Example 6-35 shows the configuration of the MPLS TE tunnel interface.

Example 6-35 *Configuration of the MPLS TE Tunnel Interface*

```
interface Tunnel10
 ip unnumbered Loopback0
 no ip directed-broadcast
 mpls ip
 tunnel destination 10.1.1.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng bandwidth 256
 tunnel mpls traffic-eng path-option 5 dynamic
```

Highlighted line 1 shows the key difference in the configuration of a MPLS TE tunnel between P routers. The command **mpls ip** enables LDP (or TDP) on the tunnel interface. This is crucial if VPN traffic is not to be dropped by the tail-end router.

Note that although it is not strictly required, you may also want to configure the **mpls ip** command on TE tunnels between PE routers—it cannot hurt, and you never know when it might become essential because of a network topology change.

Refer to the previous section for an explanation of other commands configured in Example 6-35.

NOTE For more information regarding MPLS TE, see *Traffic Engineering with MPLS* by Eric Osborne and Ajay Simha (Cisco Press).

Troubleshooting MPLS VPNs

MPLS VPNs are relatively complex, but by adopting an end-to-end, step-by-step approach, troubleshooting can be relatively fast and efficient. The process of troubleshooting MPLS VPNs can be broken down into two basic elements, troubleshooting route advertisement between the customer sites, and troubleshooting the LSP across the provider backbone.

The flowcharts in Figure 6-29 and 6-30 describe the processes used for troubleshooting route advertisement between the customer sites and troubleshooting the LSPs across the provider backbone. You can use these flowcharts to quickly access the section of the chapter relevant to problems you are experiencing on your network.

Figure 6-29 *Flowchart for Troubleshooting Route Advertisement Between the Customer Sites in an MPLS VPN*

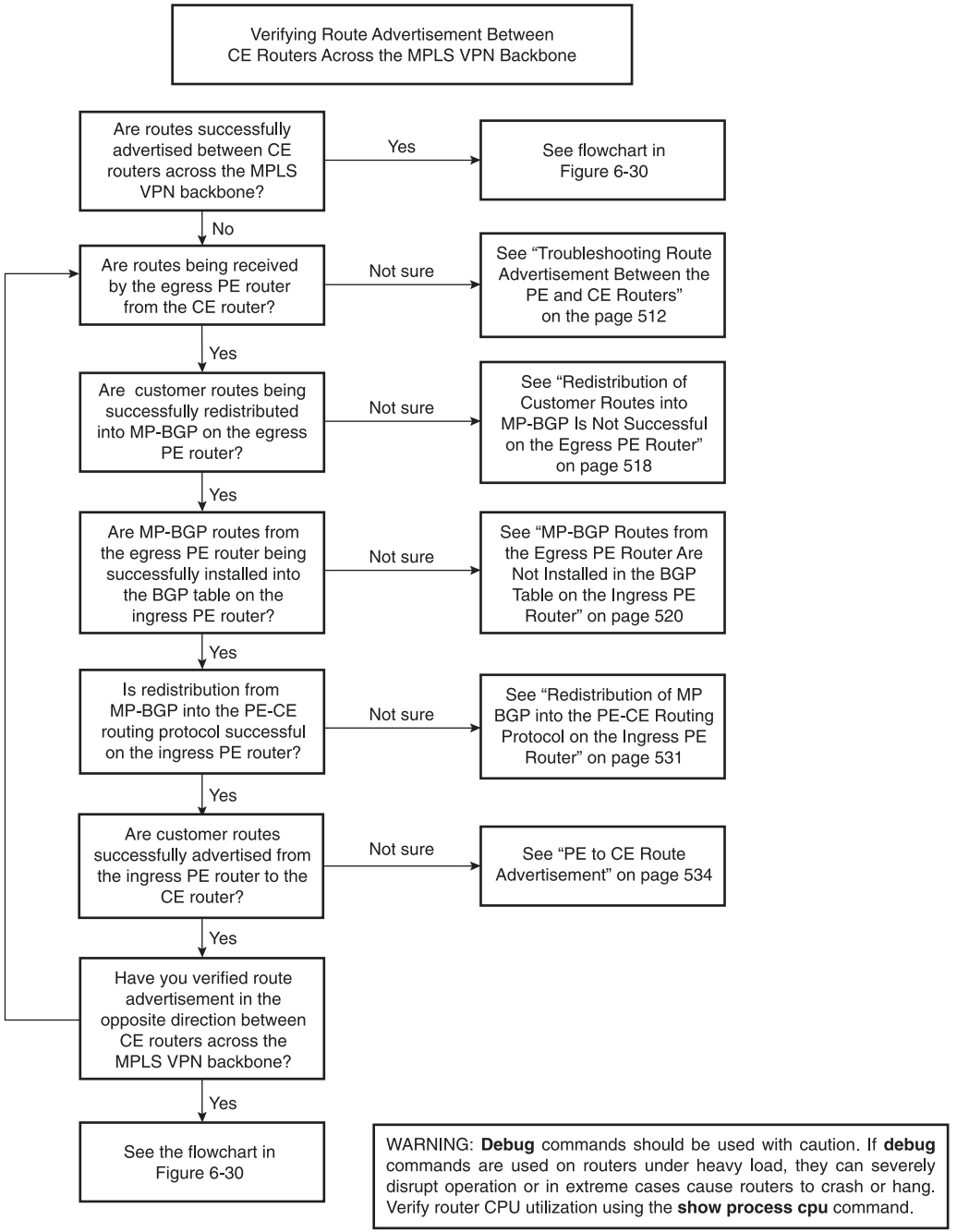
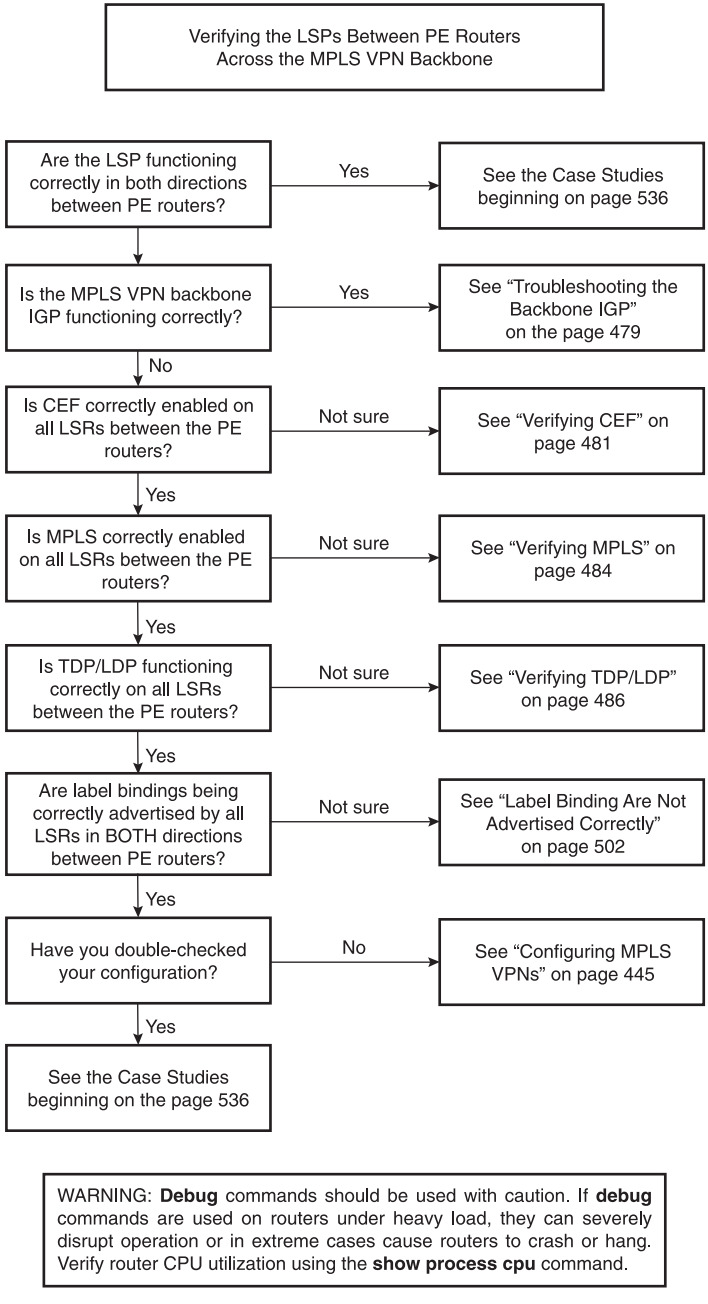


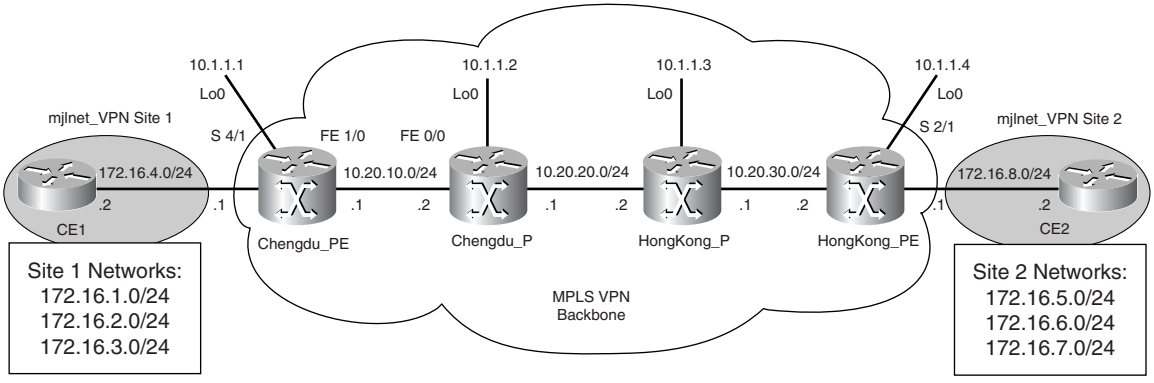
Figure 6-30 *Flowchart for Troubleshooting the LSPs Across the Provider MPLS Backbone*



These two MPLS VPN troubleshooting elements are discussed in the sections that follow. Before diving in, however, it is a good idea to try to locate the issue using the **ping** and **traceroute** commands.

The sample topology is used as a reference throughout this section is illustrated in Figure 6-31.

Figure 6-31 *Sample MPLS VPN Topology*



Newer Cisco IOS software commands (such as **show mpls ldp bindings**) are used in the sections that follow. Table 6-2 at the end of the chapter shows newer commands and their older equivalents (such as **show tag-switching tdp bindings**). Note, however, that almost without exception, older commands use the **tag-switching** keyword in place of the **mpls** keyword, and the **tdp** keyword in place of the **ldp** keyword.

Locating the Problem in an MPLS VPN

Two commands that are particularly good for locating problems in the MPLS VPN are **ping** and **traceroute**.

The **ping** command can be used to give you are general idea of the location the problem. The **ping** command can be used to verify both the LSP and route advertisement across the MPLS VPN backbone.

The **traceroute** command, on the other hand, can be used for a more detailed examination of the LSP.

Note that if you are using IOS 12.0(27)S or above, you can also take advantage of the **ping mpls** and **trace mpls** MPLS Embedded Management feature commands to test LSP connectivity and trace LSPs respectively. These commands use MPLS echo request and reply packets ([labelled] UDP packets on port 3503), and allow you to specify a range of options *including* datagram size, sweep size range, TTL (maximum number of hops), MPLS echo request timeouts, MPLS echo request intervals, and Experimental bit settings.

Verifying IP Connectivity Across the MPLS VPN

As previously mentioned, the **ping** command can be useful in locating problems in the MPLS VPN. Two tests that can be very useful are to ping from the PE router to the connected CE router, and from the ingress PE router to the egress PE router.

Can You Ping from the PE to the Connected CE?

The first step in verifying IP connectivity across the MPLS VPN is to check whether you can ping from both the ingress and egress PE routers to their respective connected CE routers. Do not forget to specify the VRF when pinging the CE router.

Example 6-36 shows a **ping** test from the PE router (Chengdu_PE) to the connected CE router (CE2).

Example 6-36 *Pinging the Connected CE Router*

```
Chengdu_PE#ping vrf mjlnet_VPN 172.16.4.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.8.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 148/148/152 ms
Chengdu_PE#
```

If the ping is not successful, there may be a problem with the configuration of the VRF interface, the configuration of the connected CE router, or the PE-CE attachment circuit.

Can You Ping from the Ingress PE to the Egress PE (Globally and in the VRF)?

If you are able to ping from the PE router to the attached CE router, you should now try pinging between the ingress and egress PE routers' BGP update sources (typically loopback interfaces), as shown in Example 6-37.

Example 6-37 *Pinging Between the Ingress the Egress Routers' BGP Update Sources*

```
Chengdu_PE#ping
Protocol [ip]:
Target IP address: 10.1.1.4
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/90/92 ms
Chengdu_PE#
```

If the ping is not successful, there might be a problem with the backbone IGP, or the ingress or egress router's BGP update source is not being advertised into the backbone IGP.

If you are able to ping between the ingress and egress PE routers' BGP update sources, try pinging from the VRF interface on the ingress PE to the VRF interface on the egress PE router.

Example 6-38 shows the output of a **ping** from the VRF interface on the ingress PE router to the VRF interface on the egress PE router.

Example 6-38 *Pinging the VRF Interface on the Egress PE Router*

```
Chengdu_PE#ping vrf mjlnet_VPN 172.16.8.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.8.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/90/92 ms
Chengdu_PE#
```

If the ping is not successful, it may indicate a problem with the VRF interface on either the ingress or egress PE router; it might indicate a problem with the LSP between the ingress and egress PE routers; or it might indicate a problem with the advertisement of customer VPN routes across the MPLS VPN backbone from the egress PE router to the ingress PE router.

Using **traceroute** to Verify the LSP

One very useful tool for verifying MPLS LSPs is the **traceroute** command.

When using **traceroute** on a PE or P router, the label stack used for packet forwarding is displayed.

Global **traceroute** can be used to trace an LSP across the MPLS backbone from the ingress to the egress PE router.

In Example 6-39, the LSP is traced from the ingress PE (Chengdu_PE) to the egress PE (HongKong_PE).

Example 6-39 *Tracing the LSP from the Ingress PE to the Egress PE Router*

```
Chengdu_PE#traceroute 10.1.1.4
Type escape sequence to abort.
Tracing the route to 10.1.1.4
  1 10.20.10.2 [MPLS: Label 20 Exp 0] 48 msec 48 msec 228 msec
  2 10.20.20.2 [MPLS: Label 17 Exp 0] 32 msec 32 msec 32 msec
  3 10.20.30.2 16 msec 16 msec *
Chengdu_PE#
```

Highlighted line 1 shows that ingress PE router Chengdu_PE imposes IGP label 20 on the packet and forwards it to Chengdu_P (10.20.10.2).

In highlighted line 2, Chengdu_P swaps label 20 for label 17, and the packet transits the link to HongKong_P (10.20.20.2).

In highlighted line 3, HongKong_P pops the label and forwards the unlabeled packet to egress PE router HongKong_PE (10.20.30.2).

VRF **traceroute** can used to examine a labeled VPN packet as it crosses the MPLS backbone from the mjlnet_VPN VRF interface of the ingress PE router to mjlnet_VPN site 2, as shown in Example 6-40.

Example 6-40 *VRF* **traceroute** *from the VRF Interface on the Ingress PE Router to mjlnet_VPN Site 2*

```
Chengdu_PE#traceroute vrf mjlnet_VPN 172.16.8.2
Type escape sequence to abort.
Tracing the route to 172.16.8.2
  1 10.20.10.2 [MPLS: Labels 20/23 Exp 0] 96 msec 96 msec 96 msec
  2 10.20.20.2 [MPLS: Labels 17/23 Exp 0] 80 msec 80 msec 80 msec
  3 172.16.8.1 [MPLS: Label 23 Exp 0] 76 msec 76 msec 76 msec
  4 172.16.8.2 36 msec 136 msec *
Chengdu_PE#
```

Highlighted line 1 shows that ingress PE router Chengdu_PE imposes IGP label 20, plus VPN label 23, on the packet and forwards it to Chengdu_P (10.20.10.2).

In highlighted line 2, Chengdu_P swaps IGP label 20 for label 17, and the packet transits the link to HongKong_P (10.20.20.2). Note that the VPN label (23) remains unchanged.

In highlighted line 3, HongKong_P pops the IGP label and forwards the packet to egress PE router HongKong_PE (172.16.8.1, its mjlnet_VPN VRF interface address). Again, the VPN label remains unchanged.

Finally, in highlighted line 4, egress PE router HongKong_PE removes the VPN label and forwards the unlabeled packet to the CE router (CE2, 172.16.8.2).

TIP If the **no mpls ip propagate-ttl** command is configured on the ingress PE, the MPLS backbone will be represented as 1 hop when tracing from the CE or PE routers. To allow the TTL to be propagated in traceroute on PE routers, the **mpls ip propagate-ttl local** command can be used.

Troubleshooting the Backbone IGP

Although in-depth troubleshooting of the backbone IGP is beyond the scope of this book, basic issues that will prevent correct operation of both OSPF and IS-IS are briefly discussed here.

Note that the troubleshooting steps for OSPF and IS-IS discussed here are generic in nature; they are equally applicable in a regular IP (non-MPLS) backbone.

Routing Protocol Is Not Enabled on an Interface

Check that OSPF or IS-IS is enabled on the interface using the **show ip ospf interface** or **show clns interface** commands.

Routers Are Not on a Common Subnet

Ensure that neighboring routers are configured on the same IP subnet.

Use the **show ip interface** command to verify interface IP address and mask configuration.

Passive Interface Is Configured

Ensure that an interface that should be transmitting OSPF or IS-IS packets is not configured as a passive interface.

Use the **show ip protocols** command to verify interface configuration.

Area Mismatch Exists

Ensure that areas are correctly configured on OSPF or IS-IS routers.

Check the OSPF area ID using the **show ip ospf interface** command.

Check that the IS-IS area is correctly configured using the **show clns protocol** command.

Network Type Mismatch Exists

Verify that there is not a network type mismatch between the interfaces of neighboring routers.

Use the **show ip opsf interface** command to verify the OSPF network type. Ensure that neighboring routers are configured with a consistent network type.

Use the **show running-config** command to check whether there is a network type mismatch between IS-IS routers. If IS-IS is configured on a point-to-point subinterface on one router, but a multipoint interface on the neighboring router, adjacency will fail.

Timer Mismatch Exists

Verify that there is not an OSPF or IS-IS timer mismatch between neighboring routers.

Use the **show ip ospf interface** command to check that hello and dead intervals are consistent between neighboring OSPF routers.

Use the **show running-config** command to check the configuration of the hello interval and hello multiplier timers on IS-IS routers.

Authentication Mismatch Exists

Check to see whether there is an authentication mismatch between the routers.

Use the **debug ip ospf adj** command to troubleshoot OSPF authentication issues.

Use the **debug isis adj-packets** command to troubleshoot IS-IS authentication issues.

General Misconfiguration Issues

Check the section "Step 6: Configure the MPLS VPN Backbone IGP" on page 449 to ensure that the backbone IGP is correctly configured.

NOTE For more information on the configuration of OSPF and IS-IS, see *Routing TCP/IP, Volume I (CCIE Professional Development)* by Jeff Doyle (Cisco Press).

Troubleshooting the LSP

Customer VPN traffic uses an LSP to transit the service provider backbone between the ingress and egress PE routers. When troubleshooting the LSP, you should verify correct operation of CEF, MPLS, and TDP/LDP on all LSRs along the path.

Verifying CEF

If CEF switching is not enabled on all MPLS backbone routers, label switching will not function.

In this section, you will see how to verify that CEF is enabled globally and on an interface.

CEF Is Globally Disabled

To verify that CEF switching is globally enabled on a router, use the **show ip cef** command, as demonstrated in Example 6-41.

Example 6-41 *Verifying CEF Using the* **show ip cef** *Command (CEF Is Disabled)*

```
Chengdu_P#show ip cef
%CEF not running
Prefix              Next Hop            Interface
Chengdu_P#
```

Highlighted line 1 shows that CEF is not enabled on Chengdu_P.

To enable CEF on a router, use the command **ip cef** [**distributed**]. The **distributed** keyword is used only on routers with a distributed architecture such as the 12000 and 7500 series routers.

Example 6-42 shows CEF being enabled on Chengdu_P.

Example 6-42 *Globally Enabling CEF*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#ip cef
Chengdu_P(config)#exit
Chengdu_P#
```

CEF is enabled in the highlighted line in Example 6-42.

In Example 6-43, the **show ip cef** command is again used to verify CEF.

Example 6-43 *CEF Is Enabled*

```
Chengdu_P#show ip cef
Prefix              Next Hop            Interface
0.0.0.0/0           drop                Null0 (default route handler entry)
0.0.0.0/32          receive
10.1.1.1/32         10.20.10.1          FastEthernet0/0
10.1.1.2/32         receive
10.1.1.3/32         10.20.20.2          Serial1/1
10.1.1.4/32         10.20.20.2          Serial1/1
10.20.10.0/24       attached            FastEthernet0/0
10.20.10.0/32       receive
10.20.10.1/32       10.20.10.1          FastEthernet0/0
10.20.10.2/32       receive
10.20.10.255/32     receive
10.20.20.0/24       attached            Serial1/1
10.20.20.0/32       receive
10.20.20.1/32       receive
10.20.20.2/32       attached            Serial1/1
10.20.20.255/32     receive
10.20.30.0/24       10.20.20.2          Serial1/1
224.0.0.0/4         0.0.0.0
224.0.0.0/24        receive
255.255.255.255/32  receive
Chengdu_P#
```

Example 6-43 shows a summary of the CEF forwarding information base (FIB).

Highlighted line 1 shows a default route to interface Null0 that reports a **drop** state. This indicates that packets for this FIB entry will be dropped.

In highlighted line 2, an entry for prefix 10.1.1.1/32 is shown. The entry includes the associated next-hop and (outgoing) interface.

Highlighted line 3 shows an entry for 10.1.1.2/32. This entry indicates a **receive** state. The **receive** state is used for host addresses configured on the local router. This entry corresponds to the IP address configured on Chengdu_P's interface loopback 0.

Finally, highlighted line 4 shows an entry for 10.20.10.0/24. This entry indicates an **attached** state. An **attached** state indicates that the prefix is directly reachable via the interface indicated (here, Fast Ethernet 0/0).

CEF Is Disabled on an Interface

After verifying that the CEF is globally enabled, also ensure that CEF is enabled on interfaces. CEF is responsible for label imposition and, therefore, must be enabled on the VRF interfaces on PE routers.

Use the **show cef interface** *interface_name* command to verify that CEF is enabled on an interface, as shown in Example 6-44.

Example 6-44 show cef interface *Command Output*

```
Chengdu_PE#show cef interface serial 4/1
Serial4/1 is up (if_number 6)
  Corresponding hwidb fast_if_number 6
  Corresponding hwidb firstsw->if_number 6
  Internet address is 172.16.4.1/24
  ICMP redirects are never sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  BGP based policy accounting is disabled
  Interface is marked as point to point interface
  Hardware idb is Serial4/1
  Fast switching type 7, interface type 67
  IP CEF switching disabled
  IP Feature Fast switching turbo vector
  IP Null turbo vector
  VPN Forwarding table "mjlnet_VPN"
  Input fast flags 0x1000, Output fast flags 0x0
  ifindex 5(5)
  Slot 4 Slot unit 1 Unit 1 VC -1
  Transmit limit accumulator 0x0 (0x0)
  IP MTU 1500
Chengdu_PE#
```

As you can see in the highlighted line, CEF is disabled on interface serial 4/1.

To enable CEF on the interface, use the **ip route-cache cef** command, as shown in Example 6-45.

Example 6-45 *Configuration of CEF on the Interface*

```
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#interface serial 4/1
Chengdu_PE(config-if)#ip route-cache cef
Chengdu_PE(config-if)#end
Chengdu_PE#
```

The highlighted line indicates that CEF is enabled on interface serial 4/1.

NOTE Note that some features are not supported by CEF switching. In this case, packets will be punted (sent to) to next fastest switching method.

The fastest switching path is dCEF, followed by CEF, fast switching, and process switching.

12000 series routers do not support fast switching or process switching. Instead of being punted, packets are simply dropped.

Verifying MPLS

If MPLS is disabled either globally or on an interface, label switching will not function.

This section discusses how to verify whether MPLS is either disabled globally or on an interface.

MPLS Is Globally Disabled

If MPLS has been globally enabled, label switching will not function on any interface.

The **show mpls interfaces** or **show mpls forwarding-table** commands can be used to verify that MPLS is enabled, as demonstrated in Example 6-46 and Example 6-47.

Example 6-46 *Verifying MPLS Using the* **show mpls interfaces** *Command*

```
Chengdu_PE#show mpls interfaces
IP MPLS forwarding is globally disabled on this router.
Individual interface configuration is as follows:
Interface              IP            Tunnel   Operational
Chengdu_PE#
```

The highlighted line clearly shows that MPLS is globally disabled.

Example 6-47 *Verifying MPLS Using the* **show mpls forwarding-table** *Command*

```
Chengdu_PE#show mpls forwarding-table
Tag switching is not operational.
CEF or tag switching has not been enabled.
No TFIB currently allocated.
Chengdu_PE#
```

Highlighted line 1 shows that either CEF or MPLS is disabled.

In highlighted line 2, you can see that no LFIB (shown as TFIB here) has been allocated.

MPLS can be enabled using the **mpls ip** command. In Example 6-48, MPLS is configured on Chengdu_PE.

Example 6-48 *Configuration of MPLS on Chengdu_PE*

```
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#mpls ip
Chengdu_PE(config)#exit
Chengdu_PE#
```

In Example 6-48, MPLS is globally enabled using the **mpls ip** command.

MPLS Is Disabled on an Interface

If MPLS is disabled on an interface, label switching will not function on that interface. Ensure that MPLS is enabled on all core interfaces of all PE and P routers. Note that MPLS should not be enabled on PE routers' VRF interfaces unless carrier's carrier MPLS is being used.

To verify that MPLS is enabled on core interfaces, use the **show mpls interfaces** command, as shown in Example 6-49.

Example 6-49 *Verifying MPLS on an Interface Using the* **show mpls interfaces** *Command (MPLS Is Disabled)*

```
Chengdu_PE#show mpls interfaces
Interface              IP            Tunnel   Operational
Chengdu_PE#
```

As you can see, no interfaces on Chengdu_PE are enabled for MPLS. In this case, MPLS should be enabled on core interface Fast Ethernet 1/0.

The **mpls ip** command is then used to enable MPLS on interface Fast Ethernet 1/0, as demonstrated in Example 6-50.

Example 6-50 *Enabling MPLS on Interface Fast Ethernet 1/0 Using the* **mpls ip** *Command*

```
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#interface fastethernet 1/0
Chengdu_PE(config-if)#mpls ip
Chengdu_PE(config-if)#end
Chengdu_PE#
```

In highlighted line 1, MPLS is enabled on interface fastethernet 1/0.

As shown in Example 6-51, the **show mpls interfaces** command is then used to confirm that MPLS is enabled on the interface.

Example 6-51 *Verifying MPLS on an Interface (MPLS Is Enabled)*

```
Chengdu_PE#show mpls interfaces
Interface              IP            Tunnel   Operational
FastEthernet1/0        Yes (ldp)     No       Yes
Chengdu_PE#
```

As you can see, MPLS is now enabled on interface FastEthernet1/0.

Verifying TDP/LDP

TDP and LDP are used to exchange label bindings, but if they are not functioning correctly, label bindings will not be exchanged, and MPLS will not function.

This section examines how to verify correct operation of TDP or LDP. Note that examples in this section focus on LDP.

LDP Neighbor Discovery and Session Establishment Fails

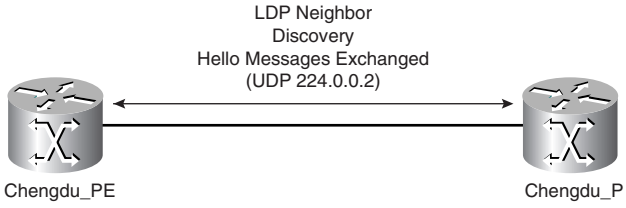
If LDP neighbor discovery fails, session establishment will fail. Similarly, if LDP session establishment fails, label bindings will not be distributed.

LDP Neighbor Discovery Fails

If LDP discovery fails, session establishment will fail between neighboring LSRs.

Figure 6-32 shows LDP neighbor discovery between Chengdu_PE and Chengdu_P.

Figure 6-32 *LDP Neighbor Discovery Between Chengdu_PE and Chengdu_P*



Note that Figure 6-32 shows LDP neighbor discovery between directly connected neighbors.

To verify that LDP neighbor discovery has been successful, the **show mpls ldp discovery** command can be used, as shown in Example 6-52.

Example 6-52 *Verifying LDP Neighbor Discovery Using the* **show mpls ldp discovery** *Command (Discovery Is Successful)*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit/recv
            LDP Id: 10.1.1.2:0
Chengdu_PE#
```

Highlighted line 1 shows the local LDP ID (10.1.1.1:0), which is comprised of a 32-bit router ID and a 16-bit label space identifier. In this case, the router ID is 10.1.1.1, and the label space identifier is 0 (which corresponds to a platform-wide label space).

Note that if an interface is using the platform-wide label space, it indicates that labels assigned on this interface are taken from a common pool. If an interface is using an interface label space, it indicates that labels assigned on the interfaces are taken from a pool of labels specific to this interface. Frame-mode interfaces use the platform-wide label space (unless a carrier's carrier architecture is deployed), and cell-mode interfaces use an interface label space.

Highlighted line 2 shows the interface on which LDP hello messages are being transmitted to (**xmit**) and received from (**recv**) the peer LSR. Note that the label protocol configured on the interface (in this case, LDP) is also indicated here. In highlighted line 3, the peer LSR's LDP ID is shown (10.1.1.2:0).

LDP neighbor discovery can fail for a number of reasons, including the following:

- A label protocol mismatch exists.
- An access list blocks neighbor discovery.
- A control-VC mismatch exists on LC-ATM interfaces.

These issues are detailed in the following sections.

Label Protocol Mismatch If there is a mismatch between the label protocol configured on neighboring LSRs, discovery will fail.

To verify neighbor discovery, use the **show mpls ldp discovery** command.

Example 6-53 shows the output of **show mpls ldp discovery** when there is a label protocol mismatch between LSRs.

Example 6-53 *Label Protocol Mismatch Between Peer LSRs*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit
Chengdu_PE#
```

As you can see, LDP hello messages are being transmitted (**xmit**) but not received (**recv**) on interface FastEthernet1/0. This might indicate that TDP is configured on the peer LSR.

To check the label protocol being used on the peer LSR, use the **show mpls interfaces** command, as shown in Example 6-54.

Example 6-54 *Verifying the Label Protocol on the Peer LSR Using the* **show mpls interfaces** *Command*

```
Chengdu_P#show mpls interfaces
Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (tdp)     No       Yes
Serial1/1              Yes (ldp)     No       Yes
Chengdu_P#
```

The highlighted line shows that TDP is indeed configured on the peer LSR's connected interface.

As shown in Example 6-55, the label protocol is changed to LDP on the interface using the **mpls label protocol** command.

Example 6-55 *Changing the Label Protocol Using the* **mpls label protocol** *Command*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#interface fastethernet0/0
Chengdu_P(config-if)#mpls label protocol ldp
Chengdu_P(config-if)#end
Chengdu_P#
```

The highlighted line shows that the label protocol is reconfigured to be LDP using the **mpls label protocol** command. Note that it is possible to configure both LDP and TDP on an interface using the **mpls label protocol both** command.

Once LDP has been configured on the peer LSR's interface, neighbor discovery is rechecked using the **show mpls ldp discovery** command, as demonstrated in Example 6-56.

Example 6-56 *Verifying Neighbor Discovery (Discovery Is Now Successful)*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit/recv
            LDP Id: 10.1.1.2:0
Chengdu_PE#
```

In Example 6-56, the highlighted line shows that LDP messages are now being both sent and received on interface FastEthernet1/0. LDP neighbor discovery has been successful.

Access List Blocks LDP Neighbor Discovery LDP neighbor discovery uses UDP port 646 and the all routers multicast address (224.0.0.2) for directly connected neighbors. If neighbors are not directly connected, then UDP port 646 is also used, but hello messages are unicast.

If an access list blocks UDP port 646 or the all routers multicast address, neighbor discovery will not function.

Note that TDP uses UDP 711 and the local broadcast address (255.255.255.255) for neighbor discovery. If neighbors are not directly connected, then unicast communication is again used.

LDP neighbor discovery can be verified using the **show mpls ldp discovery** command, as shown in Example 6-57.

Example 6-57 *Verifying LDP Neighbor Discovery Using the* **show mpls ldp discovery** *Command*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit
Chengdu_PE#
```

As highlighted line 1 shows, LDP hello messages are being transmitted (**xmit**), but not received (**recv**) on interface FastEthernet1/0. This may indicate the presence of an access list.

To check for the presence of an access list on an interface, use the **show ip interface** command, as demonstrated in Example 6-58.

Note that only the relevant portion of the output is shown.

As you can see, access list 101 is configured inbound on interface FastEthernet 1/0.

Example 6-58 *Verifying the Presence of an Access List Using the* **show ip interface** *Command*

```
Chengdu_PE#show ip interface fastethernet 1/0
FastEthernet1/0 is up, line protocol is up
  Internet address is 10.20.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.2
  Outgoing access list is not set
  Inbound  access list is 101
  Proxy ARP is disabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
```

To examine access list 101, use the **show ip access-lists** command, as demonstrated in Example 6-59.

Example 6-59 *Verifying the Contents of the Access List Using the* **show ip access-lists** *Command*

```
Chengdu_PE#show ip access-lists 101

Extended IP access list 101
    permit tcp any any eq bgp
    permit tcp any any eq ftp
    permit tcp any any eq ftp-data
    permit tcp any any eq nntp
    permit tcp any any eq pop3
    permit tcp any any eq smtp
    permit tcp any any eq www
    permit tcp any any eq telnet
    permit udp any any eq snmp
    permit udp any any eq snmptrap
    permit udp any any eq tacacs
    permit udp any any eq tftp
Chengdu_PE#
```

As you can see, UDP port 646 (LDP) is not permitted by access list 101, and it is, therefore, denied by the implicit **deny any** statement at the end of the access list.

There are two choices here:

- Modify the access list
- Remove the access list

In this case, it is decided that the access list is unnecessary, and so it is removed, as shown in Example 6-60.

Example 6-60 *Access List 101 Is Removed*

```
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#interface fastethernet 1/0
Chengdu_PE(config-if)#no ip access-group 101 in
Chengdu_PE(config-if)#end
Chengdu_PE#
```

The highlighted line shows the removal of access list 101 on interface fastethernet 1/0.

After access list 101 is removed, the **show mpls ldp discovery** command is used to verify that the LDP neighbor discovery is functioning, as demonstrated in Example 6-61.

Example 6-61 *LDP Discovery Is Now Successful*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit/recv
            LDP Id: 10.1.1.2:0
Chengdu_PE#
```

Highlighted line 1 shows that LDP hello messages are now being received (**recv**) on interface FastEthernet1/0.

Neighbor discovery is now successful.

Control VC Mismatch on LC-ATM Interfaces On LC-ATM interfaces, if there is a mismatch of the VPI/VCI for the control (plane) VC, LDP neighbor discovery will fail.

Use the **show mpls ldp discovery** command to view the neighbor discovery status on the LSR, as shown in Example 6-62.

Example 6-62 *Verifying LDP Discovery*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        ATM3/0.1 (ldp): xmit
Chengdu_PE#
```

Highlighted line 1 shows that LDP packets are being transmitted (**xmit**) but not received (**recv**) on interface ATM 3/0.1.

The next step is to check the control VC on the interface using the **show mpls interfaces detail** command, as shown in Example 6-63.

Example 6-63 *Checking the Control VC Using the* **show mpls interfaces detail** *Command on the Local LSR*

```
Chengdu_PE#show mpls interfaces atm 3/0.1 detail
Interface ATM3/0.1:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP labeling not enabled
        MPLS operational
        Optimum Switching Vectors:
          IP to MPLS Turbo Vector
          MPLS Turbo Vector
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 4470
        ATM labels: Label VPI = 1, Control VC = 0/32
Chengdu_PE#
```

Highlighted line 1 shows that the control VC used on this LC-ATM interface is 0/32 (VPI/VCI). This is the default.

The control VC is then verified on the peer LSR, as shown in Example 6-64.

Example 6-64 *Checking the Control VC on the Peer LSR*

```
HongKong_PE#show mpls interfaces atm 4/0.1 detail
Interface ATM4/0.1:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP labeling not enabled
        MPLS not operational
        Optimum Switching Vectors:
          IP to MPLS Turbo Vector
          MPLS Turbo Vector
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 4470
        ATM labels: Label VPI = 1, Control VC = 0/40
HongKong_PE#
```

As you can see, the control VC is 0/40 on HongKong_PE. There is a control VC mismatch between LDP peers.

To resolve this issue, the control VC is modified on HongKong_PE, as shown in Example 6-65.

Example 6-65 *Reconfiguration of the Control VC on HongKong_PE*

```
HongKong_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
HongKong_PE(config)#interface ATM4/0.1 mpls
HongKong_PE(config-subif)#mpls atm control-vc 0 32
HongKong_PE(config-subif)#end
HongKong_PE#
```

The control VC is reset to the default 0/32 values, as the highlighted line indicates.

Once the control VC VPI/VCI is modified, the **show mpls ldp discovery** command is again used to examine the LDP neighbor discovery state, as shown in Example 6-66.

Example 6-66 *LDP Discovery Is Now Successful*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        ATM3/0.1 (ldp): xmit/recv
            LDP Id: 10.1.1.4:1; IP addr: 10.20.60.2
Chengdu_PE#
```

In highlighted line 1, LDP hello packets are being both transmitted (xmit) and received (recv) on interface ATM 3/0.1. LDP discovery is now successful.

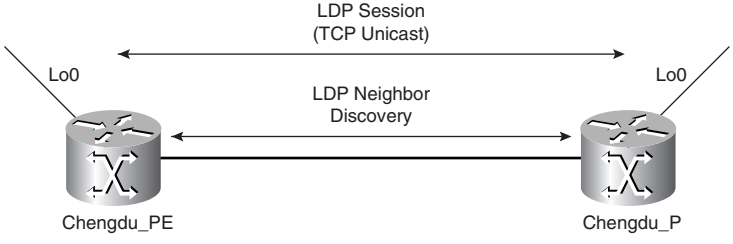
In highlighted line 2, the LDP ID (10.1.1.4:1) of HongKong_PE is shown, together with its IP address on the connected interface (10.20.60.2).

LDP Session Establishment Fails

If LDP session establishment fails, label bindings will not be advertised to neighboring LSRs.

Figure 6-33 illustrates an LDP session between Chengdu_PE and Chengdu_P.

Figure 6-33 *An LDP Session Between Chengdu_PE and Chengdu_P*



To verify LDP session establishment, use the **show mpls ldp neighbor** command.

Example 6-67 shows the output of the **show mpls ldp neighbor** command when session establishment is successful.

Example 6-67 *LDP Session Establishment Is Successful*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.1.1.2.11206 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 76/75; Downstream
        Up time: 00:56:44
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.20.10.2
        Addresses bound to peer LDP Ident:
          10.1.1.2        10.20.20.1      10.20.10.2
        Chengdu_PE#
```

Highlighted line 1 shows the peer LDP ID (10.1.1.2:0), as well as the local LDP ID (10.1.1.1:0).

In highlighted line 2, the TCP ports open on peer and local LSRs for the LDP session (11206 and 646, respectively) are shown.

In highlighted line 3, the session state is shown as operational (established). The number of messages sent and received (76 and 75), together with the method of label distribution (unsolicited downstream), are also shown.

The LDP session uptime is shown in highlighted line 4 (56 minutes and 44 seconds). In highlighted line 5, the discovery sources (local LSR interface and peer's connected IP address) are shown. Finally, the LDP peer's interface IP addresses are shown.

Numerous issues can prevent LDP session establishment, including the following:

- The neighbor's LDP ID is unreachable.
- An access list blocks LDP session establishment.
- An LDP authentication mismatch exists.
- VPI ranges do not overlap between LC-ATM interfaces.

The sections that follow discuss these issues in more detail.

Neighbor's LDP ID Is Unreachable An LDP session is established over a TCP connection between LSRs. On Cisco LSRs, the endpoint of the TCP connection corresponds to the LDP ID address by default, unless peer LSRs are connected via LC-ATM interfaces. If the LDP ID of the peer is unreachable, session establishment will fail.

Use the **show mpls ldp discovery** command to troubleshoot this issue, as shown in Example 6-68.

Example 6-68 *No Route to the LDP ID of the Neighboring LSR Exists*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit/recv
            LDP Id: 10.1.1.2:0; no route
Chengdu_PE#
```

The highlighted line shows that there is no route to the LDP ID of the neighboring LSR. As previously mentioned, LDP sessions are established between the LDP ID addresses of the neighboring LSRs. The absence of a route to the neighbor's LDP ID can be confirmed using the **show ip route** command, as demonstrated in Example 6-69.

Example 6-69 *No Route to the LDP ID of the Peer LSR Exists in the Routing Table*

```
Chengdu_PE#show ip route 10.1.1.2
% Subnet not in table
Chengdu_PE#
```

As you can see, there is no route to 10.1.1.2 (the peer's LDP ID).

When the configuration of the backbone IGP (in this case, IS-IS) is examined on the neighboring LSR, the problem is revealed.

Example 6-70 shows the output of the **show running-config** command. Note that only the relevant portions of the output are shown.

Example 6-70 *Interface Loopback0 Is Not Advertised by IS-IS*

```
Chengdu_P#show running-config
Building configuration...
!
tag-switching tdp router-id Loopback0 force
!
!
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
 no ip directed-broadcast
!
!
router isis
 net 49.0001.0000.0000.0002.00
 is-type level-2-only
 metric-style wide
!
```

In highlighted line 1, the MPLS LDP ID (shown as the TDP ID) is configured as the IP address on interface Loopback0.

The configuration of interface Loopback 0 begins in highlighted line 2. The IP address is 10.1.1.2/32. This is the LDP ID.

Notice that the command **ip router isis** is not configured on the interface. This command is one way to advertise the interface address in IS-IS.

The configuration of IS-IS begins in highlighted line 3. Notice the absence of the **passive-interface Loopback0** command. The **passive-interface** command alone can also be used to advertise the loopback interface although some versions of IOS may require you to configure the **ip router isis** on the loopback interface in addition to the **passive interface** command.

Interface Loopback0 is not being advertised in IS-IS. IS-IS must, therefore, be configured to advertise interface Loopback0, as shown in Example 6-71.

Example 6-71 *Configuring IS-IS to Advertise Interface Loopback0*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#router isis
Chengdu_P(config-router)#passive-interface Loopback0
Chengdu_P(config-router)#end
Chengdu_P#
```

The highlighted line shows where IS-IS is configured to advertise interface loopback 0.

The LDP discovery state is now rechecked using the **show mpls ldp discovery** command, as shown in Example 6-72.

Example 6-72 *LDP Discovery Is Now Successful*

```
Chengdu_PE#show mpls ldp discovery
 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet1/0 (ldp): xmit/recv
            LDP Id: 10.1.1.2:0
Chengdu_PE#
```

The highlighted line shows the peer LDP ID, and crucially, the absence of the "no route" message (as shown in Example 6-68) indicates that there is now a route to the neighbor's LDP ID.

The LDP session state can then be verified using the **show mpls ldp neighbor** command, as shown in Example 6-73.

Example 6-73 *LDP Session Is Established*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.1.1.2.11007 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 12/11; Downstream
        Up time: 00:00:43
        LDP discovery sources:
```

Example 6-73 *LDP Session Is Established (Continued)*

```
          FastEthernet1/0, Src IP addr: 10.20.10.2
        Addresses bound to peer LDP Ident:
          10.20.10.2      10.1.1.2        10.20.20.1
Chengdu_PE#
```

Highlighted line 1 shows the peer (10.1.1.2:0) and local LDP IDs (10.1.1.1:0).

In highlighted line 2, the session state is shown as operational (established). The number of messages sent and received (12 and 11), together with the label distribution method (unsolicited downstream) are also shown.

The LDP session uptime is shown in highlighted line 3 (43 seconds). The session has now been established.

It is worth noting that reachability issues between LDP ID addresses in a carrier's carrier configuration between the PE and CE routers can easily be resolved by using the **mpls ldp discovery transport-address interface** command. If this command is configured on connected PE and CE interfaces, the LDP session will be established between the connected interface addresses rather than LDP ID addresses.

Access List Blocks LDP Session Establishment LDP sessions are established between two peers over a unicast connection on TCP port 646. The unicast connection is between the LDP ID addresses of the adjacent LSRs. If an access list blocks TCP port 646 or the LDP ID addresses, then session establishment will fail.

When designing access lists, consider that the passive peer (the peer with the lower LDP ID) opens TCP port 646, and the active peer (the peer with the higher LDP ID) opens an ephemeral (short-lived) port for LDP session establishment.

Note that TDP uses TCP port 711 and a unicast connection for session establishment.

Use the **show ip interface** command to check for an access list on an interface, as demonstrated in Example 6-74.

Example 6-74 *Verifying the Presence of an Access List*

```
Chengdu_PE#show ip interface fastethernet 1/0
FastEthernet1/0 is up, line protocol is up
  Internet address is 10.20.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.2
  Outgoing access list is not set
  Inbound  access list is 101
  Proxy ARP is disabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
```

The highlighted line shows that access list 101 is configured inbound on interface FastEthernet 1/0.

Use the **show ip access-lists** command to examine access list 101, as shown in Example 6-75.

Example 6-75 *Verifying the Contents of the Access List*

```
Chengdu_PE#show ip access-lists
Extended IP access list 101
    permit icmp any any
    permit gre any any
    permit tcp any any eq bgp
    permit tcp any any eq domain
    permit tcp any any eq ftp
    permit tcp any any eq ftp-data
    permit tcp any any eq telnet
    permit tcp any any eq www
    permit udp any any eq 646
    permit udp any any eq ntp
    permit udp any any eq snmp
    permit udp any any eq snmptrap
    permit udp any any eq tacacs
    permit udp any any eq tftp

Chengdu_PE#
```

As you can see, access list 101 does not permit TCP port 646 and, therefore, it is blocked by the implicit **deny any** statement at the end of the access list.

The two choices here are:

- Modify the access list
- Remove the access list

In this case, it is decided that the access list is not needed, and it is removed, as shown in Example 6-76.

Example 6-76 *Removing the Access List*

```
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#interface fastethernet 1/0
Chengdu_PE(config-if)#no ip access-group 101 in
Chengdu_PE(config-if)#end
Chengdu_PE#
```

The highlighted line shows the removal of access list 101 on interface fastethernet 1/0.

Once the access list is removed, session establishment is verified using the **show mpls ldp neighbor** command, as demonstrated in Example 6-77.

Example 6-77 *LDP Session Establishment Succeeds*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.1.1.2.11075 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 15/14; Downstream
        Up time: 00:02:49
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.20.10.2
        Addresses bound to peer LDP Ident:
          10.1.1.2        10.20.20.1      10.20.10.2
Chengdu_PE#
```

Highlighted line 1 shows the peer (10.1.1.2:0) and local LDP IDs (10.1.1.1:0).

Highlighted line 2 shows that the session state is now operational (established). The number of messages sent and received (15 and 14), together with the label distribution method (unsolicited downstream), is also shown.

The LDP session uptime is shown in highlighted line 3 (2 minutes 49 seconds).

LDP Authentication Mismatch LDP can be configured to use the TCP MD5 authentication for session connections. If LDP authentication is configured on one peer, but not the other, or if passwords are mismatched, session establishment will fail.

LDP Authentication Is Configured on One Peer But Not the Other If LDP authentication is configured on one LDP peer, but not the other, session establishment will fail, and an error message will be logged.

Example 6-78 shows the error message logged if the LDP session messages do not contain an MD5 digest.

Example 6-78 *LDP Authentication Is Not Configured on the Peer LSR*

```
*Jan 20 08:34:16.775 UTC: %TCP-6-BADAUTH: No MD5 digest from 10.1.1.2(11023) to
  10.1.1.1(646)
```

In Example 6-78, an LDP session message has been received from LDP peer 10.1.1.2 without the expected MD5 digest.

To resolve this issue, either peer 10.1.1.2 can be configured for LDP authentication or LDP authentication can be removed on peer 10.1.1.1. In this case, LDP authentication is configured on peer 10.1.1.2, as shown in Example 6-79.

Example 6-79 *Configuration of LDP Authentication on Peer 10.1.1.2*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#mpls ldp neighbor 10.1.1.1 password cisco
Chengdu_P(config)#exit
Chengdu_P#
```

Once LDP authentication has been configured, the LDP session is established. This is verified using the **show mpls ldp neighbor** command, as shown in Example 6-80.

Example 6-80 *LDP Session Establishment Is Successful*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.1.1.2.11115 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 12/11; Downstream
        Up time: 00:00:21
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.20.10.2
        Addresses bound to peer LDP Ident:
          10.1.1.2        10.20.20.1      10.20.10.2
Chengdu_PE#
```

In highlighted line 1, the peer (10.1.1.2:0) and local LDP IDs (10.1.1.1:0) are shown.

Highlighted line 2 shows that the session state is operational (established). This line also shows the number of messages sent and received (12 and 11), together with the label distribution method (unsolicited downstream).

Finally, highlighted line 3 shows the LDP session uptime (21 seconds).

LDP Authentication Password Mismatch If there is a LDP authentication password mismatch between peers, session establishment will fail, and an error message will be logged.

Example 6-81 shows the error message logged if there is an LDP password mismatch.

Example 6-81 *LDP Passwords Are Mismatched*

```
*Jan 20 09:42:54.091 UTC: %TCP-6-BADAUTH: Invalid MD5 digest from 10.1.1.2
  (11034) to 10.1.1.1(646)
```

As the highlighted portion shows, an invalid MD5 digest is received from LDP peer 10.1.1.2.

To ensure that the LDP password is consistent, reconfigure the password on both peers (10.1.1.1 and 10.1.1.2) as shown in Example 6-82.

Example 6-82 *Reconfiguration of the LDP Password*

```
! On Chengdu_PE (10.1.1.1):
Chengdu_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_PE(config)#mpls ldp neighbor 10.1.1.2 password cisco
Chengdu_PE(config)#exit
Chengdu_PE#
```

```
! On Chengdu_P (10.1.1.2):
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#mpls ldp neighbor 10.1.1.1 password cisco
Chengdu_P(config)#exit
Chengdu_P#
```

Once the LDP password has been reconfigured, use the **show mpls neighbor** command to verify LDP session establishment as demonstrated in Example 6-83.

Example 6-83 *LDP Session Establishment Is Successful After Reconfiguration of the LDP Password*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.1.1.2.11118 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 12/11; Downstream
        Up time: 00:00:10
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.20.10.2
        Addresses bound to peer LDP Ident:
          10.1.1.2        10.20.20.1      10.20.10.2
Chengdu_PE#
```

The peer (10.1.1.2:0) and local LDP IDs (10.1.1.1:0) are shown in highlighted line 1.

In highlighted line 2, you can see that the session state is now operational (established). The number of messages sent and received (12 and 11), together with the label distribution method (unsolicited downstream), are also shown.

Highlighted line 3 shows the LDP session uptime (10 seconds).

VPI Ranges Do Not Overlap Between LC-ATM Interfaces During LDP session initialization, session parameters—such as LDP protocol version, label distribution method, and (on LC-ATM interfaces) VPI/VCI ranges used for label switching—are negotiated between peers.

If there is no overlap between VPI ranges configured on LDP peers, an error message is logged and session establishment fails, as shown in Example 6-84.

Example 6-84 *VPI Ranges Do Not Overlap Between LC-ATM Interfaces*

```
*Feb  8 14:09:06.038 UTC: %TDP-3-TAGATM_BAD_RANGE: Interface ATM3/0.1, Bad
  VPI/VCI range. Can't start a TDP session
```

In Example 6-84, the error message indicates that the VPI/VCI negotiation has failed on interface atm3/0.1, and the LSRs are unable to start a LDP (shown as TDP) session.

You can also use the **debug mpls atm-ldp api** command to troubleshoot this issue, as shown in Example 6-85.

Example 6-85 **debug atm-ldp api** *Command Output*

```
Chengdu_PE#debug mpls atm-ldp api
LC-ATM API debugging is on
Chengdu_PE#
*Feb  8 14:27:07.226 UTC: TAGATM_API: Disjoint VPI local[1-1], peer[2-3]
Chengdu_PE#
```

The highlighted portion reveals that VPI range 1–1 is configured locally, and VPI range 2–3 is configured on the peer LSR. Note that the default VPI range is 1–1.

To correct this problem, the VPI range is reconfigured on the peer LSR. This is shown in Example 6-86.

Example 6-86 *Reconfiguration of the VPI Range on the Peer LSR*

```
HongKong_PE#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
HongKong_PE(config)#interface atm4/0.1 mpls
HongKong_PE(config-subif)#no mpls atm vpi 2-3
HongKong_PE(config-subif)#end
HongKong_PE#
```

The highlighted line shows that the VPI range 2–3 is removed. This resets the VPI to the default range of 1–1.

After the VPI range on the peer LSR (Hongkong_PE) is reconfigured, LDP session establishment is successful.

To verify successful session establishment, the **show mpls ldp neighbor** command is used on HongKong_PE, as shown in Example 6-87.

Example 6-87 *LDP Session Establishment Succeeds After Reconfiguration of the VPI Range on HongKong_PE*

```
Chengdu_PE#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.4:1; Local LDP Ident 10.1.1.1:1
        TCP connection: 10.20.60.2.11036 - 10.20.60.1.646
        State: Oper; Msgs sent/rcvd: 14/14; Downstream on demand
        Up time: 00:06:03
        LDP discovery sources:
          ATM3/0.1, Src IP addr: 10.20.60.2
Chengdu_PE#
```

The peer (10.1.1.4:1) and local LDP IDs (10.1.1.1:1) are shown in highlighted line 1.

Note that the label space identifier used here is 1. Remember that LC-ATM interfaces do not use the platform-wide label space, which is indicated by the label space identifier 0.

Highlighted line 2 shows that the session state is now operational (established). The number of messages sent and received (14 and 14) and the label distribution method (downstream-on-demand) are also shown.

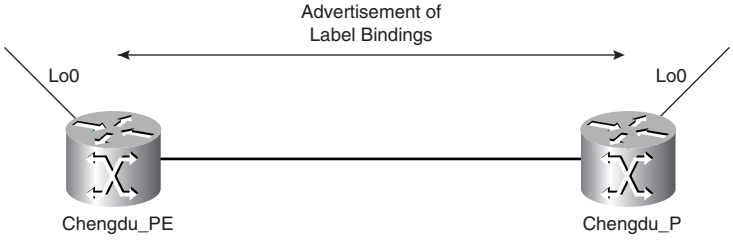
Highlighted line 3 shows the LDP session uptime (6 minutes 3 seconds).

Label Bindings Are Not Advertised Correctly

If LDP session establishment is successful, but label bindings are not advertised correctly, label switching will not function correctly.

Figure 6-34 shows the advertisement of label bindings between Chengdu_PE and Chengdu_P.

Figure 6-34 *Advertisement of Label Bindings Between Chengdu_PE and Chengdu_P*



To verify that labels are being advertised correctly, use the **show mpls ldp bindings** command, as shown in Example 6-88. The resulting output shows the contents of the Label Information Base (LIB).

Example 6-88 *Verifying the Contents of the LIB*

```
Chengdu_PE#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: 19
  tib entry: 10.1.1.2/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.1.1.3/32, rev 14
        local binding:  tag: 20
        remote binding: tsr: 10.1.1.2:0, tag: 18
  tib entry: 10.1.1.4/32, rev 18
        local binding:  tag: 22
        remote binding: tsr: 10.1.1.2:0, tag: 20
  tib entry: 10.20.10.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.0/24, rev 10
        local binding:  tag: 18
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.1/32, rev 16
        local binding:  tag: 21
  tib entry: 10.20.20.2/32, rev 6
        local binding:  tag: 16
        remote binding: tsr: 10.1.1.2:0, tag: 16
  tib entry: 10.20.30.0/24, rev 12
        local binding:  tag: 19
        remote binding: tsr: 10.1.1.2:0, tag: 17
Chengdu_PE#
```

Example 6-88 shows that label bindings are being received for all prefixes from the peer LSR.

For example, highlighted line 1 shows the LIB (shown here as TIB) entry for prefix 10.1.1.4/32. In highlighted line 2, the locally assigned label for this prefix is shown (22). In highlighted line 3, the label assigned by the peer LSR (10.1.1.2:0) for this prefix is shown (20).

The label bindings that correspond to the best routes are also contained within the LFIB. To examine the contents of the LFIB, use the **show mpls forwarding-table** command, as shown in Example 6-89.

Example 6-89 *Verifying the Contents of the LFIB*

```
Chengdu_PE#show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     16          10.20.20.2/32     0          Fa1/0      10.20.10.2
17     Pop tag     10.1.1.2/32       0          Fa1/0      10.20.10.2
18     Pop tag     10.20.20.0/24     0          Fa1/0      10.20.10.2
19     17          10.20.30.0/24     0          Fa1/0      10.20.10.2
20     18          10.1.1.3/32       0          Fa1/0      10.20.10.2
21     Untagged    10.20.20.1/32     0          Fa1/0      10.20.10.2
22     20          10.1.1.4/32       0          Fa1/0      10.20.10.2
23     Untagged    172.16.1.0/24[V]  0          Se4/1      point2point
24     Untagged    172.16.2.0/24[V]  0          Se4/1      point2point
25     Untagged    172.16.3.0/24[V]  0          Se4/1      point2point
26     Aggregate   172.16.4.0/24[V]  2080
27     Untagged    172.16.4.2/32[V]  0          Se4/1      point2point
Chengdu_PE#
```

The LFIB contains the locally assigned and outgoing (advertised by the peer LSR) labels for each prefix. Additionally, the number of bytes label switched, the outgoing interface, and the next-hop are shown.

As an example, the locally assigned and outgoing labels for prefix 10.1.1.4/32 are 22 and 20 respectively (see highlighted line 1). The number of bytes switched in 0, the outgoing interface is Fast Ethernet 1/0, and the next-hop is 10.20.10.2.

If label bindings are not advertised correctly, it may be because of a number of reasons, including:

- The **no mpls ldp advertise-labels** command is configured on the peer LSR.
- Conditional label advertisement blocks label bindings.
- CEF disables local label assignment.

The sections that follow discuss these issues.

no mpls ldp advertise-labels Command Is Configured on the Peer LSR If no label bindings are being received from a peer LSR, this may indicate that the peer LSR is configured not to advertise its locally assigned label bindings.

To verify that label bindings are being received from the peer LSR, use the **show mpls ldp bindings** command, as shown in Example 6-90.

Example 6-90 *No Label Bindings Are Received from the Peer LSR*

```
Chengdu_PE#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 2
        local binding:  tag: imp-null
  tib entry: 10.1.1.2/32, rev 8
        local binding:  tag: 17
  tib entry: 10.1.1.3/32, rev 14
        local binding:  tag: 20
  tib entry: 10.1.1.4/32, rev 18
        local binding:  tag: 22
  tib entry: 10.20.10.0/24, rev 4
        local binding:  tag: imp-null
  tib entry: 10.20.20.0/24, rev 10
        local binding:  tag: 18
  tib entry: 10.20.20.1/32, rev 16
        local binding:  tag: 21
  tib entry: 10.20.20.2/32, rev 6
        local binding:  tag: 16
  tib entry: 10.20.30.0/24, rev 12
        local binding:  tag: 19
Chengdu_PE#
```

In Example 6-90, no label bindings are being received from LSR 10.1.1.2:0.

The highlighted line shows the LIB entry for prefix 10.1.1.4/32. As you can see, there is no label binding from LSR 10.1.1.2:0; there is only a local binding.

The configuration of the peer LSR is checked using the **show running-config** command, as demonstrated in Example 6-91. Note that only the relevant portion of the output is shown.

Example 6-91 *Checking the Configuration of the Peer LSR Using the* **show running-config** *Command*

```
Chengdu_P#show running-config
Building configuration...
!
ip multicast-routing
mpls label protocol ldp
no tag-switching advertise-tags
!
```

As you can see, the **no mpls ldp advertise-labels** (shown as **no tag-switching advertise-tags**) command is configured on the peer LSR. This command disables advertisement of label bindings by the LSR.

To enable that the LSR advertises labels, use the **mpls ldp advertise-labels** command, as shown in Example 6-92.

Example 6-92 *Label Advertisement Is Enabled on Chengdu_P*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#mpls ldp advertise-labels
Chengdu_P(config)#exit
Chengdu_P#
```

Once label advertisement on the peer LSR is enabled, the **show mpls ldp bindings** command is used to verify that the bindings are being received, as shown in Example 6-93.

Example 6-93 *Label Bindings Are Now Received from the Peer LSR*

```
Chengdu_PE#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: 19
  tib entry: 10.1.1.2/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.1.1.3/32, rev 14
        local binding:  tag: 20
        remote binding: tsr: 10.1.1.2:0, tag: 18
  tib entry: 10.1.1.4/32, rev 18
        local binding:  tag: 22
        remote binding: tsr: 10.1.1.2:0, tag: 20
  tib entry: 10.20.10.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.0/24, rev 10
        local binding:  tag: 18
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.1/32, rev 16
        local binding:  tag: 21
  tib entry: 10.20.20.2/32, rev 6
        local binding:  tag: 16
        remote binding: tsr: 10.1.1.2:0, tag: 16
  tib entry: 10.20.30.0/24, rev 12
        local binding:  tag: 19
        remote binding: tsr: 10.1.1.2:0, tag: 17
Chengdu_PE#
```

As you can see, label bindings are now being received from peer LSR 10.1.1.2:0. In highlighted line 1, the LIB entry for prefix 10.1.1.4/32 is shown. Highlighted line 2 shows the label binding for this prefix advertised by LSR 10.1.1.2:0.

Conditional Label Advertisement Blocks Label Bindings If some, but not all, expected label bindings are being received from a peer LSR, this might indicate the presence of conditional label advertisement on the peer LSR.

You can use the **show mpls ldp bindings** command to examine label bindings advertised from the peer LSRs, as shown in Example 6-94.

Example 6-94 *Verifying Label Bindings Advertised by Peer LSRs*

```
Chengdu_PE#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: 19
  tib entry: 10.1.1.2/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.1.1.3/32, rev 14
        local binding:  tag: 20
        remote binding: tsr: 10.1.1.2:0, tag: 18
  tib entry: 10.1.1.4/32, rev 18
        local binding:  tag: 22
  tib entry: 10.20.10.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.0/24, rev 10
        local binding:  tag: 18
        remote binding: tsr: 10.1.1.2:0, tag: imp-null
  tib entry: 10.20.20.1/32, rev 16
        local binding:  tag: 21
  tib entry: 10.20.20.2/32, rev 6
        local binding:  tag: 16
        remote binding: tsr: 10.1.1.2:0, tag: 16
  tib entry: 10.20.30.0/24, rev 12
        local binding:  tag: 19
        remote binding: tsr: 10.1.1.2:0, tag: 17
Chengdu_PE#
```

If you look closely at the output in Example 6-94, you will notice that there are both local and remote bindings for all prefixes, with the exception of 10.1.1.4/32 (highlighted). There is no remote binding for this prefix, which indicates that the peer LSR is not advertising one.

To check for the presence of conditional label advertisement on the peer LSR, use the **show running-config** command, as demonstrated in Example 6-95. Note that only the relevant portion of the configuration is shown.

Example 6-95 *Checking for the Presence of Conditional Label Advertisement*

```
Chengdu_P#show running-config
Building configuration...
!
ip multicast-routing
mpls label protocol ldp
no tag-switching advertise-tags
tag-switching advertise-tags for 1
!
!
access-list 1 permit 10.1.1.2
access-list 1 permit 10.1.1.3
access-list 1 permit 10.1.1.1
access-list 1 permit 10.20.10.0 0.0.0.255
```

Example 6-95 *Checking for the Presence of Conditional Label Advertisement (Continued)*

```
access-list 1 permit 10.20.20.0 0.0.0.255
access-list 1 permit 10.20.30.0 0.0.0.255
!
```

In highlighted lines 1 and 2, the peer LSR (Chengdu_P) is configured to advertise only labels for those prefixes specified in access list 1.

Highlighted lines 3 to 8 show access list 1. As you can see, prefix 10.1.1.4/32 is not permitted, which prevents the advertisement of a binding for this prefix.

To allow the advertisement of a binding for prefix 10.1.1.4/32, you can either modify or remove access list 1. In this scenario, conditional label advertisement is unnecessary, so it is removed, as shown in Example 6-96.

Example 6-96 *Conditional Label Advertisement Is Removed on Chengdu_P*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#mpls ldp advertise-labels
Chengdu_P(config)#no mpls ldp advertise-labels for 1
Chengdu_P(config)#exit
Chengdu_P#
```

In highlighted lines 1 and 2, conditional label advertisement is removed on Chengdu_P.

Having removed conditional label advertisement on Chengdu_P, use the **show mpls ldp bindings** command to confirm proper label bindings advertisement, as demonstrated in Example 6-97.

Example 6-97 *Confirming Advertisement of a Label Binding for Prefix 10.1.1.4/32*

```
Chengdu_PE#show mpls ldp bindings 10.1.1.4 32
  tib entry: 10.1.1.4/32, rev 18
        local binding:  tag: 22
        remote binding: tsr: 10.1.1.2:0, tag: 20
Chengdu_PE#
```

As you can see, a label binding for prefix 10.1.1.4/32 has now been received from the Chengdu_P.

Label bindings can also be filtered as they are *received* on an LSR using the **mpls ldp neighbor** [**vrf** *vpn-name*] *neighbor-address* **labels accept** *acl* command. Labels corresponding to prefixes permitted in a standard access list are accepted from the specified neighbor. Verify the presence of this command using the **show mpls ldp neighbor** *neighbor-address* **detail** command.

CEF Disables Local Label Assignment If labels are not being bound to prefixes locally, this might indicate that CEF is disabled on the LSR.

You can use the **show mpls ldp bindings** command to verify local label bindings as shown in Example 6-98.

Example 6-98 *Local Label Assignment Is Disabled*

```
Chengdu_P#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 5
        remote binding: tsr: 10.1.1.1:0, tag: imp-null
        remote binding: tsr: 10.1.1.3:0, tag: 19
  tib entry: 10.1.1.2/32, rev 2
        remote binding: tsr: 10.1.1.1:0, tag: 17
        remote binding: tsr: 10.1.1.3:0, tag: 17
  tib entry: 10.1.1.3/32, rev 7
        remote binding: tsr: 10.1.1.1:0, tag: 20
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
  tib entry: 10.1.1.4/32, rev 8
        remote binding: tsr: 10.1.1.1:0, tag: 22
        remote binding: tsr: 10.1.1.3:0, tag: 20
  tib entry: 10.20.10.0/24, rev 1
        remote binding: tsr: 10.1.1.1:0, tag: imp-null
        remote binding: tsr: 10.1.1.3:0, tag: 18
  tib entry: 10.20.20.0/24, rev 4
        remote binding: tsr: 10.1.1.1:0, tag: 18
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
  tib entry: 10.20.20.1/32, rev 9
        remote binding: tsr: 10.1.1.1:0, tag: 21
        remote binding: tsr: 10.1.1.3:0, tag: 16
  tib entry: 10.20.20.2/32, rev 3
        remote binding: tsr: 10.1.1.1:0, tag: 16
  tib entry: 10.20.30.0/24, rev 6
        remote binding: tsr: 10.1.1.1:0, tag: 19
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
Chengdu_P#
```

As you can see, the LIB contains remote label bindings but no local label bindings.

As shown in Example 6-99, you can use the **show ip cef summary** command to check whether CEF is running.

Example 6-99 *Verifying CEF Operation*

```
Chengdu_P#show ip cef summary
IP CEF without switching (Table Version 1), flags=0x0
  4294967293 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
  0 leaves, 0 nodes, 0 bytes, 4 inserts, 4 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 88235174
  2(0) CEF resets, 0 revisions of existing leaves
  Resolution Timer: Exponential (currently 1s, peak 0s)
  0 in-place/0 aborted modifications
  refcounts:  0 leaf, 0 node
  Table epoch: 0
%CEF not running
Chengdu_P#
```

The highlighted line shows that CEF is disabled. To enable CEF, use the **ip cef** command, as shown in Example 6-100.

Example 6-100 *Enabling CEF on Chengdu_P*

```
Chengdu_P#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Chengdu_P(config)#ip cef
Chengdu_P(config)#exit
Chengdu_P#
```

Once CEF has been enabled, the LIB is again examined using the **show mpls ldp bindings** command, as shown in Example 6-101.

Example 6-101 *Local Label Assignment Is Now Enabled*

```
Chengdu_P#show mpls ldp bindings
  tib entry: 10.1.1.1/32, rev 15
        local binding:  tag: 19
        remote binding: tsr: 10.1.1.1:0, tag: imp-null
        remote binding: tsr: 10.1.1.3:0, tag: 19
  tib entry: 10.1.1.2/32, rev 12
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.1:0, tag: 17
        remote binding: tsr: 10.1.1.3:0, tag: 17
  tib entry: 10.1.1.3/32, rev 13
        local binding:  tag: 18
        remote binding: tsr: 10.1.1.1:0, tag: 20
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
  tib entry: 10.1.1.4/32, rev 16
        local binding:  tag: 20
        remote binding: tsr: 10.1.1.1:0, tag: 22
        remote binding: tsr: 10.1.1.3:0, tag: 20
  tib entry: 10.20.10.0/24, rev 17
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.1:0, tag: imp-null
        remote binding: tsr: 10.1.1.3:0, tag: 18
  tib entry: 10.20.20.0/24, rev 14
        local binding:  tag: imp-null
        remote binding: tsr: 10.1.1.1:0, tag: 18
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
  tib entry: 10.20.20.1/32, rev 9
        remote binding: tsr: 10.1.1.1:0, tag: 21
        remote binding: tsr: 10.1.1.3:0, tag: 16
  tib entry: 10.20.20.2/32, rev 11
        local binding:  tag: 17
        remote binding: tsr: 10.1.1.1:0, tag: 16
  tib entry: 10.20.30.0/24, rev 10
        local binding:  tag: 16
        remote binding: tsr: 10.1.1.1:0, tag: 19
        remote binding: tsr: 10.1.1.3:0, tag: imp-null
Chengdu_P#
```

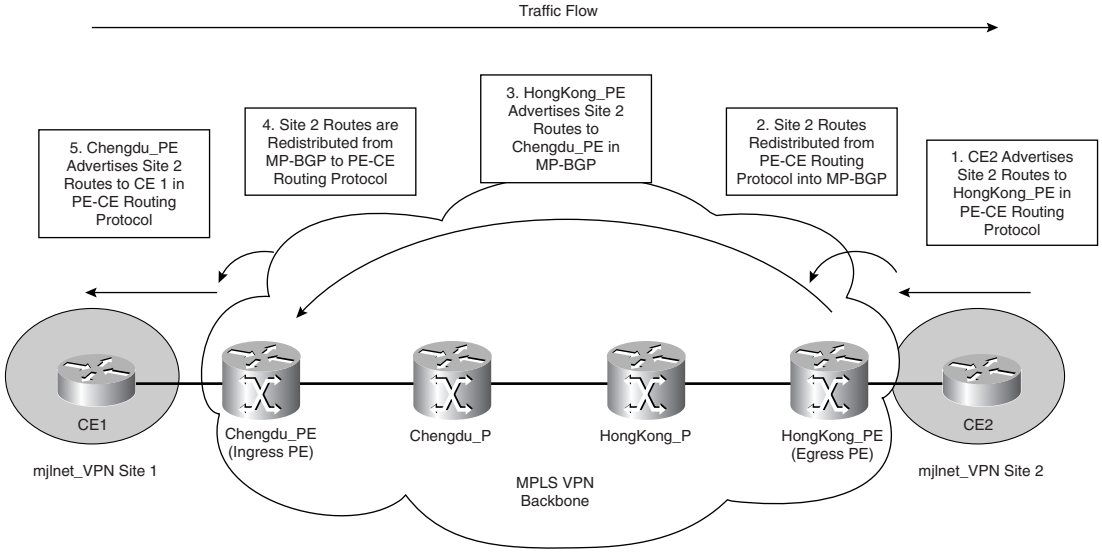
As you can see, the LIB now contains local label bindings.

Troubleshooting Route Advertisement Between VPN Sites

When troubleshooting route advertisement across the MPLS VPN backbone, you need to consider a number of issues. Before examining end-to-end troubleshooting of route advertisement, it is worthwhile to briefly review the issues involved.

Figure 6-35 illustrates route advertisement across the MPLS VPN backbone.

Figure 6-35 *Route Advertisement Across the MPLS VPN Backbone*



In Figure 6-35, route advertisement from CE2 to CE1 is as follows:

1. CE2 advertises customer site 2 routes to HongKong_PE using the PE-CE routing protocol (assuming that static routes are not being used).
2. HongKong_PE redistributes customer routes into MP-BGP.
3. HongKong_PE advertises the routes across the MPLS VPN backbone to Chengdu_PE, which imports the routes into its VRF.
4. Chengdu_PE redistributes the MP-BGP routes into the PE-CE routing protocol.
5. Chengdu_PE advertises the routes to CE1.