MPLS Multilink PPP Support

Note This chapter is not applicable on the ASR 900 RSP3 Module for the Cisco IOS XE Release 3.16.

The MPLS Multilink PPP Support feature ensures that MPLS Layer 3 Virtual Private Networks (VPNs) with quality of service (QoS) can be enabled for bundled links. This feature supports Multiprotocol Label Switching (MPLS) over Multilink PPP (MLP) links in the edge (provider edge [PE]-to-customer edge [CE]) or in the MPLS core (PE-to-PE and PE-to-provider [P] device).

Service providers that use relatively low-speed links can use MLP to spread traffic across them in their MPLS networks. Link fragmentation and interleaving (LFI) should be deployed in the CE-to-PE link for efficiency, where traffic uses a lower link bandwidth (less than 768 kbps). The MPLS Multilink PPP Support feature can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

Prerequisites for MPLS Multilink PPP Support

- Multiprotocol Label Switching (MPLS) must be enabled on provider edge (PE) and provider (P) devices

Restrictions for MPLS Multilink PPP Support

- Only 168 multilink bundles can be created per the OC-3 interface module on the router.
- The maximum number of members per multilink bundle is 16.
- Links in multilink bundles must be on the same interface module.
- On the 8 T1/E1, a maximum of 8 bundles can be supported.
- On the 16T1/E1, a maximum of 16 bundles can be supported.

- On the 32 T1/E1, a maximum of 32 bundles can be supported.

For information on how to configure, Protocol-Field-Compression (PFC) and Address-and-Control-Field-Compression (AFC), see *Configuring PPP and Multilink PPP on the Cisco ASR 903 Router*.

Information About MPLS Multilink PPP Support

MPLS Layer 3 Virtual Private Network Features Supported for Multilink PPP

The table below lists Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 1: MPLS Layer 3 VPN Features Supported for MLP

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Static routes	Supported	Not supported	Not supported
External Border Gateway Protocol (eBGP)	Supported	Not applicable to this configuration	Supported
Intermediate System-to-Intermediate System (IS-IS)	Not supported	Supported	Not supported
Open Shortest Path First (OSPF)	Supported	Supported	Not supported
Enhanced Interior Gateway Routing Protocol (EIGRP)	Supported	Supported	Not supported
Interprovider interautonomous (Inter-AS) VPNs (with Label Distribution Protocol [LDP])	Not applicable to this configuration	Supported (MLP between Autonomous System Boundary Routers [ASBRs])	Not applicable to this configuration
Inter-AS VPNs with IPv4 Label Distribution	Not applicable to this configuration	Supported (MLP between ASBRs)	Not applicable to this configuration
CSC VPNs (with LDP)	Not supported	Not applicable to this configuration	Supported
CSC VPNs with IPv4 label distribution	Supported	Not applicable to this configuration	Supported
External and internal BGP (eiBGP) Multipath	Not supported	Not supported	Not applicable to this configuration
Internal BGP (iBGP) Multipath	Not applicable to this configuration	Not supported	Not applicable to this configuration

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
eBGP Multipath	Not supported	Not supported	Not supported

MPLS Quality of Service Features Supported for Multilink PPP

The table below lists the Multiprotocol Label Switching (MPLS) quality of service (QoS) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 2: MPLS QoS Features Supported for MLP

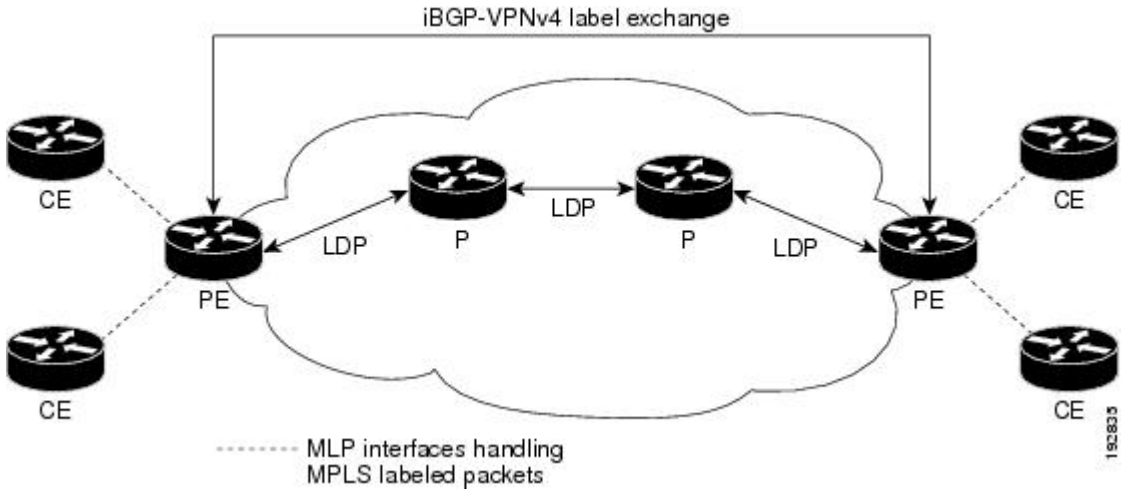
MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Default copy of IP Precedence to EXP bits and the reverse	Supported	Not supported	Not supported
Set MPLS EXP bits using the modular QoS Command-Line Interface (MQC)	Supported	Supported	Supported
Matching on MPLS EXP using MQC	Supported	Supported	Supported
Low Latency Queueing (LLQ)/Class-Based Weighted Fair Queueing (CBWFQ) support	Supported	Supported	Supported
Weighted Random Early Detection (WRED) based on EXP bits using MQC	Supported	Supported	Supported
Policer with EXP bit-marking using MQC-3 action	Supported	Supported	Supported
Support for EXP bits in MPLS accounting	Supported	Supported	Supported

MPLS Multilink PPP Support and PE-to-CE Links

The figure below shows a typical Multiprotocol Label Switching (MPLS) network in which the provider edge (PE) device is responsible for label imposition (at ingress) and disposition (at egress) of the MPLS traffic.

In this topology, Multilink PPP (MLP) is deployed on the PE-to-customer edge (CE) links. The Virtual Private Network (VPN) routing and forwarding instance (VRF) interface is in a multilink bundle. There is no MPLS interaction with MLP; all packets coming into the MLP bundle are IP packets.

Figure 1: MLP and Traditional PE-to-CE Links



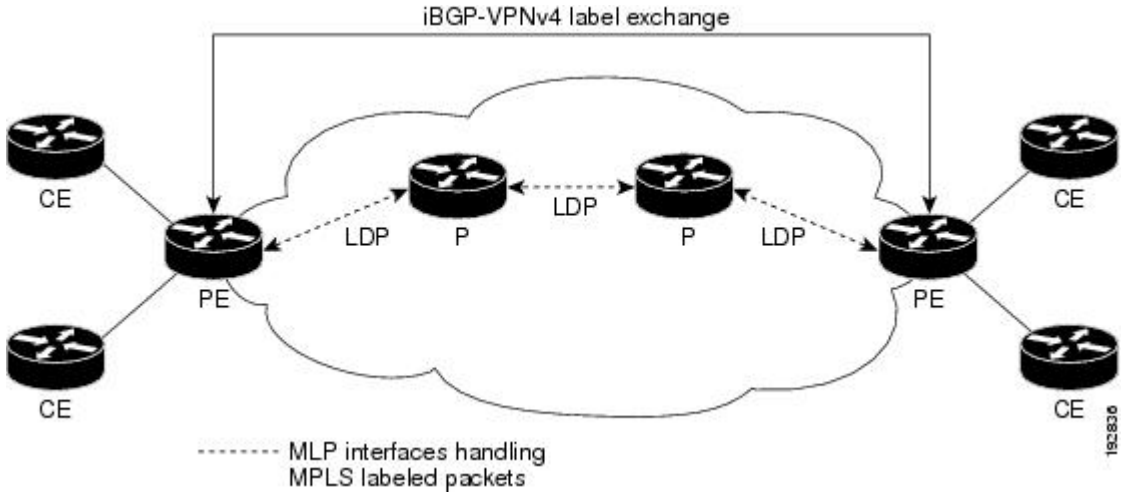
The PE-to-CE routing protocols that are supported for the MPLS Multilink PPP Support feature are external Border Gateway Protocol (eBGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). Static routes are also supported between the CE and PE devices.

Quality of service (QoS) features that are supported for the MPLS Multilink PPP Support feature on CE-to-PE links are link fragmentation and interleaving (LFI), compressed Real-Time Transport Protocol (cRTP), policing, marking, and classification.

MPLS Multilink PPP Support and Core Links

The figure below shows a sample topology in which Multiprotocol Label Switching (MPLS) is deployed over Multilink PPP (MLP) on provider edge-to-provider (PE-to-P) and P-to-P links. Enabling MPLS on MLP for PE-to-P links is similar to enabling MPLS on MLP for P-to-P links.

Figure 2: MLP on PE-to-P and P-to-P Links



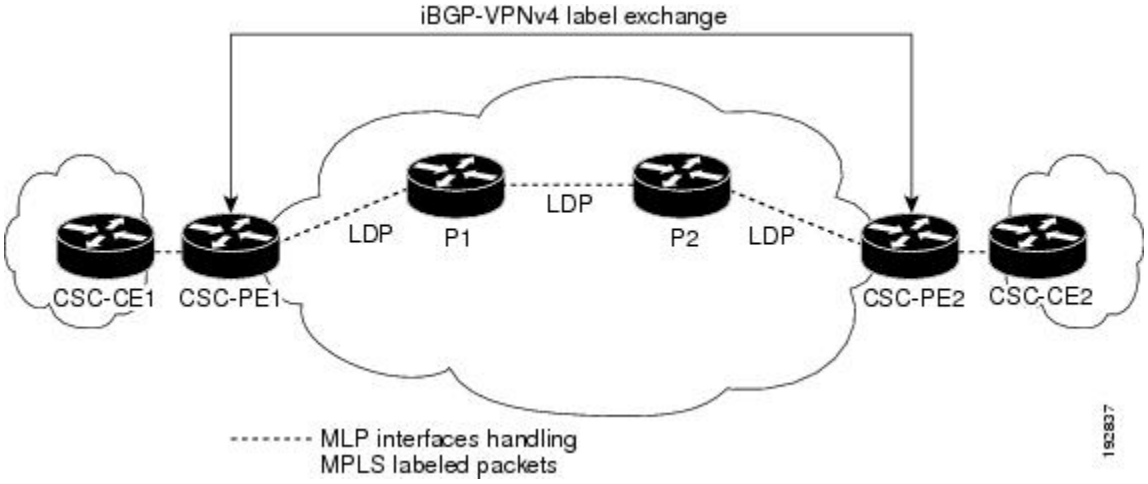
You employ MLP in the PE-to-P or P-to-P links primarily so that you can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate the load sharing of traffic.

In addition to requiring MLP on the PE-to-P links, the MPLS Multilink PPP Support feature requires the configuration of an IGP routing protocol and the Label Distribution Protocol (LDP).

MPLS Multilink PPP Support in a CSC Network

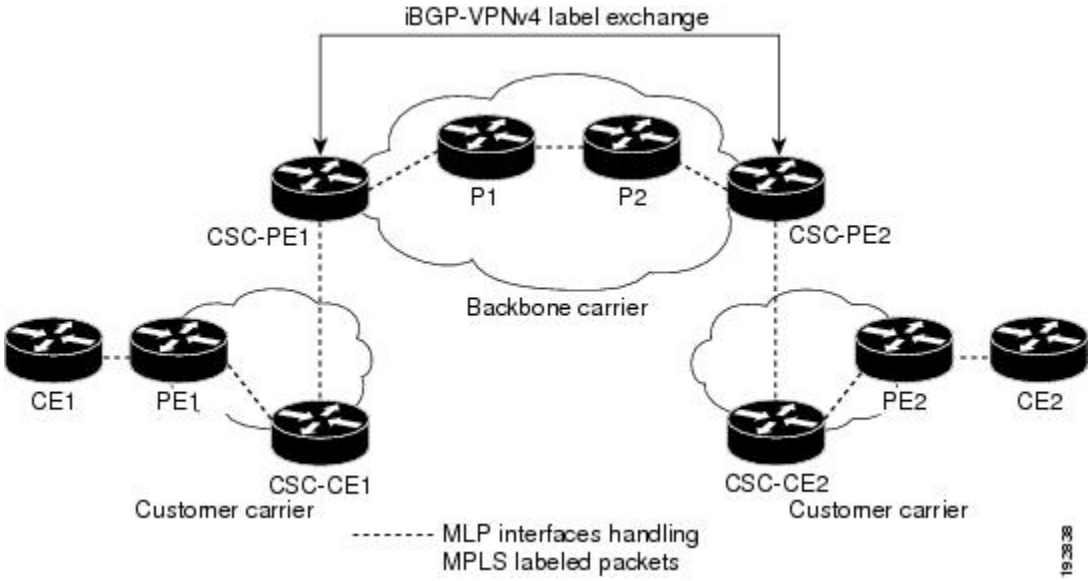
The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) network where Multilink PPP (MLP) is configured on the CSC customer edge (CE)-to-provider edge (PE) links.

Figure 3: MLP on CSC CE-to-PE Links with MPLS VPN Carrier Supporting Carrier



The MPLS Multilink PPP Support feature supports MLP between CSC-CE and CSC-PE links with the Label Distribution Protocol (LDP) or with external Border Gateway Protocol (eBGP) IPv4 label distribution. This feature also supports link fragmentation and interleaving (LFI) for an MPLS VPN CSC configuration. The figure below shows all MLP links that this feature supports for CSC configurations.

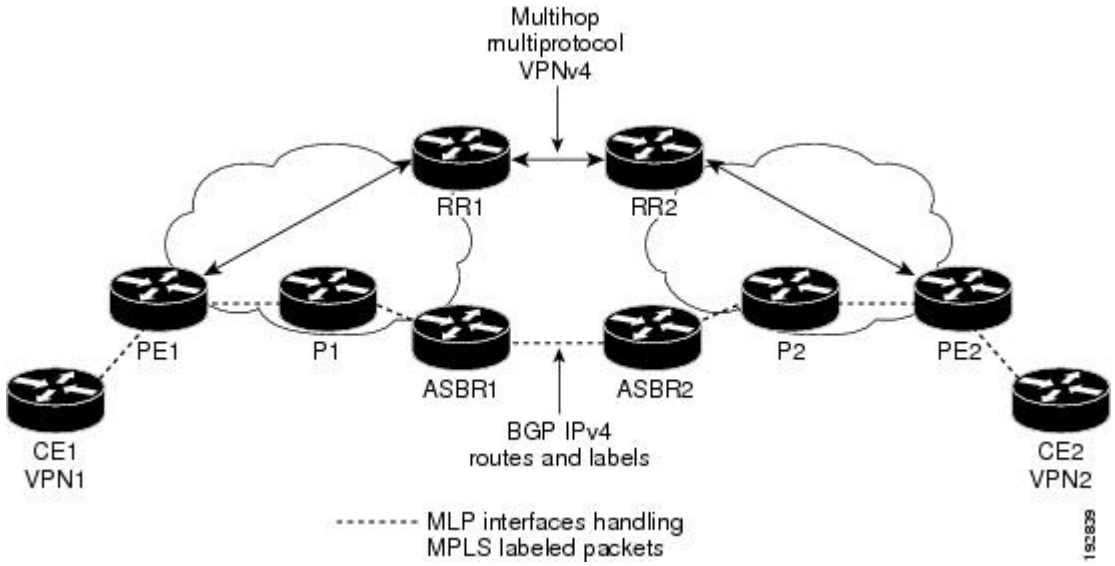
Figure 4: MLP Supported Links with MPLS VPN Carrier Supporting Carrier



MPLS Multilink PPP Support in an Interautonomous System

The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) network where Multilink PPP (MLP) is configured on the provider edge-to-customer edge (PE-to-CE) links.

Figure 5: MLP on ASBR-to-PE Links in an MPLS VPN Inter-AS Network



The MPLS Multilink PPP Support feature supports MLP between Autonomous System Boundary Router (ASBR) links for Inter-AS VPNs with Label Distribution Protocol (LDP) and with external Border Gateway Protocol (eBGP) IPv4 label distribution.

How to Configure MPLS Multilink PPP Support

The tasks in this section can be performed on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, P-to-P links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Creating a Multilink Bundle

Perform this task to create a multilink bundle for the MPLS Multilink PPP Support feature. This multilink bundle can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask* [**secondary**]
5. **encapsulation** *encapsulation-type*
6. **ppp multilink**

7. **mpls ip**
8. **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	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 1`	Creates a multilink bundle and enters multilink interface configuration mode. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 4	**ip address** *address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.0 255.255.0.0`	Sets a primary or secondary IP address for an interface. • The *address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address. This command is used to assign an IP address to the multilink interface.
Step 5	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method as PPP to be used by the interface. • The *encapsulation-type* argument specifies the encapsulation type.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on an interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables label switching on the interface.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Assigning an Interface to a Multilink Bundle

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller** {**t1** | **e1**} *slot/port*
4. **channel-group** *channel-number* **timeslots** *fulltimeslots*
5. **exit**
6. **interface serial** *slot/subslot* / *port* **:** *channel-group*
7. **ip route-cache** [**cef** | **distributed**]
8. **no ip address**
9. **keepalive** [*period* [*retries*]]
10. **encapsulation** *encapsulation-type*
11. **ppp multilink group** *group-number*
12. **ppp multilink**
13. **ppp authentication chap**
14. **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	**controller** {**t1** | **e1**} *slot/port* **Example:** `Device# controller t1 0/0/1`	Configures a T1 or E1 controller and enters controller configuration mode. • The **t1** keyword indicates a T1 line card. • The **e1** keyword indicates an E1 line card. • The *slot/port* arguments are the backplane slot number and port number on the interface. Refer to your

	Command or Action	Purpose
		hardware installation manual for the specific slot numbers and port numbers.
Step 4	**channel-group** *channel-number* **timeslots** *fulltimeslots* **Example:** `Device(config-controller)# channel-group 1 timeslots 1-24`	Defines the time slots that belong to each T1 or E1 circuit. • The *channel-number* argument is the channel-group number. When a T1 data line is configured, channel-group numbers can be values from 1 to 24. When an E1 data line is configured, channel-group numbers can be values from 1 to 31. • The **timeslots** *fulltimeslots* keyword and argument specifies time slots. For a T1 controller, the time slot is 1-24. For an E1 controller the time slot is 1-31.
Step 5	**exit** **Example:** `Device(config-controller)# exit`	Returns to global configuration mode.
Step 6	**interface serial** *slot/subslot* / *port* **:** *channel-group* **Example:** `Device(config)# interface serial 0/0/1:1`	Configures a serial interface for a Cisco 7500 series router with channelized T1 or E1 and enters interface configuration mode. • The *slot* argument indicates the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument indicates the port number. Refer to the appropriate hardware manual for slot and port information. • The *:channel-group* argument indicates the channel group number. Cisco 7500 series routers specify the channel group number in the range of 0 to 4 defined with the **channel-group** controller configuration command.
Step 7	**ip route-cache** [**cef** | **distributed**] **Example:** `Device(config-if)# ip route-cache cef`	Controls the use of switching methods for forwarding IP packets. • The **cef** keyword enables Cisco Express Forwarding operation on an interface after Cisco Express Forwarding operation was disabled. • The **distributed** keyword enables distributed switching on the interface.
Step 8	**no ip address** **Example:**	Removes any specified IP address.

	Command or Action	Purpose
	`Device(config-if)# no ip address`	
Step 9	**keepalive** [*period* [*retries*]] **Example:** `Device(config-if)# keepalive`	Enables keepalive packets and specifies the number of times that the Cisco software tries to send keepalive packets without a response before bringing down the interface or before bringing the tunnel protocol down for a specific interface. • The *period* argument is an integer value, in seconds, greater than 0. The default is 10. • The *retries* argument specifies the number of times that the device continues to send keepalive packets without a response before bringing the interface down. Enter an integer value greater than 1 and less than 255. If you do not enter a value, the value that was previously set is used; if no value was specified previously, the default of 5 is used. If you are using this command with a tunnel interface, the command specifies the number of times that the device continues to send keepalive packets without a response before bringing the tunnel interface protocol down.
Step 10	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface. • The *encapsulation-type* argument specifies the encapsulation type. The example specifies PPP encapsulation.
Step 11	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 1`	Restricts a physical link to join only one designated multilink group interface. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 12	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on the interface.
Step 13	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication on the serial interface.
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the Multilink PPP Configuration

SUMMARY STEPS

1. **enable**
2. **show ip interface brief**
3. **show ppp multilink**
4. **show ppp multilink interface** *interface-bundle*
5. **show interface** *type number*
6. **show mpls forwarding-table**
7. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show ip interface brief**

Verifies logical and physical Multilink PPP (MLP) interfaces.

Example:

```
Device# show ip interface brief

Locolrface                  IP-Address      OK? Method Status                Prot
GigabitEthernet1/0/0        10.3.62.106      YES NVRAM  up                     up
GigabitEthernet0/0/1        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/0        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/1        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/2        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/0        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/1        unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/2        unassigned      YES NVRAM  administratively down down
Serial0/1/0:1              unassigned      YES NVRAM  administratively down down
Serial0/1/0:2              unassigned      YES NVRAM  administratively down down
Serial0/1/1:1              unassigned      YES NVRAM  up                    up
Serial0/1/1:2              unassigned      YES NVRAM  up                    down
Serial0/1/3:1              unassigned      YES NVRAM  up                    up
Serial0/1/3:2              unassigned      YES NVRAM  up                    up
Multilink6                 10.30.0.2       YES NVRAM  up                    up
Multilink8                 unassigned      YES NVRAM  administratively down down
Multilink10                10.34.0.2       YES NVRAM  up                    up
Loopback0                  10.0.0.1        YES NVRAM  up                    up
```

Step 3 **show ppp multilink**

Verifies that you have created a multilink bundle.

Example:

```
Device# show ppp multilink

Multilink1, bundle name is group 1
 Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
 Serial0/0/0/:1
 Serial0/0/0/:2
 Serial0/0/0/:3
 Serial0/0/0/:4
```

Step 4 **show ppp multilink interface** *interface-bundle*

Displays information about a specific MLP interface.

Example:

```
Device# show ppp multilink interface multilink6

Multilink6, bundle name is router
  Bundle up for 00:42:46, 1/255 load
  Receive buffer limit 24384 bytes, frag timeout 1524 ms
  Bundle is Distributed
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 48 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x4D7 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se0/1/3:1, since 00:42:46, 240 weight, 232 frag size
    Se0/1/3:2, since 00:42:46, 240 weight, 232 frag size
```

Step 5 **show interface** *type number*

Displays information about serial interfaces in your configuration.

Example:

```
Device# show interface serial 0/1/3:1

Serial0/1/3:1 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters 00:47:13
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     722 packets input, 54323 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
     697 packets output, 51888 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:1, subrate: 64Kb/s, transmit delay is 0 flags
  Transmit queue length 25
```

```
Device# show interface serial 0/1/3:2

Serial0/1/3:2 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:47:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     725 packets input, 54618 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
     693 packets output, 53180 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:2, subrate: 64Kb/s, transmit delay is 0 flags
  Transmit queue length 26
```

You can also use the **show interface** command to display information about the multilink interface:

Example:

```
Device# show interface multilink6

Multilink6 is up, line protocol is up
  Hardware is multilink group interface
  Internet address is 10.30.0.2/8
  MTU 1500 bytes, BW 128 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open
  Open: CDPCP, IPCP, TAGCP, loopback not set
  DTR is pulsed for 2 seconds on reset
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:48:43
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     1340 packets input, 102245 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
     1283 packets output, 101350 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Step 6 **show mpls forwarding-table**

Displays contents of the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB). Look for information on multilink interfaces associated with a point2point next hop.

Example:

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
```

```
tag    tag or VC   or Tunnel Id      switched   interface
16     Untagged    10.30.0.1/32      0          Mu6        point2point
17     Pop tag     10.0.0.3/32       0          Mu6        point2point
18     Untagged    10.0.0.9/32[V]    0          Mu10       point2point
19     Untagged    10.0.0.11/32[V]   6890       Mu10       point2point
20     Untagged    10.32.0.0/8[V]    530        Mu10       point2point
21     Aggregate   10.34.0.0/8[V]    0
22     Untagged    10.34.0.1/32[V]   0          Mu10       point2point
```

Use the **show ip bgp vpnv4** command to display VPN address information from the Border Gateway Protocol (BGP) table.

Example:

```
Device# show ip bgp vpnv4 all summary

BGP router identifier 10.0.0.1, local AS number 100
BGP table version is 21, main routing table version 21
10 network entries using 1210 bytes of memory
10 path entries using 640 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1994 total bytes of memory
BGP activity 10/0 prefixes, 10/0 paths, scan interval 5 secs
10.0.0.3  4 100 MsgRc52 MsgSe52  TblV21  0  0 00:46:35 State/P5xRcd
```

Step 7 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS Multilink PPP Support

Sample MPLS Multilink PPP Support Configurations

The following examples show sample configurations on a Carrier Supporting Carrier (CSC) network. The configuration of MLP on an interface is the same for provider edge-to-customer edge (PE-to-CE) links, PE-to-provider (P) links, and P-to-P links.

Example: Configuring Multilink PPP on an MPLS CSC PE Device

The following example shows how to configure for Multiprotocol Label Switching (MPLS) Carrier Supporting Carrier (CSC) provider edge (PE) device.

```
!
mpls label protocol ldp
ip cef
ip vrf vpn2
```

```
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!

controller T1 0/0/1
framing esf
clock source internal
linecode b8zs
channel-group 1 timeslots 1-24
!
interface Serial0/0:1
 no ip address
 encapsulation ppp

 ppp multilink
 ppp multilink group 1

interface Multilink1
 ip vrf forwarding vpn2
 ip address 10.35.0.2 255.0.0.0
 no peer neighbor-route
 load-interval 30
 ppp multilink
 ppp multilink interleave
 ppp multilink group 1
mpls ip
mpls label protocol ldp
!
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Multilink1
 network 10.0.0.7 0.0.0.0 area 200
 network 10.31.0.0 0.255.255.255 area 200
!
!
 router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.0.0.11 remote-as 200
 neighbor 10.0.0.11 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.0.0.11 activate
 neighbor 10.0.0.11 send-community extended
 bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.35.0.1 remote-as 300
 neighbor 10.35.0.1 activate
 neighbor 10.35.0.1 as-override
 neighbor 10.35.0.1 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

Example: Creating a Multilink Bundle

The following example shows how to create a multilink bundle for the MPLS Multilink PPP Support feature:

```
Device(config)# interface multilink 1
Device(config-if)# ip address 10.0.0.0 10.255.255.255
Device(config-if)# encapsulation ppp
Device(config-if)# ppp chap hostname group 1
Device(config-if)# ppp multilink
Device(config-if)# ppp multilink group 1
Device(config-if)# mpls ip
Device(config-if)# mpls label protocol ldp
```

Example: Assigning an Interface to a Multilink Bundle

The following example shows how to create four multilink interfaces with Cisco Express Forwarding switching and Multilink PPP (MLP) enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
mpls ip
mpls label protocol ldp

interface serial 0/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```