp vpnv4 vrf MY_RIV`	Displays VPN address information from the BGP table. This command is used to verify that the specified VRF has been received by BGP. The BGP table entry should show that the route-map has worked and that the next hop is showing in the RiV. If the VRF route is not in the BGP VRF, reconfigure the VRF and route distinguisher.
Step 3	**show ip route vrf vrf-name [connected] [protocol [as-number] [tag] [output-modifiers]] [ip-prefix] [list number [output-modifiers]] [profile] [static [output-modifiers]] [summary [output-modifiers]] [supernets-only [output-modifiers]] [traffic-engineering [output-modifiers]]** **Example:** `Router# show ip route vrf MY_RIV`	Displays the IP routing table associated with a VRF instance. The **show ip route vrf** command is used to verify that the VRF is in the routing table. If the VRF is in the routing table but the PE router still cannot be reached with the **ping** command, reconfigure the VRF and route distinguisher.
Step 4	**show ip cef vrf vrf-name [ip-prefix [mask [longer-prefixes]] [detail] [output-modifiers]] [interface interface-number] [adjacency] [interface interface-number] [detail] [discard] [drop] [glean] [null] [punt] [output-modifiers]] [detail [output-modifiers]] [non-recursive [detail] [output-modifiers]] [summary [output-modifiers]] [traffic [prefix-length] [output-modifiers]] [unresolved [detail] [output-modifiers]]** **Example:** `Router# show ip cef vrf MY_RIV`	Displays the CEF forwarding table associated with VRF that was configured for the VPN. This command is used to verify that the correct VRF routes are in the CEF table. If the VRF route is not in the CEF table, reconfigure the VRF and route distinguisher.

Verifying the Multipoint L2TPv3 Tunnel

Use the following steps to verify the configuration of the multipoint L2TPv3 tunnel.

SUMMARY STEPS

1. **enable**
2. **show interface** *interface*
3. **show l2tun**
4. **show tunnel endpoint** *vrf-name*
5. **show ip bgp ipv4 tunnel** [*ip-address* | **summary**]

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**`show interface interface`**
	Example:
	`Router# show interface Tunnel 1`
		Displays the information about the specified interface.
		• This command used to verify that the tunnel interface has been correctly configured and is functioning properly.

Step 3	**`show l2tun`**
	Example:
	`Router# show l2tun`
		Displays the state of L2TPv3 tunnels and currently configured sessions.
		• This command is used to verify tunnel and session information.

Step 4	**`show tunnel endpoint interface`**
	Example:
	`Router# show tunnel endpoint`
		Displays source and destination information for tunnel endpoints.
		• This command is used to verify that the tunnel endpoints have been created correctly.

Step 5	**`show ip bgp ipv4 tunnel [ip-address	summary]`** **Example:** `Router# show ip bgp ipv4 tunnel summary`	Displays “tunnel” SAFI specific information. • This command is used to verify the tunnel type, tunnel capabilities, tunnel-specific attributes, and tunnel endpoints.

Configuration Examples for MPLS VPNs over IP Tunnels

Configuring the VRF and RiV Example

The following sample configuration creates and configures the VRF and RiV:

```
ip vrf vrf-name
 rd 100:110
 route-target import 100:1000
 route-target export 100:1000
 exit
ip vrf MY_RIV
 rd 1:1
 end
```

Configuring the Multipoint L2TPv3 Tunnel Example

The following sample configuration creates and configures the L2TPv3 tunnel:

```
interface tunnel 1
 ip vrf forwarding MY_RIV
 ip-address 172.16.1.3 255.255.255.255
 tunnel source loopback 0
 tunnel mode l3vpn l2tpv3 multipoint
 end
```

Configuring a Route Map for the Layer 3 VPN Example

The following sample configuration creates an inbound route map to set the next hop to be resolved within the VRF:

```
route-map SELECT_UPDATE_FOR_L3VPN permit 10
 set ip next-hop in-vrf MY_RIV
 end
```

Defining Address Space and Configuring BGP Example

The following sample configuration defines address space for the VPN and configures BGP:

```
ip route vrf MY_RIV 0.0.0.0 0.0.0.0 tunnel 1
router bgp 100
 neighbor 172.16.1.2 remote-as 100
 neighbor 172.16.1.2 update-source Loopback 0
 address-family vpnv4 unicast
 neighbor 172.16.1.2 activate
 neighbor 172.16.1.2 route-map SELECT_UPDATE_FOR_L3VPN in
 exit-address-family
 address-family ipv4 tunnel
 neighbor 176.16.1.2 activate
 end
```

Verifying the VRF Example

Use the **show ip bgp vpnv4**, **show ip route vrf**, and **show ip cef vrf** commands to verify that VRF and RiV are configured correctly propagating to the appropriate routing and forwarding tables.

Verify that the specified VRF prefix has been received by BGP. The BGP table entry should show that the route-map has worked and that the next hop is showing in the RiV. Use the **show ip bgp vpnv4** command as shown in this example:

```
Router# show ip bgp vpnv4 vrf vrf-name 10.10.10.4
BGP routing table entry for 100:1:10.10.10.4/24, version 12
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
    172.16.1.2 in "vrf-name" from 172.16.1.2 (172.16.1.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Use the **show ip route vrf** command to confirm that the same information has been propagated to the routing table:

```
Router# show ip route vrf vrf-name 10.10.10.4
Routing entry for 10.10.10.4/24
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 172.16.1.2 00:23:07 ago
  Routing Descriptor Blocks:
  * 172.16.1.2 (vrf-name), from 172.16.1.2, 00:23:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Use the **show ip cef vrf** command to verify that the same information has been propagated to the CEF forwarding table:

```
Router# show ip cef vrf vrf-name
Prefix              Next Hop             Interface
0.0.0.0/0           attached             Tunnel1
0.0.0.0/32          receive
10.10.10.4/32       10.10.10.4           Tunnel1
172.16.1.2/32       receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive

Router# show ip cef vrf CUSTOMER_A
Prefix              Next Hop             Interface
0.0.0.0/0           drop                 Null0 (default route handler entry)
0.0.0.0/32          receive
192.168.0.0/8       10.10.10.4           Tunnel1
10.0.4.0/24         10.10.10.4           Tunnel1
10.0.6.0/24         attached             Serial2/0
10.0.6.0/32         receive
10.0.6.1/32         receive
10.0.6.255/32       receive
224.0.0.0/4         drop
224.0.0.0/24        receive
```

```
255.255.255.255/32  receive
```

Verifying the Multipoint L2TPv3 Tunnel Examples

Use the **show interface**, **show l2tun**, and **show tunnel endpoint** commands to verify the configuration of the, tunnel interface, L2TPv3 tunnel and tunnel end points.

Use the **show interface** command, as show in the display, to verify that the tunnel interface is up and configured correctly:

```
Router# show interface Tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.2/32
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.10.10.6 (Loopback0)
  Tunnel protocol/transport Multi-L2TPv3 (L3VPN), sequencing disabled
  Transporting l3vpn traffic to routes recursing through "MY_RIV"
  Key disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Use the **show l2tun** command, as shown in the display, to verify tunnel and session information:

```
Router# show l2tun
 Tunnel and Session Information Total tunnels 0 sessions 1

L3VPN Session Information Total sessions 1

LocID      Cookie
1025       0xC0DEE550DEADBEEF
```

Use the **show tunnel endpoint** command, as shown in the display to verify that the tunnel end points have been created correctly:

```
Router# show tunnel endpoint
 Tunnel1 running in Multi-L2TPv3 (L3VPN) mode
  RFC2547/L3VPN Tunnel endpoint discovery is active on Tu1
  Transporting l3vpn traffic to all routes recursing through "MY_RIV"
 Endpoint 10.10.10.4 via destination 10.10.10.4
```

Use the **show ip bgp ipv4 tunnel** command, as shown in the display to verify tunnel specific information configured under the “tunnel” SAFI:

```
Router# show ip bgp ipv4 tunnel

BGP table version is 3, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.3.3.3/32       0.0.0.0                  0         32768 ?
*>i10.5.5.5/32       10.5.5.5                 0    100      0 ?
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 401,cookie_high 6FB30F92,cookie_low A7E61105
```

Additional References

For additional information related to this feature, refer to the following references:

Related Documents

Related Topic	Document Title
VPN configuration	*Cisco IOS Dial Services Configuration Guide*, Release 12.3 and *Cisco IOS Switching Services Configuration Guide*, Release 12.3
CEF switching	*Cisco IOS Switching Services Configuration Guide*, Release 12.3
VPN Routing and Forwarding (VRF) instances	*Cisco IOS Switching Services Configuration Guide*, Release 12.3

Standards

Standard	Title
IPv4-Tunnel SAFI	*IPv4-Tunnel SAFI* http://www.ietf.org/internet-drafts/draft-nalawade-kapoor-tunnel-safi-03.txt

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents modified commands .

- **address-family ipv4 (BGP)**
- **clear ip bgp**
- **clear tunnel l3vpn l2tpv3**
- **show ip bgp ipv4**

address-family ipv4 (BGP)

To enter address family configuration mode to configure an IPv4 routing session under a BGP routing process, use the **address-family ipv4** command in router configuration mode. To disable an address family specific IPv4 routing session, use the **no** form of this command.

address-family ipv4 [**multicast** | **unicast** | **vrf** *vrf-name* | **tunnel**]

no address-family ipv4 [**multicast** | **unicast** | **vrf** *vrf-name* | **tunnel**]

Syntax Description

multicast	(Optional) Specifies an IPv4 multicast address prefix routing session.
unicast	(Optional) Specifies an IPv4 unicast address prefix routing session.
vrf *vrf-name*	(Optional) Specifies the name of the virtual routing and forwarding (VRF) instance to associate with an IPv4 routing session and subsequent address family configuration mode commands.
tunnel	(Optional) Specifies a IPv4 routing session for multipoint tunnelling.

Defaults

IP Version 4 address prefixes are not enabled. Unicast address prefixes are the default when IP Version 4 address prefixes are configured.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced. The **address-family ipv4** command replaces the **match nlri** and **set nlri** commands.
12.0(28)S	The **tunnel** keyword was integrated in Cisco IOS Release 12.0(28)S.
12.0(30)S	Support for the Cisco 12000 Series Internet Router was added.

Usage Guidelines

The **address-family ipv4** command places the router in an address family configuration mode, from which you can configure address family and subaddress family specific routing sessions that use standard IP Version 4 address prefixes. To leave an address family configuration mode and return to router configuration mode, type **exit**.

Routing information for address family IP Version 4 is advertised by default when you configure a BGP routing session using the **neighbor remote-as** command unless you enter the **no bgp default ipv4-unicast** command.

The **tunnel** keyword is used to enable the tunnel subaddress family identifier (SAFI) under the IPv4 address family identifier. This SAFI is used to advertise the tunnel endpoints and the SAFI specific attributes (which contain the tunnel type and tunnel capabilities). Redistribution of tunnel end points into the BGP IPv4 tunnel SAFI table occurs automatically when the tunnel address-family is configured. However, peers need to be activated under the tunnel address-family before the sessions can exchange tunnel information.

Examples

The following example places the router in tunnel address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 tunnel
Router(config-router-af)#
```

The following example places the router in IPv4 address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4
Router(config-router-af)#
```

The following example places the router in IPv4 multicast address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 multicast
Router(config-router-af)#
```

The following example places the router in IPv4 unicast address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)#
```

The following example places the router in address family configuration mode and specifies **cisco** as the name of the VRF instance to associate with subsequent IP Version 4 address family configuration mode commands:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf cisco
Router(config-router-af)#
```

Note Use this form of the command, which specifies a VRF, only to configure routing exchanges between provider edge (PE) and customer edge (CE) devices.

Related Commands

Command	Description
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPN Version 4 address prefixes.
neighbor activate	Enables the exchange of information with a BGP neighboring router.

clear ip bgp

To reset a BGP connection using BGP soft reconfiguration, use the **clear ip bgp** command in privileged EXEC mode.

clear ip bgp {* | *neighbor-address* | **peer-group** *peer-group-name*} [**ipv4 tunnel**] [**soft** [**in** | **out**]]

Syntax Description

*	Specifies that all current BGP sessions will be reset.
neighbor-address	Specifies that only the identified BGP neighbor will be reset.
peer-group *peer-group-name*	Specifies that only the specified BGP peer group will be reset.
ipv4 tunnel	(Optional) Specifies that only the “tunnel” SAFI IPv4 session will be reset.
soft	(Optional) Soft reset. Does not reset the session.
in | **out**	(Optional) Triggers inbound or outbound soft reconfiguration. If the **in** or **out** option is not specified, both inbound and outbound soft reset is triggered.

Defaults

No reset is initiated.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.0(6)T 12.0(2)S	The dynamic inbound soft reset capability was added.

Usage Guidelines

You can reset inbound routing table updates dynamically or by generating new updates using stored update information. Using stored update information required additional memory for storing the updates.

To reset inbound routing table updates dynamically, all BGP routers must support the route refresh capability. To determine whether a BGP router supports this capability, use the **show ip bgp neighbors** command. If a router supports the route refresh capability, the following message is displayed:

```
Received route refresh capability from peer.
```

If all BGP routers support the route refresh capability, use the **clear ip bgp** {* | *address* | *peer-group-name*} **in** command. You need not use the **soft** keyword, because soft reset is automatically assumed when the route refresh capability is supported.

To generate new inbound updates from stored update information (rather than dynamically) without resetting the BGP session, you must preconfigure the local BGP router using the **neighbor soft-reconfiguration inbound** command. This preconfiguration causes the software to store all received updates without modification regardless of whether an update is accepted by the inbound policy. Storing updates is memory intensive and should be avoided if possible.

Outbound BGP soft configuration has no memory overhead and does not require any preconfiguration. You can trigger an outbound reconfiguration on the other side of the BGP session to make the new inbound policy take effect.

Use this command whenever any of the following changes occur:

- Additions or changes to the BGP-related access lists
- Changes to BGP-related weights
- Changes to BGP-related distribution lists
- Changes to BGP-related route maps

Examples

The following example clears the inbound session with the neighbor 10.108.1.1 without resetting the session:

```
Router# clear ip bgp 10.108.1.1 soft in
```

The following example clears the outbound session with the peer group named corp without resetting the session:

```
Router# clear ip bgp peer-group corp soft out
```

Related Commands

Command	Description
neighbor soft-reconfiguration	Configures the Cisco IOS software to start storing updates.
show ip bgp	Displays entries in the BGP routing table.

clear tunnel l3vpn l2tpv3

To reset a layer 3 VPN session over a L2TPv3 tunnel, use the **clear tunnel l3vpn l2tpv3** command in privileged EXEC mode.

clear tunnel l3vpn l2tpv3 [*hold-time*]

Syntax Description

hold-time	(optional) Configures the amount of time that the existing tunnel session will remain valid, while the new session is propagated to peers. The range of the *hold-time* argument is from 1 to 59 seconds. The default value is 15 seconds.

Defaults

No reset is initiated.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(28)S	This command was integrated in Cisco IOS Release 12.0(28)S.
12.0(30)S	Support for the Cisco 12000 Series Internet Router was added.

Usage Guidelines

This command is used to generate and distribute a new L2TPv3 session for a layer 3 VPN. This command is issued on the PE router. The *hold-time* argument is used to configure the amount of time that the existing session will remain valid, while the new session is propagated to peers. The default value for the *hold-time* argument is 15 seconds. This should be enough time for most networks. However, this value can be increased if it takes longer for the new session to propagate to all other PE routers.

Examples

The following example resets the existing L2TPv3 session for a layer 3 VPN and generates a new session:

```
Router# clear tunnel l3vpn l2tpv3
```

show ip bgp ipv4

To display entries in the IP version 4 (IPv4) Border Gateway Protocol (BGP) routing table, use the **show ip bgp ipv4** command in EXEC mode.

show ip bgp ipv4 {**multicast** | **unicast** | **tunnel** [*ip-address* | **summary**]}

Syntax Description

multicast	Displays entries for multicast routes.
unicast	Displays entries for unicast routes.
tunnel	Displays entries configured under the “tunnel” SAFI.
ip-address	Displays tunnel specific information for the specified IP address.
summary	Displays a summary of all routing information configured under the “tunnel” SAFI.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Examples

The following is sample output from the **show ip bgp ipv4 unicast** command:

```
Router# show ip bgp ipv4 unicast

 BGP table version is 4, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24   172.16.10.1              0            0  300 i
*> 10.10.20.0/24   172.16.10.1              0            0  300 i
*  10.20.10.0/24   172.16.10.1              0            0  300 i
```

The following is sample output from the **show ip bgp ipv4 multicast** command:

```
Router# show ip bgp ipv4 multicast

 BGP table version is 4, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24   172.16.10.1              0            0  300 i
*> 10.10.20.0/24   172.16.10.1              0            0  300 i
*  10.20.10.0/24   172.16.10.1              0            0  300 i
```

The following is sample output from the **show ip bgp ipv4 tunnel** command:

```
Router# show ip bgp ipv4 tunnel

BGP table version is 3, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.3.3.3/32       0.0.0.0                  0         32768 ?
*>i10.5.5.5/32       10.5.5.5                  0    100      0 ?
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 401,cookie_high 6FB30F92,cookie_low A7E61105
```

The following is sample output from the **show ip bgp ipv4 summary** command:

```
Router# show ip bgp ipv4 tunnel summary
```

```
BGP router identifier 10.3.3.3, local AS number 1
BGP table version is 3, main routing table version 3
..
2 BGP SAFI-Specific-Attr entries using 80 bytes of memory
..
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.5.5.5        4     1     422     413        3    0    0 05:28:23        1
```

The following is sample output from the **show ip bgp tunnel** command when a single IP address is specified:

```
Router# show ip bgp ip tunnel 10.5.5.5
```

```
BGP routing table entry for 10.5.5.5/32, version 2
Paths: (1 available, best #1, table IPv4-Tunnel-BGP-Table)
  Not advertised to any peer
  Local
    10.5.5.5 (metric 30) from 10.5.5.5 (10.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      SAFI Specific Attribute: ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
```

Table 2 describes the significant fields shown in the display.

Table 2   show ip bgp ipv4 unicast Field Descriptions

Field	Description
BGP table version	Internal version number of the table. This number is incremented whenever the table changes.
local router ID	IP address of the router.
Status codes	Status of the table entry. The status is displayed at the beginning of each line in the table. It can be one of the following values: s—The table entry is suppressed. d—The table entry is damped. h—The table entry history. *—The table entry is valid. >—The table entry is the best entry to use for that network. i—The table entry was learned via an internal BGP (iBGP) session.

Table 2 show ip bgp ipv4 unicast Field Descriptions (continued)

Field	Description
Origin codes	Origin of the entry. The origin code is displayed at the end of each line in the table. It can be one of the following values: i—Entry originated from an Interior Gateway Protocol (IGP) and was advertised with a **network** router configuration command. e—Entry originated from an Exterior Gateway Protocol (EGP). ?—Origin of the path is not clear. Usually, this is a router that is redistributed into BGP from an IGP.
Network	IP address of a network entity.
Next Hop	IP address of the next system that is used when forwarding a packet to the destination network. An entry of 0.0.0.0 indicates that the router has some non-BGP routes to this network.
Metric	If shown, the value of the interautonomous system metric.
LocPrf	Local preference value as set with the **set local-preference** route-map configuration command. The default value is 100.
Weight	Weight of the route as set via autonomous system filters.
Path	Autonomous system paths to the destination network. There can be one entry in this field for each autonomous system in the path.

Related Commands

Command	Description
show ip bgp	Displays entries in the BGP routing table.