CHAPTER 19

Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) remains a vitally important part of many service provider (SP) networks. MPLS is still growing in popularity in enterprise networks as well, particularly in larger enterprise internetworks. This chapter introduces the core concepts with MPLS, particularly its use for unicast IP forwarding and for MPLS VPNs.

"Do I Know This Already?" Quiz

Table 19-1 outlines the major headings in this chapter and the corresponding "Do I Know This Already?" quiz questions.

Table 19-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundation Topics Section	Questions Covered in This Section	Score
MPLS Unicast IP Forwarding	1–4	
MPLS VPNs	5–8	
Other MPLS Applications	9	
Total Score		

In order to best use this pre-chapter assessment, remember to score yourself strictly. You can find the answers in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes."

1. Imagine a frame-based MPLS network configured for simple unicast IP forwarding, with four routers, R1, R2, R3, and R4. The routers connect in a mesh of links so that they are all directly connected to the other routers. R1 uses LDP to advertise prefix 1.1.1.0/24, label 30, to the other three routers. What must be true in order for R2 to advertise a label for 1.1.1.0/24 to R1 using LDP?

a. R2 must learn an IGP route to 1.1.1.0/24.

b. R2 will not advertise a label to R1 due to split horizon rules.

c. R2 can advertise a label back to R1 before learning an IGP route to 1.1.1.0/24.

d. R2 must learn a route to 1.1.1.0/24 using MP-BGP before advertising a label.

2. In a frame-based MPLS network configured for unicast IP forwarding, LSR R1 receives a labeled packet, with a label value of 55. Which of the following could be true?

   a. R1 makes its forwarding decision by comparing the packet to the IPv4 prefixes found in the FIB.

   b. R1 makes its forwarding decision by comparing the packet to the IPv4 prefixes found in the LFIB.

   c. R1 makes its forwarding decision by comparing the packet to the MPLS labels found in the FIB.

   d. R1 makes its forwarding decision by comparing the packet to the MPLS labels found in the LFIB.

3. R1, R2, and R3 are all MPLS LSRs that use LDP and connect to the same LAN. None of the three LSRs advertise a transport IP address. Which of the following could be true regarding LDP operation?

   a. The LSRs discover the other two routers using LDP Hellos sent to IP address 224.0.0.20.

   b. Each pair of LSRs forms a TCP connection before advertising MPLS labels.

   c. The three LSRs must use their LAN interface IP addresses for any LDP TCP connections.

   d. The LDP Hellos use port 646, with the TCP connections using port 711.

4. In a frame-based MPLS network configured for simple unicast IP forwarding, MPLS TTL propagation has been enabled for all traffic. Which of the following could be true?

   a. A **traceroute** command issued from outside the MPLS network will list IP addresses of the LSRs inside the MPLS network.

   b. A **traceroute** command issued from outside the MPLS network will not list IP addresses of the LSRs inside the MPLS network.

   c. Any IP packet with a TCP header, entering the MPLS network from outside the MPLS network, would not have its IP TTL field copied into the MPLS TTL field.

   d. An ICMP echo sent into the MPLS network from outside the MPLS network would have its IP TTL field copied into the MPLS TTL field.

5. Which of the following is an extension to the BGP NLRI field?

   a. VRF

   b. Route Distinguisher

   c. Route Target

   d. BGP Extended Community

6. Which of the following controls into which VRFs a PE adds routes when receiving an IBGP update from another PE?

   a. Route Distinguisher

   b. Route Target

   c. IGP metric

   d. AS Path length

7. An ingress PE router in an internetwork configured for MPLS VPN receives an unlabeled packet. Which of the following is true?

   a. It injects a single MPLS header.

   b. It injects at least two MPLS headers.

   c. It injects (at least) a VPN label, which is used by any intermediate P routers.

   d. It uses both the FIB and LFIB to find all the required labels to inject before the IP header.

8. An internetwork configured to support MPLS VPNs uses PHP. An ingress PE receives an unlabeled packet and then injects the appropriate label(s) to the packet before sending the packet into the MPLS network. Which of the following is/are true about this packet?

   a. The number of MPLS labels in the packet will only change when the packet reaches the egress PE router, which extracts the entire MPLS header.

   b. The number of MPLS labels in the packet will change before the packet reaches the egress PE.

   c. The PHP feature will cause the egress PE to act differently than it would without PHP enabled.

   d. None of the other answers is correct.

9. Which of the following answers help define which packets are in the same MPLS FEC when using MPLS VPNs?

   a. IPv4 prefix

   b. ToS byte

   c. The MPLS VRF

   d. The TE tunnel

Foundation Topics

MPLS defines protocols that create a different paradigm for how routers forward packets. Instead of forwarding packets based on the packets' destination IP address, MPLS defines how routers can forward packets based on an MPLS label. By disassociating the forwarding decision from the destination IP address, MPLS allows forwarding decisions based on other factors, such as traffic engineering, QoS requirements, and the privacy requirements for multiple customers connected to the same MPLS network, while still considering the traditional information learned using routing protocols.

MPLS includes a wide variety of applications, with each application considering one or more of the possible factors that influence the MPLS forwarding decisions. For the purposes of the CCIE Routing and Switching written exam, this book covers two such applications in the first two major sections of this chapter:

- MPLS unicast IP
- MPLS VPNs

This chapter ends with a brief introduction to many of the other MPLS applications. Also, as usual, please take the time to check http://www.ciscopress.com/title/9781587201967 for the latest version of Appendix C, "CCIE Routing and Switching Exam Updates," to find out if you should read further about any of the MPLS topics.

NOTE MPLS includes frame-mode MPLS and cell-mode MPLS, while this chapter only covers frame-mode MPLS. The generalized comments in this chapter may not apply to cell-mode MPLS.

MPLS Unicast IP Forwarding

MPLS can be used for simple unicast IP forwarding. With MPLS unicast IP forwarding, the MPLS forwarding logic forwards packets based on labels. However, when choosing the interfaces out which to forward the packets, MPLS considers only the routes in the unicast IP routing table, so the end result of using MPLS is that the packet flows over the same path as it would have if MPLS were not used, but all other factors were unchanged.

MPLS unicast IP forwarding does not provide any significant advantages by itself; however, many of the more helpful MPLS applications, such as MPLS VPNs and MPLS traffic engineering (TE), use MPLS unicast IP forwarding as one part of the MPLS network. So to understand MPLS as you would typically implement it, you need a solid understanding of MPLS in its most basic form: MPLS unicast IP forwarding.

MPLS requires the use of control plane protocols (for example, OSPF and LDP) to learn labels, correlate those labels to particular destination prefixes, and build the correct forwarding tables. MPLS also requires a fundamental change to the data plane's core forwarding logic. This section begins by examining the data plane, which defines the packet-forwarding logic. Following that, this section examines the control plane protocols, particularly the Label Distribution Protocol (LDP), which MPLS routers use to exchange labels for unicast IP prefixes.

MPLS IP Forwarding: Data Plane

MPLS defines a completely different packet-forwarding paradigm. However, hosts do not and should not send and receive labeled packets, so at some point, some router will need to add a label to the packet and, later, another router will remove the label. The MPLS routers—the routers that inject (push), remove (pop), or forward packets based on their labels—use MPLS forwarding logic.

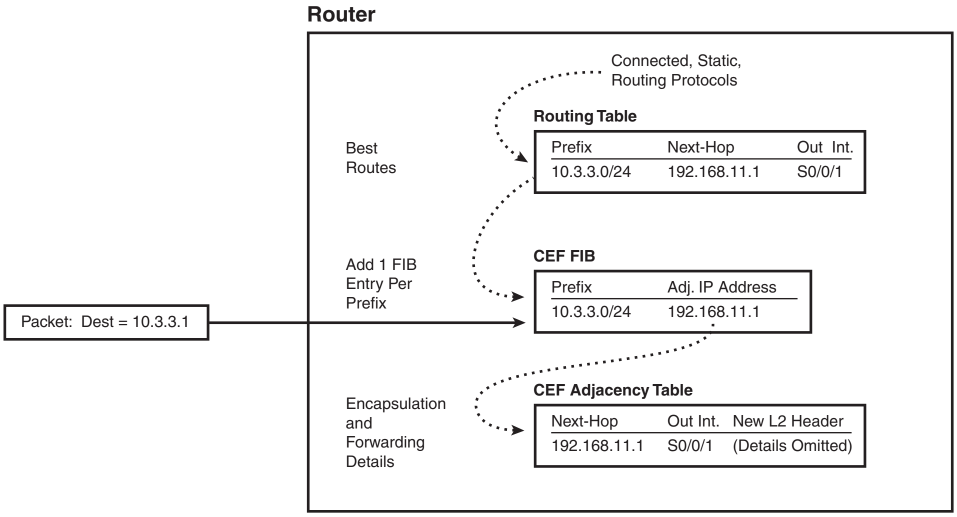
MPLS relies on the underlying structure and logic of Cisco Express Forwarding (CEF) while expanding the logic and data structures as well. First, a review of CEF is in order, followed by details about a new data structure called the MPLS *Label Forwarding Information Base (LFIB)*.

CEF Review

A router's unicast IP forwarding control plane uses routing protocols, static routes, and connected routes to create a *Routing Information Base (RIB)*. With CEF enabled, a router's control plane processing goes a step further, creating the CEF *Forwarding Information Base (FIB)*, adding a FIB entry for each destination IP prefix in the routing table. The FIB entry details the information needed for forwarding: the next-hop router and the outgoing interface. Additionally, the CEF *adjacency table* lists the new data-link header that the router will then copy in front of the packet before forwarding.

For the data plane, a CEF router compares the packet's destination IP address to the CEF FIB, ignoring the IP routing table. CEF optimizes the organization of the FIB so that the router spends very little time to find the correct FIB entry, resulting in a smaller forwarding delay and a higher volume of packets per second through a router. For each packet, the router finds the matching FIB entry, then finds the adjacency table entry referenced by the matching FIB entry, and forwards the packet. Figure 19-1 shows the overall process.

Figure 19-1 *IP Routing Table and CEF FIB—No MPLS*

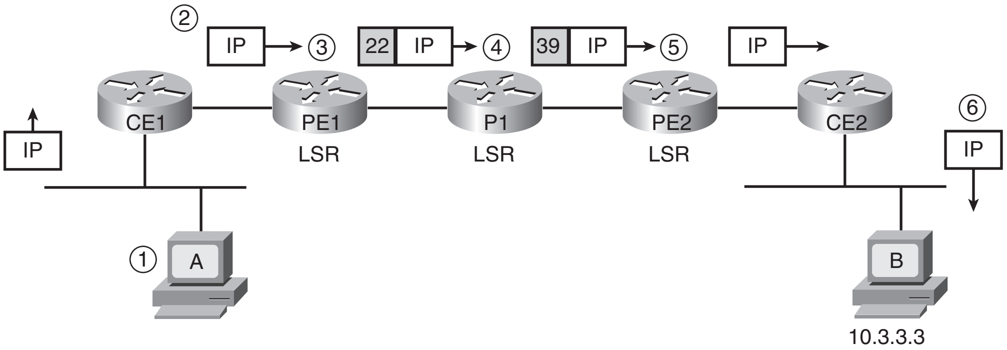


With this backdrop in mind, the text next looks at how MPLS changes the forwarding process using labels.

Overview of MPLS Unicast IP Forwarding

The MPLS forwarding paradigm assumes that hosts generate packets without an MPLS label; then, some router imposes an MPLS label, other routers forward the packet based on that label, and then other routers remove the label. The end result is that the host computers have no awareness of the existence of MPLS. To appreciate this overall forwarding process, Figure 19-2 shows an example, with steps showing how a packet is forwarded using MPLS.

Figure 19-2 *MPLS Packet Forwarding—End to End*



The steps from the figure are explained as follows:

1. Host A generates and sends an unlabeled packet destined to host 10.3.3.3.
2. Router CE1, with no MPLS features configured, forwards the unlabeled packet based on the destination IP address, as normal, without any labels. (Router CE1 may or may not use CEF.)
3. MPLS router PE1 receives the unlabeled packet and decides, as part of the MPLS forwarding process, to impose (push) a new label (value 22) into the packet and forwards the packet.
4. MPLS router P1 receives the labeled packet. P1 swaps the label for a new label value (39) and then forwards the packet.
5. MPLS router PE2 receives the labeled packet, removes (pops) the label, and forwards the packet toward CE2.
6. Non-MPLS router CE2 forwards the unlabeled packet based on the destination IP address, as normal. (CE2 may or may not use CEF.)

The steps in Figure 19-2 show a relatively simple process and provide a great backdrop from which to introduce a few terms. The term *Label Switch Router (LSR)* refers to any router that has awareness of MPLS labels, for example, routers PE1, P1, and PE2 in Figure 19-2. Table 19-2 lists the variations of the term LSR, and a few comments about the meaning of each term.

Table 19-2 *MPLS LSR Terminology Reference*

Key Topic

LSR Type	Actions Performed by This LSR Type
Label Switch Router (LSR)	Any router that pushes labels onto packets, pops labels from packets, or simply forwards labeled packets.
Edge LSR (E-LSR)	An LSR at the edge of the MPLS network, meaning that this router processes both labeled and unlabeled packets.
Ingress E-LSR	For a particular packet, the router that receives an unlabeled packet and then inserts a label stack in front of the IP header.
Egress E-LSR	For a particular packet, the router that receives a labeled packet and then removes all MPLS labels, forwarding an unlabeled packet.
ATM-LSR	An LSR that runs MPLS protocols in the control plane to set up ATM virtual circuits. Forwards labeled packets as ATM cells.
ATM E-LSR	An E-edge LSR that also performs the ATM Segmentation and Reassembly (SAR) function.

MPLS Forwarding Using the FIB and LFIB

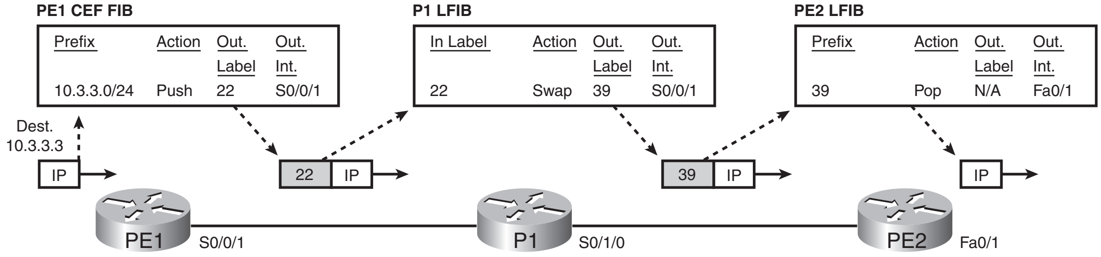
To forward packets as shown in Figure 19-2, LSRs use both the CEF FIB and the MPLS LFIB when forwarding packets. Both the FIB and LFIB hold any necessary label information, as well as the outgoing interface and next-hop information.

The FIB and LFIB differ in that routers use one table to forward incoming unlabeled packets, and the other to forward incoming labeled packets, as follows:

- **FIB**—Used for incoming unlabeled packets. Cisco IOS matches the packet's destination IP address to the best prefix in the FIB and forwards the packet based on that entry.
- **LFIB**—Used for incoming labeled packets. Cisco IOS compares the label in the incoming packet to the LFIB's list of labels and forwards the packet based on that LFIB entry.

Figure 19-3 shows how the three LSRs in Figure 19-2 use their respective FIBs and LFIB. Note that Figure 19-3 just shows the FIB on the LSR that forwards the packet using the FIB and the LFIB on the two LSRs that use the LFIB, although all LSRs have both a FIB and an LFIB.

Figure 19-3 *Usage of the CEF FIB and MPLS LFIB for Forwarding Packets*



PE1 CEF FIB

Prefix	Action	Out. Label	Out. Int.
10.3.3.0/24	Push	22	S0/0/1

P1 LFIB

In Label	Action	Out. Label	Out. Int.
22	Swap	39	S0/0/1

PE2 LFIB

Prefix	Action	Out. Label	Out. Int.
39	Pop	N/A	Fa0/1

The figure shows the use of the FIB and LFIB, as follows:

- **PE1**—When the unlabeled packet arrives at PE1, PE1 uses the FIB. PE1 finds the FIB entry that matches the packet's destination address of 10.3.3.1—namely, the entry for 10.3.3.0/24 in this case. Among other things, the FIB entry includes the instructions to push the correct MPLS label in front of the packet.
- **P1**—Because P1 receives a labeled packet, P1 uses its LFIB, finding the label value of 22 in the LFIB, with that entry stating that P1 should swap the label value to 39.
- **PE2**—PE2 uses the LFIB as well, because PE2 receives a labeled packet; the matching LFIB entry lists a pop action, so PE2 removes the label, forwarding an unlabeled packet to CE2.

Note that P1 and PE2 in this example never examined the packet's destination IP address as part of the forwarding process. Because the forwarding process does not rely on the destination IP address, MPLS can then enable forwarding processes based on something other than the destination IP address, such as forwarding based on the VPN from which the packet originated, forwarding to balance traffic with traffic engineering, and forwarding over different links based on QoS goals.

The MPLS Header and Label

The MPLS header is a 4-byte header, located immediately before the IP header. Many people simply refer to the MPLS header as the MPLS label, but the label is actually a 20-bit field in the MPLS header. You may also see this header referenced as an MPLS *shim header*. Figure 19-4 shows the entire label, and Table 19-3 defines the fields.

Figure 19-4 *The MPLS Header*

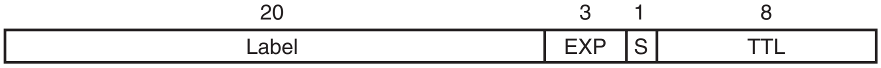


Table 19-3 *MPLS Header Fields*

Key Topic

Field	Length (Bits)	Purpose
Label	20	Identifies the portion of a label switched path (LSP).
Experimental (EXP)	3	Used for QoS marking; the field is no longer used for truly experimental purposes.
Bottom-of-Stack (S)	1	Flag, which when set to 1, means that this is the label immediately preceding the IP header.
Time-to-Live (TTL)	8	Used for the same purposes as the IP header's TTL field.

Of the four fields in the MPLS header, the first two, Label and EXP, should already be familiar. The 20-bit Label is usually listed as a decimal value in **show** commands. The MPLS EXP bits allow for QoS marking, which can be done using CB Marking, as covered in Chapter 12, "Classification and Marking." The S bit will make more sense once you examine how MPLS VPNs work, but in short, when packets hold multiple MPLS headers, this bit allows an LSR to recognize the last MPLS header before the IP header. Finally, the TTL field requires a little more examination, as covered in the next section.

The MPLS TTL Field and MPLS TTL Propagation

The IP header's TTL field supports two important features: a mechanism to identify looping packets, and a method for the **traceroute** command to find the IP address of each router in a particular end-to-end route. The MPLS header's TTL field supplies the same features—in fact, using all defaults, the presence or absence of MPLS LSRs in a network has no impact on the end results of either of the TTL-related processes.

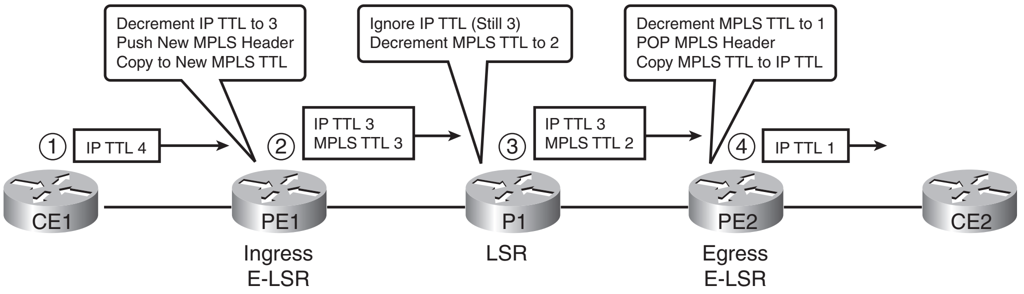
MPLS needs a TTL field so that LSRs can completely ignore the encapsulated IP header when forwarding IP packets. Essentially, the LSRs will decrement the MPLS TTL field, and not the IP TTL field, as the packet passes through the MPLS network. To make the whole process work, using all default settings, ingress E-LSRs, LSRs, and egress E-LSRs work as follows:

Key Topic

- **Ingress E-LSRs**— After an ingress E-LSR decrements the IP TTL field, it pushes a label into an unlabeled packet and then copies the packet's IP TTL field into the new MPLS header's TTL field.
- **LSRs**—When an LSR swaps a label, the router decrements the MPLS header's TTL field, and always ignores the IP header's TTL field.
- **Egress E-LSRs**—After an egress E-LSR decrements the MPLS TTL field, it pops the final MPLS header and then copies the MPLS TTL field into the IP header TTL field.

Figure 19-5 shows an example in which a packet arrives at PE1, unlabeled, with IP TTL 4. The callouts in the figure list the main actions for the three roles of the LSRs as described in the previous list.

Figure 19-5 *Example of MPLS TTL Propagation*



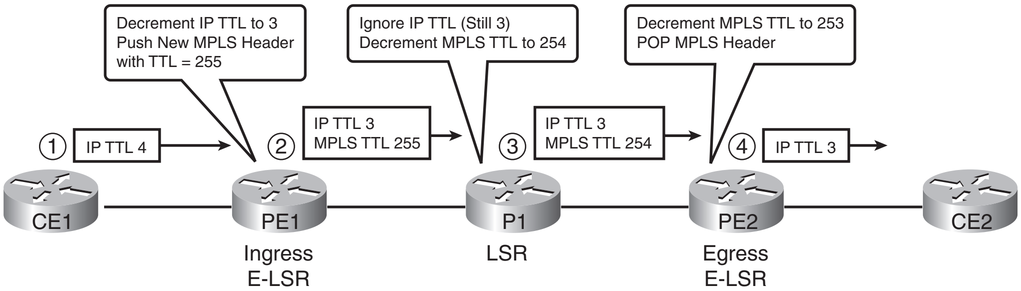
The term *MPLS TTL propagation* refers to the combined logic as shown in the figure. In effect, the MPLS routers propagate the same TTL value across the MPLS network—the same TTL values that would have occurred if MPLS was not used at all. As you might expect, a truly looping packet would eventually decrement to TTL 0 and be discarded. Additionally, a **traceroute** command would receive ICMP Time Exceeded messages from each of the routers in the figure, including the LSRs.

However, many engineers do not want hosts outside the MPLS network to have visibility into the MPLS network with the **traceroute** command. SPs typically implement MPLS networks to create Layer 3 WAN services, and the SP's customers sit outside the MPLS network. If the SP's customers can find the IP addresses of the MPLS LSRs, it may annoy the customer who wants to see only customer routers, and it may create a security exposure for the SP.

Key Topic

Cisco routers can be configured to disable MPLS TTL propagation. When disabled, the ingress E-LSR sets the MPLS header's TTL field to 255, and the egress E-LSR leaves the original IP header's TTL field unchanged. As a result, the entire MPLS network appears to be a single router hop from a TTL perspective, and the routers inside the MPLS network are not seen from the customer's **traceroute** command. Figure 19-6 shows the same example as in Figure 19-5 but now with MPLS TTL propagation disabled.

Figure 19-6 *Example with MPLS TTL Propagation Disabled*



Cisco supports the ability to disable MPLS TTL propagation for two classes of packets. Most MPLS SPs may want to disable TTL propagation for packets forwarded by customers, but allow TTL propagation for packets created by the SP's routers. Using Figure 19-5 again for an example, an SP engineer may be logged in to router PE1 in order to issue a **traceroute** command. PE1 can be configured to use TTL propagation for locally created packets, which allows the **traceroute** command issued from PE1 to list all the routers in the MPLS cloud. At the same time, PE1 can be configured to disable TTL propagation for "forwarded" packets (packets received from customers), preventing the customer from learning router IP addresses inside the MPLS network. (The command is **no mpls ttl-propagation** [**local** | **forwarded**].)

NOTE Although the PE1 router has TTL-Propagation disabled, *all* routers in the MPLS domain should also have TTL disabled for consistent output of the TTL propagation.

MPLS IP Forwarding: Control Plane

For pure IP routing to work using the FIB, routers must use control plane protocols, like routing protocols, to first populate the IP routing table and then populate the CEF FIB. Similarly, for MPLS forwarding to work, MPLS relies on control plane protocols to learn which MPLS labels to use to reach each IP prefix, and then populate both the FIB and the LFIB with the correct labels.

MPLS supports many different control plane protocols. However, an engineer's choice of which control plane protocol to use is mainly related to the MPLS application used, rather than any detailed comparison of the features of each control plane protocol. For example, MPLS VPNs use two control plane protocols: LDP and multiprotocol BGP (MP-BGP).

While multiple control plane protocols may be used for some MPLS applications, MPLS unicast IP forwarding uses an IGP and one MPLS-specific control plane protocol: LDP. This section, still focused on unicast IP forwarding, examines the details of label distribution using LDP.

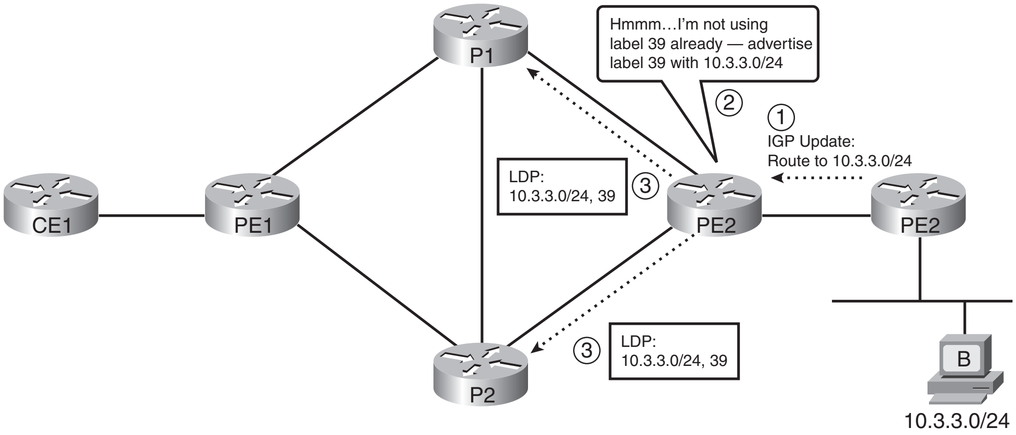
NOTE The earliest pre-standard version of LDP was called *Tag Distribution Protocol (TDP)*. The term *tag switching* was also often used instead of label switching.

MPLS LDP Basics

For unicast IP routing, LDP simply advertises labels for each prefix listed in the IP routing table. To do so, LSRs use LDP to send messages to their neighbors, with the messages listing an IP prefix and corresponding label. By advertising an IP prefix and label, the LSR is essentially saying, "If you want to send packets to this IP prefix, send them to me with the MPLS label listed in the LDP update."

The LDP advertisement is triggered by a new IP route appearing in the unicast IP routing table. Upon learning a new route, the LSR allocates a label called a local label. The local label is the label that, on this one LSR, is used to represent the IP prefix just added to the routing table. An example makes the concept much clearer. Figure 19-7 shows a slightly expanded version of the MPLS network shown earlier in this chapter. The figure shows the basic process of what occurs when an LSR (PE2) learns about a new route (10.3.3.0/24), triggering the process of advertising a new local label (39) using LDP.

Figure 19-7 *LDP Process Triggered by New Unicast IP Route*


P1
Hmmm…I'm not using label 39 already — advertise label 39 with 10.3.3.0/24
2
1
IGP Update: Route to 10.3.3.0/24
LDP: 10.3.3.0/24, 39
3
CE1
PE1
PE2
PE2
3
LDP: 10.3.3.0/24, 39
P2
B
10.3.3.0/24

The figure shows the following simple three-step process on PE2:

1. PE2 learns a new unicast IP route, which appears in the IP routing table.
2. PE2 allocates a new *local label*, which is a label not currently advertised by that LSR.
3. PE2 uses LDP to advertise to neighbors the mapping between the IP prefix and label to all LDP neighbors.

Although the process itself is simple, it is important to note that PE2 must now be ready to process labeled packets that arrive with the new local label value in it. For example, in Figure 19-7, PE2 needs to be ready to forward packets received with label 39; PE2 will forward the packets with the same next-hop and outgoing interface information learned in the IGP Update at step 1 in the figure.

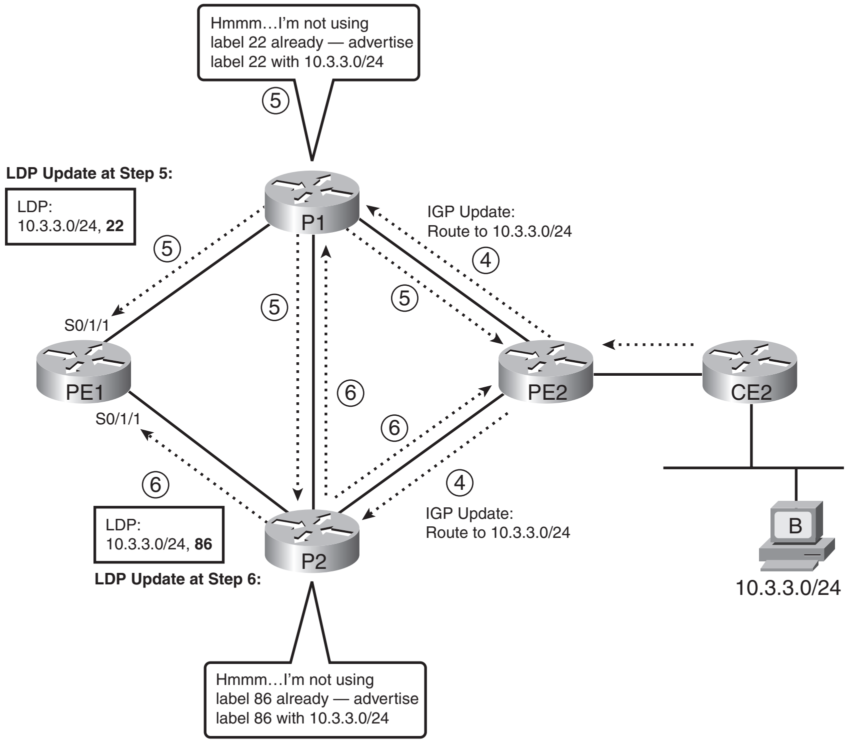
Key Topic

Although interesting, the process shown in Figure 19-7 shows only the advertisement of one segment of the full label switched path (LSP). An MPLS LSP is the combined set of labels that can be used to forward the packets correctly to the destination. For example, Figures 19-2 and 19-3 show a short LSP with label values 22 and 39, over which packets to subnet 10.3.3.0/24 were sent. Figure 19-7 shows the advertisement of one part, or segment, of the LSP.

NOTE LSPs are unidirectional.

The routers in the MPLS cloud must use some IP routing protocol to learn IP routes in order to trigger the LDP process of advertising labels. Typically, for MPLS unicast IP routing, you would use an IGP to learn all the IP routes, triggering the process of advertising the corresponding labels. For example, Figure 19-8 picks up the process where Figure 19-7 ended, with PE2 advertising a route for 10.3.3.0/24 using EIGRP, causing other routers to then use LDP to advertise labels.

Figure 19-8 *Completed Process of Advertising an Entire LSP*


Hmmm…I'm not using label 22 already — advertise label 22 with 10.3.3.0/24
5
LDP Update at Step 5:
LDP: 10.3.3.0/24, **22**
P1
IGP Update: Route to 10.3.3.0/24
4
5
5
5
S0/1/1
PE1
S0/1/1
6
6
PE2
CE2
6
6
4
LDP: 10.3.3.0/24, **86**
P2
IGP Update: Route to 10.3.3.0/24
B
10.3.3.0/24
LDP Update at Step 6:
Hmmm…I'm not using label 86 already — advertise label 86 with 10.3.3.0/24

The steps in the figure are as follows, using numbering that continues the numbering from Figure 19-7:

4. PE2 uses EIGRP to advertise the route for 10.3.3.0/24 to both P1 and P2.
5. P1 reacts to the newly learned route by allocating a new local label (22) and using LDP to advertise the new prefix (10.3.3.0/24) to label (20) mapping. Note that P1 advertises this label to all its neighbors.
6. P2 also reacts to the newly learned route by allocating a new local label (86) and using LDP to advertise the new prefix (10.3.3.0/24) to label (86) mapping. P2 advertises this label to all its neighbors.

This same process occurs on each LSR, for each route in the LSR's routing table: each time an LSR learns a new route, the LSR allocates a new local label and then advertises the label and prefix mapping to all its neighbors—even when it is obvious that advertising the label may not be useful. For example, in Figure 19-8, P2 advertises a label for 10.3.3.0/24 back to router PE2—not terribly useful, but it is how frame-mode MPLS LSRs work.

Once the routers have all learned about a prefix using the IGP protocol, and LDP has advertised label/prefix mappings (bindings) to all other neighboring LSRs, each LSR has enough information with which to label switch packets from ingress E-LSR to egress E-LSR. For example, the same data plane process shown in Figures 19-2 and 19-3 could occur when PE1 receives an unlabeled packet destined to an address in 10.3.3.0/24. In fact, the labels advertised in Figures 19-7 and 19-8 purposefully match the earlier MPLS data plane figures (19-2 and 19-3). However, to complete the full process, you need to understand a bit more about what occurs inside an individual router, in particular, a data structure called the MPLS Label Information Base (LIB).

The MPLS Label Information Base Feeding the FIB and LFIB

Key Topic

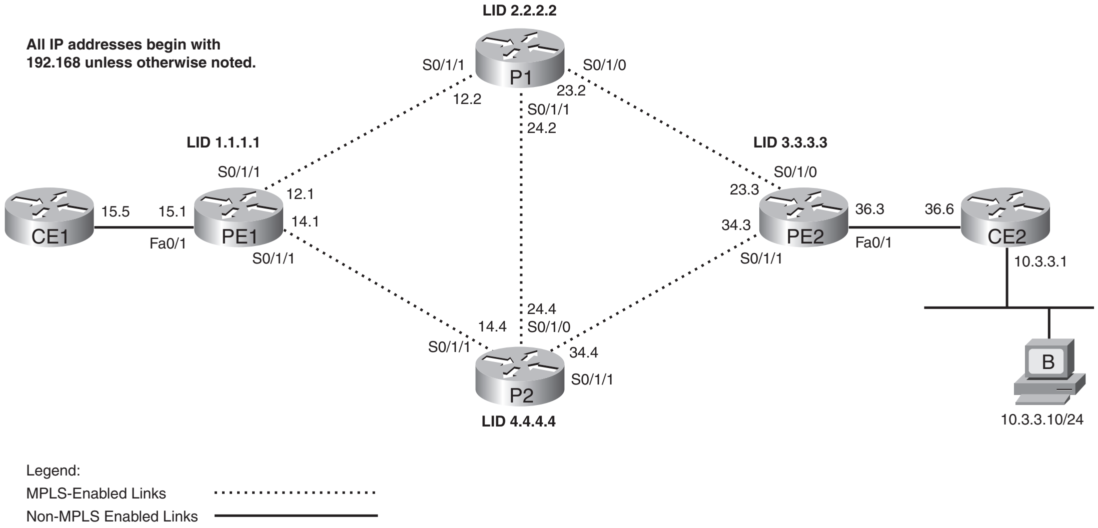
LSRs store labels and related information inside a data structure called LIB. The LIB essentially holds all the labels and associated information that could possibly be used to forward packets. However, each LSR must choose the best label and outgoing interface to actually use and then populate that information into the FIB and the LFIB. As a result, the FIB and LFIB contain labels only for the currently used best LSP segment, while the LIB contains all labels known to the LSR, whether the label is currently used for forwarding or not.

To make a decision about the best label to use, LSRs rely on the routing protocol's decision about the best route. By relying on the routing protocol, the LSRs can take advantage of the routing protocol's loop-prevention features and react to the routing protocol's choice for new routes when convergence occurs. In short, an LSR makes the following decision:

> For each route in the routing table, find the corresponding label information in the LIB, based on the outgoing interface and next-hop router listed in the route. Add the corresponding label information to the FIB and LIB.

To better understand how an LSR adds information to the FIB and LFIB, this section continues the same example as used throughout the chapter so far. At this point, it is useful to examine the output of some **show** commands, but first, you need a little more detail about the example network and the configuration. Figure 19-9 repeats the same example network used in earlier figures in this chapter, with IP address and interface details included. The figure also notes on which interfaces MPLS has been enabled (dashed lines) and on which interfaces MPLS has not been enabled (solid lines).

Figure 19-9 *Example Network for Seeing the LIB, FIB, and LFIB*



The configuration of MPLS unicast IP routing is relatively simple. In this case, all six routers use EIGRP, advertising all subnets. The four LSRs enable MPLS globally and on the links noted with dashed lines in the figure. To enable MPLS for simple unicast IP forwarding, as has been described so far in this chapter, an LSR simply needs to enable CEF, globally enable MPLS, and enable MPLS on each desired interface. Also, because IOS uses TDP instead of LDP by default, this configuration overrides the default to use LDP. Example 19-1 shows a sample generic configuration.

Example 19-1 *MPLS Configuration on LSRs for Unicast IP Support*

```
! The first three commands enable CEF and MPLS globally, and
! use LDP instead of TDP
ip cef
mpls ip
mpls label protocol ldp
!
! Repeat the next two lines for each MPLS-enabled interface
interface type x/y/z
 mpls ip
! Normal EIGRP configuration next – would be configured for all interfaces
router eigrp 1
 network …
```

To see how LSRs populate the FIB and LFIB, consider subnet 10.3.3.0/24 again, and think about MPLS from router PE1's perspective. PE1 has learned a route for 10.3.3.0/24 with EIGRP. PE1 has also learned (using LDP) about two labels that PE1 can use when forwarding packets destined for 10.3.3.0/24—one label learned from neighboring LSR P1, and the other from neighboring LSR P2. Example 19-2 highlights these details. Note that the labels do match the figures and examples used earlier in this chapter.

Example 19-2 *PE1's LIB and IP Routing Table*

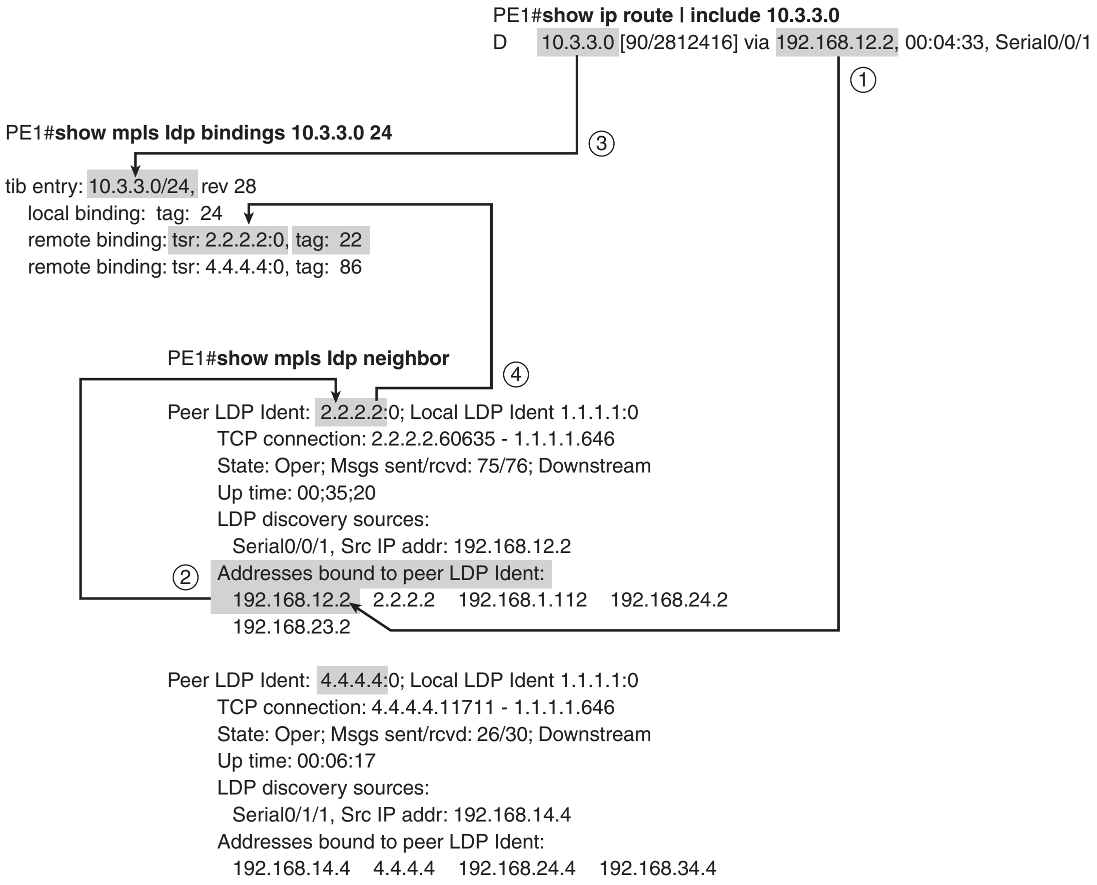
```
PE1# show ip route 10.0.0.0
Routing entry for 10.0.0.0/24, 1 known subnets
  Redistributing via eigrp 1
D       10.3.3.0 [90/2812416] via 192.168.12.2, 00:44:16, Serial0/0/1
PE1# show mpls ldp bindings 10.3.3.0 24
  tib entry: 10.3.3.0/24, rev 28
        local binding:  tag: 24
        remote binding: tsr: 2.2.2.2:0, tag: 22
        remote binding: tsr: 4.4.4.4:0, tag: 86
```

Example 19-2 shows some mundane information and a few particularly interesting points. First, the **show ip route** command does not list any new or different information for MPLS, but it is useful to note that PE1's best route to 10.3.3.0/24 is through P1. The **show ip mpls bindings 10.3.3.0 24** command lists the LIB entries from 10.3.3.0/24. Note that two remote bindings are listed—one from P1 (LDP ID 2.2.2.2) and one from P2 (LDP ID 4.4.4.4). This command also lists the local binding, which is the label that PE1 allocated and advertised to its neighbors.

NOTE The term *remote binding* refers to a label-prefix binding learned via LDP from some LDP neighbor.

From Example 19-2, you could anticipate that PE1 will use a label value of 22, and an outgoing interface of S0/0/1, when forwarding packets to 10.3.3.0/24. To see the details of how PE1 arrives at that conclusion, consider the linkages shown in Figure 19-10.

Figure 19-10 *PE1's Process to Determine the Outgoing Label*



```
PE1#show ip route | include 10.3.3.0
D     10.3.3.0 [90/2812416] via 192.168.12.2, 00:04:33, Serial0/0/1
```

```
PE1#show mpls ldp bindings 10.3.3.0 24

tib entry: 10.3.3.0/24, rev 28
  local binding:  tag:  24
  remote binding: tsr: 2.2.2.2:0, tag:  22
  remote binding: tsr: 4.4.4.4:0, tag:  86
```

```
PE1#show mpls ldp neighbor

Peer LDP Ident:  2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.60635 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 75/76; Downstream
        Up time: 00;35;20
        LDP discovery sources:
          Serial0/0/1, Src IP addr: 192.168.12.2
        Addresses bound to peer LDP Ident:
          192.168.12.2    2.2.2.2    192.168.1.112    192.168.24.2
          192.168.23.2

Peer LDP Ident:  4.4.4.4:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 4.4.4.4.11711 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 26/30; Downstream
        Up time: 00:06:17
        LDP discovery sources:
          Serial0/1/1, Src IP addr: 192.168.14.4
        Addresses bound to peer LDP Ident:
          192.168.14.4    4.4.4.4    192.168.24.4    192.168.34.4
```

The figure shows the following steps:

1. The routing table entry to 10.3.3.0/24 lists a next-hop IP address of 192.168.12.2. PE1 compares that next-hop information to the list of interface IP addresses on each LDP peer and finds the LDP neighbor who has IP address 192.168.12.2.
2. That same stanza of the **show mpls ldp neighbor** command output identifies the LDP ID (LID) of this peer, namely 2.2.2.2.
3. PE1 notes that for that same prefix (10.3.3.0/24), the LIB contains one local label and two remote labels.

4. Among the known labels listed for prefix 10.3.3.0/24, one was learned from a neighbor whose LID is 2.2.2.2, with label (tag) value of 22.

NOTE Many IOS commands still use the older tag switching terminology—for example, the term Tag Switching Router (TSR) is listed instead of LSR in Figure 19-10.

As a result of these steps, PE1 knows it should use outgoing interface S0/1/0, with label 22, when forwarding packets to subnet 10.3.3.0/24.

Examples of FIB and LFIB Entries

As mentioned earlier in the chapter, the actual packet-forwarding process does not use the IP routing table (RIB) or the LIB—instead, the FIB is used to forward packets that arrived unlabeled, and the LFIB is used to forward packets that arrived already labeled. This section correlates the information in **show** commands to the conceptual view of the FIB and LFIB data structures shown back in Figure 19-3.

First, again focusing on PE1, PE1 simply adds information to the FIB stating that PE1 should impose an MPLS header, with label value 22. PE1 also populates the LFIB, with an entry for 10.3.3.0/24, using that same label value of 22 and an outgoing interface of S0/1/0. Example 19-3 shows the contents of the two tables.

Example 19-3 *FIB and LFIB Entries for 10.3.3.0/24 on PE1*

```
! This next command shows the FIB entry, which includes the local tag (24), the
! tags (label) imposed, and outgoing interface.
PE1# show ip cef 10.3.3.0
10.3.3.0/24, version 65, epoch 0, cached adjacency to Serial0/0/1
0 packets, 0 bytes
  tag information set
    local tag: 24
    fast tag rewrite with Se0/0/1, point2point, tags imposed: {22}
  via 192.168.12.2, Serial0/0/1, 0 dependencies
    next hop 192.168.12.2, Serial0/0/1
    valid cached adjacency
    tag rewrite with Se0/0/1, point2point, tags imposed: {22}
! The next command lists the LFIB entry for 10.3.3.0/24, listing the same basic
! information—the local tag, the outgoing tag (label), and outgoing interface.
PE1# show mpls forwarding-table 10.3.3.0 24
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
24     22          10.3.3.0/24       0          Se0/0/1    point2point
```

In the data plane example of Figure 19-3, PE1 received an unlabeled packet and forwarded the packet to P1, with label 22. The information in the top part of Example 19-3, showing the FIB, matches that same logic, stating that a tag (label) value of 22 will be imposed by PE1.

Next, examine the LFIB at P1 as shown in Example 19-4. As shown in Figure 19-3, P1 swaps the incoming label of 22 with outgoing label 39. For perspective, the example also includes the LIB entries for 10.3.3.0/24.

Example 19-4 *FIB and LFIB Entries for 10.3.3.0/24 on P1*

```
P1# show mpls forwarding-table 10.3.3.0 24
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     39          10.3.3.0/24       0          Se0/1/0    point2point
P1# show mpls ldp bindings 10.3.3.0 24
  tib entry: 10.3.3.0/24, rev 30
        local binding:  tag: 22
        remote binding: tsr: 1.1.1.1:0, tag: 24
        remote binding: tsr: 4.4.4.4:0, tag: 86
        remote binding: tsr: 3.3.3.3:0, tag: 39
```

The highlighted line in the output of the **show mpls forwarding-table** command lists the incoming label (22 in this case) and the outgoing label (39). Note that the incoming label is shown under the heading “local tag,” meaning that label (tag) 22 was locally allocated by this router (P1) and advertised to other routers using LDP, as shown in Figure 19-8. P1 originally allocated and advertised label 22 to tell neighboring routers to forward packets destined to 10.3.3.0/24 to P1, with a label of 22. P1 knows that if it receives a packet with label 22, P1 should indeed swap the labels, forwarding the packet out S0/1/0 with a label of 39.

The LIB entries in Example 19-4 also reinforce the concept that (frame-mode) MPLS LSRs retain all learned labels in their LIBs, but only the currently used labels in the LFIB. The LIB lists P1’s local label (22), and the three remote labels learned from P1’s three LDP neighbors. To create the LFIB entry, P1 used the same kind of logic shown in Figure 19-10 to correlate the information in the routing table and LIB and choose a label value of 39 and outgoing interface S0/1/0 to forward packets to 10.3.3.0/24.

To see an example of the pop action, consider the LFIB for PE2, as shown in Example 19-5. When PE2 receives a labeled packet from P1 (label 39), PE2 will try to use its LFIB to forward the packet. When populating the LFIB, PE2 can easily realize that PE2 should pop the label and forward an unlabeled packet out its Fa0/1 interface. Those reasons include the fact that PE2 did not enable MPLS on Fa0/1 and that PE2 has not learned any labels from CE2. Example 19-5 shows the outgoing tag as "untagged."

Example 19-5 *FIB and LFIB Entries for 10.3.3.0/24 on PE2*

```
PE2# show mpls forwarding-table 10.3.3.0 24
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
39     Untagged    10.3.3.0/24       0          Fa0/1      192.168.36.6
```

Note that while the text in Example 19-5 only showed LFIB entries, every LSR builds the appropriate FIB and LFIB entries for each prefix, in anticipation of receiving both unlabeled and labeled packets.

Label Distribution Protocol Reference

Before wrapping up the coverage of basic MPLS unicast IP forwarding, you should know a few more details about LDP itself. So far, this chapter has shown what LDP does, but it has not provided much information about how LDP accomplishes its tasks. This section hits the main concepts and summarizes the rest.

LDP uses a Hello feature to discover LDP neighbors and to determine to what IP address the ensuing TCP connection should be made. LDP multicasts the Hellos to IP address 224.0.0.2, using UDP port number 646 for LDP (TDP uses UDP port 711). The Hellos list each LSR's LDP ID (LID), which consists of a 32-bit dotted-decimal number and a 2-byte label space number. (For frame-based MPLS, the label space number is 0.) An LSR can optionally list a *transport address* in the Hello message, which is the IP address that the LSR wants to use for any LDP TCP connections. If a router does not advertise a transport address, other routers will use the IP address that is the first 4 bytes of the LDP ID for the TCP connections.

After discovering neighbors via an LDP Hello message, LDP neighbors form a TCP connection to each neighbor, again using port 646 (TDP 711). Because the TCP connection uses unicast addresses—either the neighbor's advertised transport address or the address in the LID—these addresses must be reachable according to the IP routing table. Once the TCP connection is up, each router advertises all of its bindings of local labels and prefixes.

Cisco routers choose the IP address in the LDP ID just like the OSPF router ID. LDP chooses the IP address to use as part of its LID based on the exact same logic as OSPF, as summarized in Table 19-4, along with other details.

Table 19-4 *LDP Reference*

Key Topic

LDP Feature	LDP Implementation
Transport protocols	UDP (Hellos), TCP (updates)
Port numbers	646 (LDP), 711 (TDP)
Hello destination address	224.0.0.2
Who initiates TCP connection	Highest LDP ID
TCP connection uses this address	Transport IP address (if configured), or LDP ID if no transport address is configured
LDP ID determined by these rules, in order or precedence	Configuration Highest IP address of an up/up loopback when LDP comes up Highest IP address of an up/up non-loopback when LDP comes up

This concludes the coverage of MPLS unicast IP forwarding for this chapter. Next, the chapter examines one of the more popular uses of MPLS, which happens to use unicast IP forwarding: MPLS VPNs.

MPLS VPNs

One of the most popular of the MPLS applications is called *MPLS virtual private networks (VPNs)*. MPLS VPNs allow a service provider, or even a large enterprise, to offer Layer 3 VPN services. In particular, SPs oftentimes replace older Layer 2 WAN services such as Frame Relay and ATM with an MPLS VPN service. MPLS VPN services enable the possibility for the SP to provide a wide variety of additional services to its customers because MPLS VPNs are aware of the Layer 3 addresses at the customer locations. Additionally, MPLS VPNS can still provide the privacy inherent in Layer 2 WAN services.

MPLS VPNs use MPLS unicast IP forwarding inside the SP's network, with additional MPLS-aware features at the edge between the provider and the customer. Additionally, MPLS VPNs use MP-BGP to overcome some of the challenges when connecting an IP network to a large number of customer IP internetworks—problems that include the issue of dealing with duplicate IP address spaces with many customers.

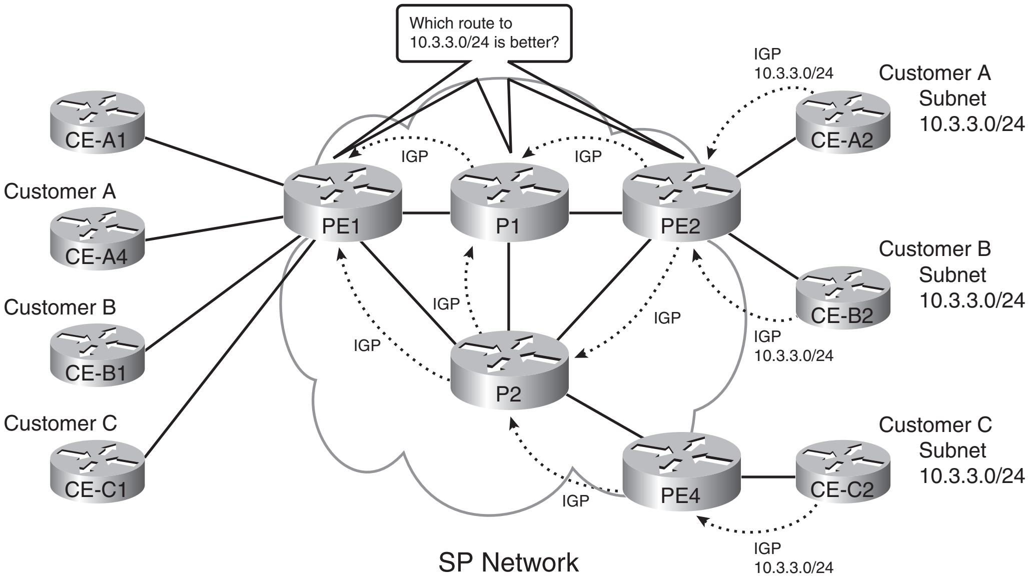
This section begins by examining some of the problems with providing Layer 3 services and then shows the core features of MPLS that solve those problems.

The Problem: Duplicate Customer Address Ranges

When an SP connects to a wide variety of customers using a Layer 2 WAN service such as Frame Relay or ATM, the SP does not care about the IP addressing and subnets used by those customers. However, in order to migrate those same customers to a Layer 3 WAN service, the SP must learn address ranges from the various customers and then advertise those routes into the SP's network. However, even if the SP wanted to know about all subnets from all its customers, many enterprises use the same address ranges—namely, the private IP network numbers, including the ever-popular network 10.0.0.0.

If you tried to support multiple customers using MPLS unicast IP routing alone, the routers would be confused by the overlapping prefixes, as shown in Figure 19-11. In this case, the network shows five of the SP's routers inside a cloud. Three customers (A, B, and C) are shown, with two customer routers connected to the SP's network. All three customers use network 10.0.0.0, with the three customer sites on the right all using subnet 10.3.3.0/24.

Figure 19-11 *The Main Challenge with Supporting Layer 3 VPNs*



The first and most basic goal for a Layer 3 VPN service is to allow customer A sites to communicate with customer A sites—and only customer A sites. However, the network in Figure 19-11 fails to meet this goal for several reasons. Because of the overlapping address spaces, several routers would be faced with the dilemma of choosing one customer's route to 10.3.3.0/24 as the best route, and ignoring the route to 10.3.3.0/24 learned from another customer. For example, PE2 would learn about two different 10.3.3.0/24 prefixes. If PE2 chooses one of the two possible routes—for example, if PE2 picked the route to CE-A2 as best—then PE2 could not forward packets to customer B's 10.3.3.0/24 off router CE-B2. Also, a possibly worse effect is that hosts in one customer site may be able to send and receive packets with hosts in another customer's network. Following this same example, hosts in customer B and C sites could forward packets to subnet 10.3.3.0/24, and the routers might forward these packets to customer A's CE-A2 router.

The Solution: MPLS VPNs

The protocols and standards defined by MPLS VPNs solve the problems shown in Figure 19-11 and provide a much larger set of features. In particular, the MPLS VPN RFCs define the concept of using multiple routing tables, called *Virtual Routing and Forwarding (VRF) tables*, which separate customer routes to avoid the duplicate address range issue. This section defines some key terminology and introduces the basics of MPLS VPN mechanics.

MPLS uses three terms to describe the role of a router when building MPLS VPNs. Note that the names used for the routers in most of the figures in this chapter have followed the convention of identifying the type of router as CE, PE, or P, as listed here.

Key Topic

- **Customer edge (CE)**—A router that has no knowledge of MPLS protocols and does not send any labeled packets but is directly connected to an LSR (PE) in the MPLS VPN.
- **Provider edge (PE)**—An LSR that shares a link with at least one CE router, thereby providing function particular to the edge of the MPLS VPN, including IBGP and VRF tables
- **Provider (P)**—An LSR that does not have a direct link to a CE router, which allows the router to just forward labeled packets, and allows the LSR to ignore customer VPNs' routes

The key to understanding the general idea of how MPLS VPNs work is to focus on the control plane distinctions between PE routers and P routers. Both P and PE routers run LDP and an IGP to support unicast IP routing—just as was described in the first half of this chapter. However, the IGP advertises routes only for subnets inside the MPLS network, with no customer routes included. As a result, the P and PE routers can together label switch packets from the ingress PE to the egress PE.

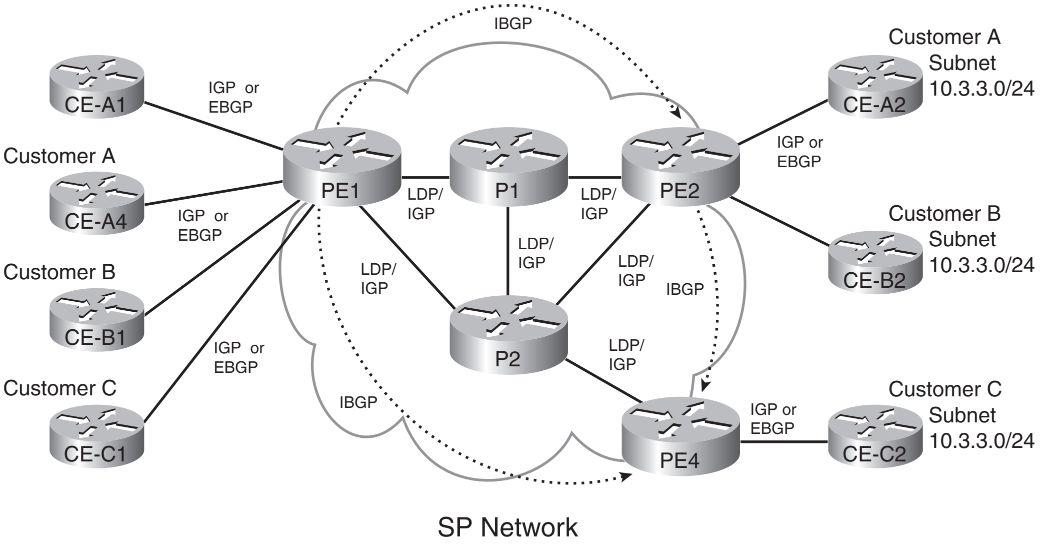
Key Topic

PEs have several other duties as well, all geared toward the issue of learning customer routes and keeping track of which routes belong to which customers. PEs exchange routes with the connected CE routers from various customers, using either EBGP, RIP-2, OSPF, or EIGRP, noting which routes are learned from which customers. To keep track of the possibly overlapping prefixes, PE routers do not put the routes in the normal IP routing table—instead, PEs store those routes in separate per-customer routing tables, called VRFs. Then the PEs use IBGP to exchange these customer routes with other PEs—never advertising the routes to the P routers. Figure 19-12 shows the control plane concepts.

> **NOTE** The term *global routing table* is used to refer to the IP routing table normally used for forwarding packets, as compared with the VRF routing tables.

Figure 19-12 *Overview of the MPLS VPN Control Plane*

Key Topic



The MPLS VPN data plane also requires more work and thought by the PE routers. The PE routers do not have any additional work to do, with one small exception, as compared with simple unicast IP routing. The extra work for the PE relates to the fact that the MPLS VPN data plane causes the ingress PE to place two labels on the packet, as follows:

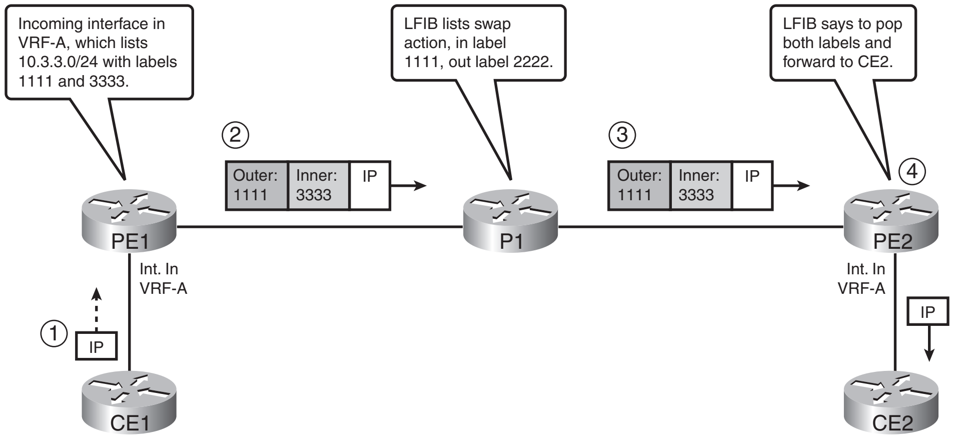
Key Topic

- An outer MPLS header (S-bit = 0), with a label value that causes the packet to be label switched to the egress PE
- An inner MPLS header (S-bit = 1), with a label that identifies the egress VRF on which to base the forwarding decision

Figure 19-13 shows a general conceptual view of the two labels and the forwarding process. The figure shows a subset of Figure 19-12, with parts removed to reduce clutter. In this case, a host in customer A on the left side of the figure sends a packet to host 10.3.3.3, located on the right side of the figure.

Figure 19-13 *Overview of the MPLS VPN Data Plane*

Key Topic


Incoming interface in VRF-A, which lists 10.3.3.0/24 with labels 1111 and 3333.
LFIB lists swap action, in label 1111, out label 2222.
LFIB says to pop both labels and forward to CE2.
2
Outer: 1111
Inner: 3333
IP
3
Outer: 1111
Inner: 3333
IP
4
PE1
P1
PE2
Int. In VRF-A
Int. In VRF-A
1
IP
IP
CE1
CE2

The figure shows the following steps:

1. CE1 forwards an unlabeled packet to PE1.
2. PE1, having received the packet in an interface assigned to VRF-A, compares the packet's destination (10.3.3.3) to the VRF-A CEF FIB, which is based on VRF-A's routing table. PE1 adds two labels based on the FIB and forwards the labeled packet.
3. P1, acting just the same as with unicast IP routing, processes the received labeled packet using its LFIB, which simply causes a label swap. P1 forwards the packet to PE2.
4. PE2's LFIB entry for label 2222 lists a pop action, causing PE2 to remove the outer label. PE2's LFIB entry for label 3333, populated based on the VRF for customer A's VPN, also lists a pop action and the outgoing interface. As a result, PE2 forwards the unlabeled packet to CE2.

NOTE In actual practice, Steps 3 and 4 differ slightly from the descriptions listed here, due to a feature called penultimate hop popping (PHP). This example is meant to show the core concepts. Figure 19-23, toward the end of this chapter, refines this logic when the router uses the PHP feature, which is on by default in MPLS VPNs.

The control plane and data plane processes described around Figures 19-12 and 19-13 outline the basics of how MPLS VPNs work. Next, the chapter takes the explanations a little deeper with a closer look at the new data structures and control plane processes that support MPLS VPNs.

The MPLS VPN Control Plane

The MPLS VPN control plane defines protocols and mechanisms to overcome the problems created by overlapping customer IP address spaces, while adding mechanisms to add more functionality to an MPLS VPN, particularly as compared to traditional Layer 2 WAN services. To understand the mechanics, you need a good understanding of BGP, IGPs, and several new concepts created by both MP-BGP RFCs and MPLS RFCs. In particular, this section introduces and explains the concepts behind three new concepts created for MPLS VPNs:

- VRFs
- Route Distinguishers (RDs)
- Route Targets (RTs)

The next several pages of text examine these topics in order. While reading the rest of the MPLS VPN coverage in this chapter, note that the text will keep expanding a single example. The example focuses on how the control plane learns about routes to the duplicate customer subnets 10.3.3.0/24 on the right side of Figure 19-12, puts the routes into the VRFs on PE2, and advertises the routes with RDs over to PE1 and then how RTs then dictate how PE1 adds the routes to its VRFs.

Virtual Routing and Forwarding Tables

To support multiple customers, MPLS VPN standards include the concept of a virtual router. This feature, called a VRF table, can be used to store routes separately for different customer VPNs. The use of separate tables solves part of the problems of preventing one customer's packets from leaking into another customer's network due to overlapping prefixes, while allowing all sites in the same customer VPN to communicate.

A VRF exists inside a single MPLS-aware router. Typically, routers need at least one VRF for each customer attached to that particular router. For example, in Figure 19-12, router PE2 connects to CE routers in customers A and B but not in customer C, so PE2 would not need a VRF for customer C. However, PE1 connects to CE routers for three customers, so PE1 will need three different VRFs.

For more complex designs, a PE might need multiple VRFs to support a single customer. Using Figure 19-12 again as an example, PE1 connects to two CEs of customer A (CE-A1 and CE-A4). If hosts near CE-A1 were allowed to access a centralized shared service (not shown in the figure) and hosts near CE-A4 were not allowed access, then PE1 would need two VRFs for customer A—one with routes for the shared service's subnets and one without those routes.

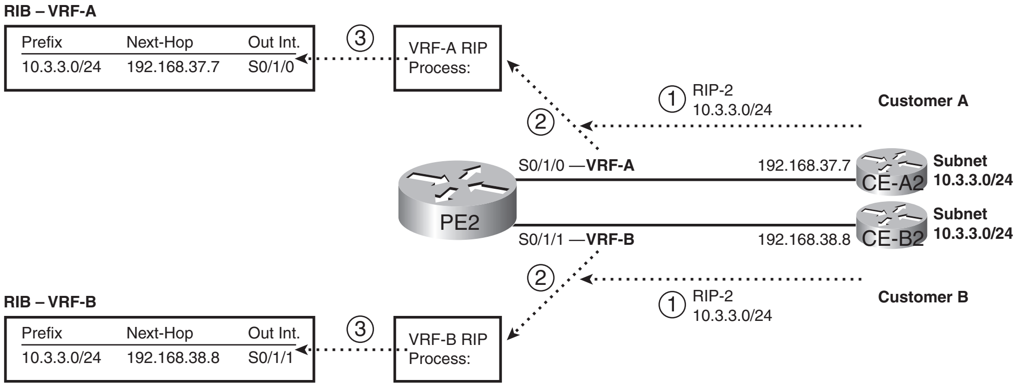
Each VRF has three main components, as follows:

Key Topic

- An IP routing table (RIB)
- A CEF FIB, populated based on that VRF's RIB
- A separate instance or process of the routing protocol used to exchange routes with the CEs that need to be supported by the VRF

For example, Figure 19-14 shows more detail about router PE2 from Figure 19-12, now with MPLS VPNs implemented. In this case, PE2 will use RIP-2 as the IGP to both customer A (router CE-A2) and customer B (router CE-B2). (The choice of routing protocol used from PE-CE is unimportant to the depth of explanations shown here.)

Figure 19-14 *Adding Routes Learned from a CE to VRFs on Router PE2*



The figure shows three parallel steps that occur with each of the two customers. Note that step 1 for each customer does not occur at the same instant in time, nor does step 2, nor step 3; the figure lists these steps with the same numbers because the same function occurs at each step. The explanation of the steps is as follows:

1. The CE router, which has no knowledge of MPLS at all, advertises a route for 10.3.3.0/24 as normal—in this case with RIP-2.
2. In the top instance of step 2, the RIP-2 update arrives on PE3's S0/1/0, which has been assigned to customer A's VRF, VRF-A. PE2 uses a separate RIP process for each VRF, so PE2's VRF-A RIP process interprets the update. Similarly, the VRF-B RIP process analyzes the update received on S0/1/1 from CE-B2.

3. In the top instance of step 3, the VRF-A RIP process adds an entry for 10.3.3.0/24 to the RIB for VRF-A. Similarly, the bottom instance of step 3 shows the RIP process for VRF-B adding a route to prefix 10.3.3.0/24 to the VRF-B RIB.

NOTE Each VRF also has a FIB, which was not included in the figure. IOS would add an appropriate FIB entry for each RIB entry.

MP-BGP and Route Distinguishers

Now that PE2 has learned routes from both CE-A2 and CE-B2, PE2 needs to advertise those routes to the other PEs, in order for the other PEs to know how to forward packets to the newly learned subnets. MPLS VPN protocols define the use of IBGP to advertise the routes—all the routes, from all the different VRFs. However, the original BGP specifications did not provide a way to deal with the fact that different customers may use overlapping prefixes.

MPLS deals with the overlapping prefix problem by adding another number in front of the original BGP NLRI (prefix). Each different number can represent a different customer, making the NLRI values unique. To do this, MPLS took advantage of a BGP RFC, called MP-BGP (RFC 4760), which allows for the re-definition of the NLRI field in BGP Updates. This re-definition allows for an additional variable-length number, called an *address family*, to be added in front of the prefix. MPLS RFC 4364, "BGP/MPLS IP Virtual Private Networks (VPNs)," defines a specific new address family to support IPv4 MPLS VPNs—namely, an MP-BGP address family called *Route Distinguishers (RDs)*.

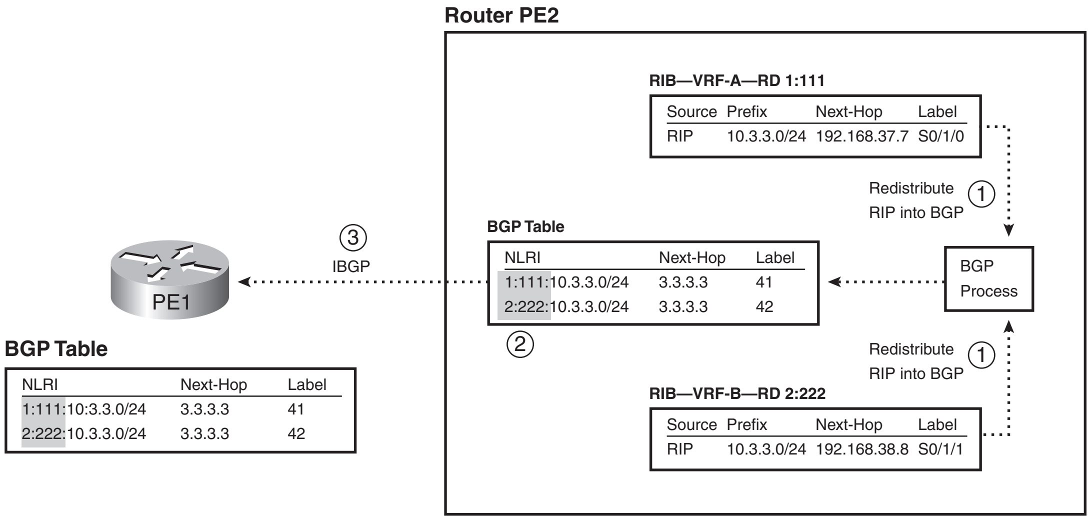
Key Topic

RDs allow BGP to advertise and distinguish between duplicate IPv4 prefixes. The concept is simple: advertise each NLRI (prefix) as the traditional IPv4 prefix, but add another number (the RD) that uniquely identifies the route. In particular, the new NLRI format, called VPN-V4, has the following two parts:

- A 64-bit RD
- A 32-bit IPv4 prefix

For example, Figure 19-15 continues the story from Figure 19-14, with router PE2 using MP-BGP to advertise its two routes for IPv4 prefix 10.3.3.0/24 to PE1—one from VRF-A and one from VRF-B. The BGP Update shows the new VPN-V4 address family format for the NLRI information, using RD 1:111 to represent VPN-A, and 2:222 to represent VPN-B.

Figure 19-15 *Making Prefixes Unique Using an RD*



RIB—VRF-A—RD 1:111

Source	Prefix	Next-Hop	Label
RIP	10.3.3.0/24	192.168.37.7	S0/1/0

BGP Table (Router PE2)

NLRI	Next-Hop	Label
1:111:10.3.3.0/24	3.3.3.3	41
2:222:10.3.3.0/24	3.3.3.3	42

RIB—VRF-B—RD 2:222

Source	Prefix	Next-Hop	Label
RIP	10.3.3.0/24	192.168.38.8	S0/1/1

BGP Table

NLRI	Next-Hop	Label
1:111:10:3.3.0/24	3.3.3.3	41
2:222:10.3.3.0/24	3.3.3.3	42

> **NOTE** PE3 uses next-hop-self, and an update source of a loopback interface with IP address 3.3.3.3.

Without the RD as part of the VPN-V4 NLRI, PE1 would have learned about two identical BGP prefixes (10.3.3.0/24) and would have had to choose one of the two as the best route—giving PE1 reachability to only one of the two customer 10.3.3.0/24 subnets. With VPN-V4 NLRI, IBGP advertises two unique NLRI—a 1:111:10.3.3.0 (from VRF-A) and 2:222:10.3.3.0 (from VRF-B). As a result, PE1 keeps both NLRI in its BGP table. The specific steps shown in the figure are explained as follows:

1. PE2 redistributes from each of the respective per-VRF routing protocol instances (RIP-2 in this case) into BGP.
2. The redistribution process pulls the RD from each respective VRF and includes that RD with all routes redistributed from the VRF's routing table.
3. PE3 uses IBGP to advertise these routes to PE1, causing PE1 to know both routes for 10.3.3.0/24, each with the differing RD values.

> **NOTE** Every VRF must be configured with an RD; the IOS **rd** VRF subcommand configures the value.

The RD itself is 8 bytes with some required formatting conventions. The first 2 bytes identify which of the three formats is followed. Incidentally, because IOS can tell which of the three formats is used based on the value, the IOS **rd** VRF subcommand only requires that you type the integer values for the last 6 bytes, with IOS inferring the first 2 bytes (the type) based on the value. The last 6 bytes, as typed in the **rd** command and seen in **show** commands, follow one of these formats:

- 2-byte-integer:4-byte-integer
- 4-byte-integer:2-byte-integer
- 4-byte-dotted-decimal:2-byte-integer

In all three cases, the first value (before the colon) should be either an ASN or an IPv4 address. The second value, after the colon, can be any value you wish. For example, you might choose an RD that lists an LSR's BGP ID using the third format, like 3.3.3.3:100, or you may use the BGP ASN, for example, 432:1.

At this point in the ongoing example, PE1 has learned about the two routes for 10.3.3.0/24—one for VPN-A and one for VPN-B—and the routes are in the BGP table. The next section describes how PE1 then chooses the VRFs into which to add these routes, based on the concept of a Route Target.

Route Targets

One of the most perplexing concepts for engineers, when first learning about MPLS VPNs, is the concept of Route Targets. Understanding the basic question of what RTs do is relatively easy, but understanding why MPLS needs RTs and how to best choose the actual values to use for RTs, can be a topic for long conversation when building an MPLS VPN. In fact, MPLS RTs enable MPLS to support all sorts of complex VPN topologies—for example, allowing some sites to be reachable from multiple VPNs, a concept called overlapping VPNs.

PEs advertise RTs in BGP Updates as BGP Extended Community path attributes (PAs). Generally speaking, BGP extended communities are 8 bytes in length, with the flexibility to be used for a wide variety of purposes. More specifically, MPLS defines the use of the BGP Extended Community PA to encode one or more RT values.

RT values follow the same basic format as the values of an RD. However, note that while a particular prefix can have only one RD, that same prefix can have one or more RTs assigned to it.

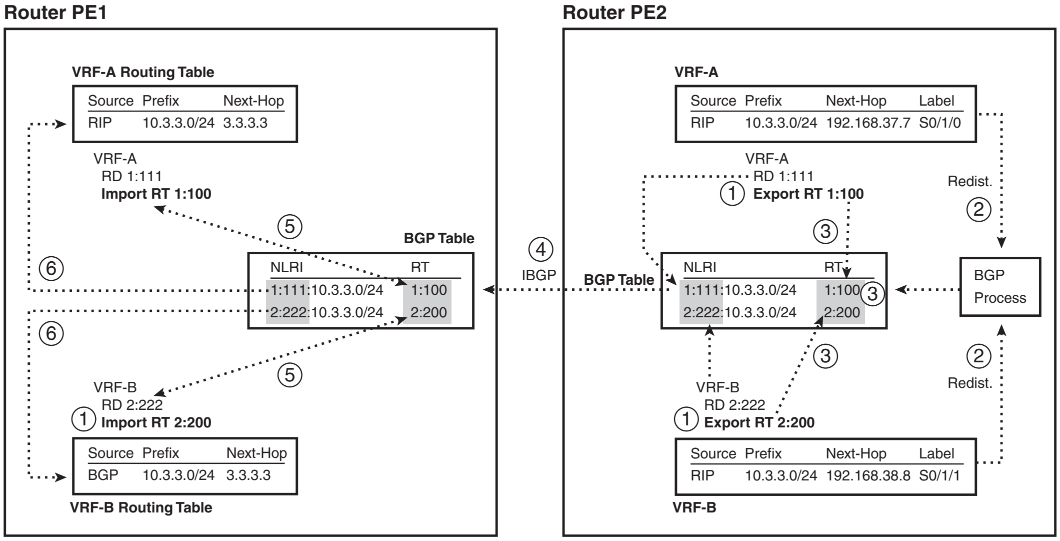
To best understand how MPLS uses RTs, first consider a more general definition of the purpose of RTs, followed by an example of the mechanics by which PEs use the RT:

Key Topic

> MPLS uses Route Targets to determine into which VRFs a PE places IBGP-learned routes.

Figure 19-16 shows a continuation of the same example in Figures 19-14 and 19-15, now focusing on how the PEs use the RTs to determine into which VRFs a route is added. In this case, the figure shows an *export RT*—a configuration setting in VRF configuration mode—with a different value configured for VRF-A and VRF-B, respectively. PE1 shows its import RT for each VRF—again a configuration setting in VRF configuration mode—which allows PE1 to choose which BGP table entries it pulls into each VRF's RIB.

Figure 19-16 *The Mechanics of the MPLS Route Target*



PE1 VRF-A Routing Table:

Source	Prefix	Next-Hop
RIP	10.3.3.0/24	3.3.3.3

PE1 BGP Table:

NLRI	RT
1:111:10.3.3.0/24	1:100
2:222:10.3.3.0/24	2:200

PE1 VRF-B Routing Table:

Source	Prefix	Next-Hop
BGP	10.3.3.0/24	3.3.3.3

PE2 VRF-A:

Source	Prefix	Next-Hop	Label
RIP	10.3.3.0/24	192.168.37.7	S0/1/0

PE2 BGP Table:

NLRI	RT
1:111:10.3.3.0/24	1:100
2:222:10.3.3.0/24	2:200

PE2 VRF-B:

Source	Prefix	Next-Hop	Label
RIP	10.3.3.0/24	192.168.38.8	S0/1/1

The figure has a lot of details, but the overall flow of concepts is not terribly difficult. Pay particular attention to the last two steps. Following the steps in the figure:

1. The two VRFs on PE2 are configured with an export RT value.
2. Redistribution out of the VRF into BGP occurs.
3. This step simply notes that the export process—the redistribution out of the VRF into BGP—sets the appropriate RT values in PE2's BGP table.
4. PE2 advertises the routes with IBGP.

5. PE1 examines the new BGP table entries and compares the RT values to the configured import RT values, which identifies which BGP table entries should go into which VRF.
6. PE1 redistributes routes into the respective VRFs, specifically the routes whose RTs match the import RT configured in the VRFs, respectively.

NOTE It is sometimes helpful to think of the term *export* to mean "redistribute out of the VRF into BGP" and the term *import* to mean "redistribute into the VRF from BGP."

Each VRF needs to export and import at least one RT. The example in Figure 19-16 shows only one direction: exporting on the right (PE2) and importing on the left (PE1). However, PE2 needs to know the routes for the subnets connected to CE-A1 and CE-B1, so PE1 needs to learn those routes from the CEs, redistribute them into BGP with some exported RT value, and advertise them to PE2 using IBGP, with PE2 then importing the correct routes (based on PE2's import RTs) into PE2's VRFs.

In fact, for simple VPN implementations, in which each VPN consists of all sites for a single customer, most configurations simply use a single RT value, with each VRF for a customer both importing and exporting that RT value.

NOTE The examples in this chapter show different numbers for the RD and RT values, so that it is clear what each number represents. In practice, you can set a VRF's RD and one of its RTs to the same value.

Overlapping VPNs

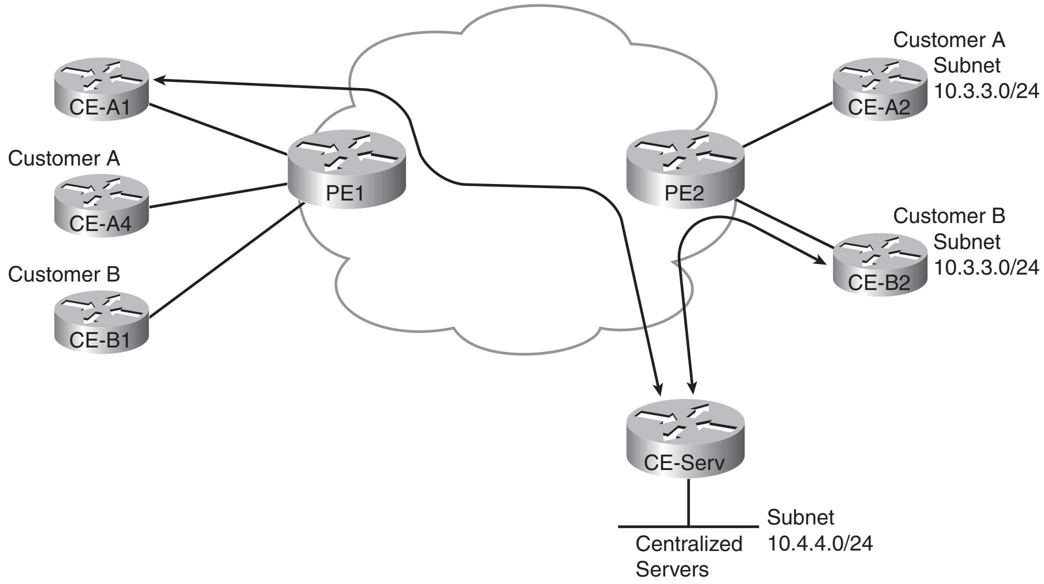
MPLS can support overlapping VPNs by virtue of the RT concept. An overlapping VPN occurs when at least one CE site needs to be reachable by CEs in different VPNs.

Many variations of overlapping VPNs exist. An SP may provide services to many customers, so the SP actually implements CE sites that need to be reached by a subset of customers. Some SP customers may want connectivity to one of their partners through the MPLS network—for example, customer A may want some of its sites to be able to send packets to some of customer B's sites.

Regardless of the business goals, the RT concept allows an MPLS network to leak routes from multiple VPNs into a particular VRF. BGP supports the addition of multiple Extended Community PAs to each BGP table entry. By doing so, a single prefix can be exported with one RT that essentially means "make sure all VRFs in VPN-A have this route," while assigning another RT value to that same prefix—an RT that means "leak this route into the VRFs of some overlapping VPN."

Figure 19-17 shows an example of the concepts behind overlapping MPLS VPNs, in particular, a design called a central services VPN. As usual, all customer A sites can send packets to all other customer A sites, and all customer B sites can send packets to all other customer B sites. Also, none of the customer A sites can communicate with the customer B sites. However, in addition to these usual conventions, CE-A1 and CE-B2 can communicate with CE-Serv, which connects to a set of centralized servers.

Figure 19-17 *Central Services VPN*



To accomplish these design goals, each PE needs several VRFs, with several VRFs exporting and importing multiple RTs. For example, PE1 needs two VRFs to support customer A—one VRF that just imports routes for customer A, and a second VRF that imports customer A routes as well as routes to reach the central services VPN. Similarly, PE2 needs a VRF for the central services VPN, which needs to import some of the routes in VPN-A and VPN-B.

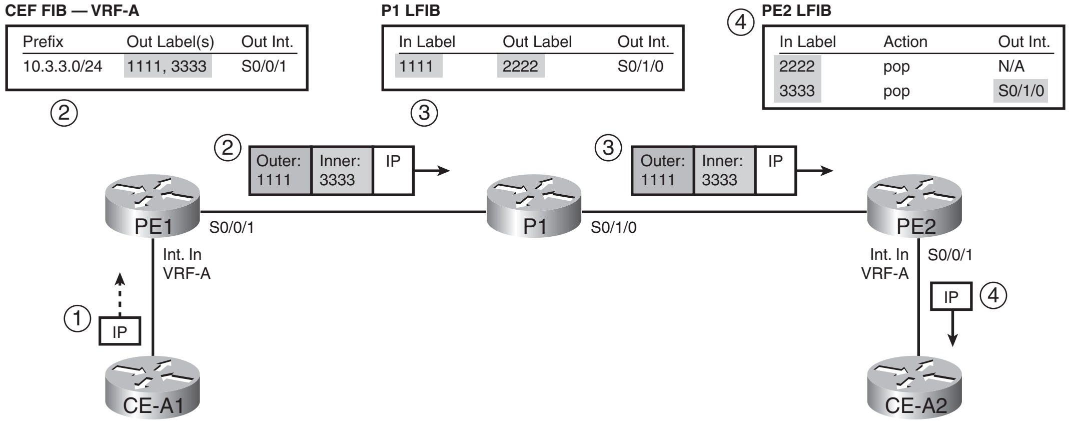
The MPLS VPN Data Plane

The explanations of the VRF, RD, and RT features explain most of the details of the MPLS VPN control plane. VRFs allow PEs to store routes learned from various CEs, even if the prefixes overlap. The RD allows PEs to advertise routes as unique prefixes, even if the IPv4 prefixes happen to overlap. Finally, the RT tells the PEs which routes should be added to each VRF, which provides greater control and the ability to allow sites to be reachable from multiple VPNs.

At the end of the process, however, to support the forwarding of packets, ingress PEs need appropriate FIB entries, with Ps and PEs needing appropriate LFIB entries. This section focuses on explaining how LSRs fill the FIB and LFIB when using MPLS VPNs.

As usual for this chapter, this section focuses on how to forward packets to subnet 10.3.3.0/24 in the customer A VPN. To begin this examination of the MPLS VPN data plane, consider Figure 19-18. This figure repeats the same forwarding example in Figure 19-13 but now shows a few details about the FIB in the ingress PE and the LFIB entries in the P and egress PE routers.

Figure 19-18 *The Ingress PE FIB and Other Routers' LFIBs*



The numbered steps in the figure are as follows:

1. An unlabeled packet arrives on an interface assigned to VRF-A, which will cause ingress PE1 to use VRF-A's FIB to make a forwarding decision.

2. Ingress PE1's VRF-A FIB entry for 10.3.3.0/24 lists an outgoing interface of S0/0/1, and a label stack with two labels—an inner label of 3333 and an outer label of 1111. So PE1 forwards the packet with these two labels pushed in front of the IP header.

3. P1 uses the LFIB entry for incoming (local) label 1111, swapping this outer label value to 2222.

4. PE2 does two LFIB lookups. PE2 finds label 2222 in the table and pops that label, leaving the inner label. Then PE2 looks up the inner label 3333 in the LFIB, noting the pop action as well, along with the outgoing interface. So PE2 forwards the unlabeled packet out interface S0/1/0.

NOTE As was the case with the example shown in Figure 19-13, the details at Steps 3 and 4 will differ slightly in practice, as a result of the PHP feature, which is explained around Figure 19-23 at the end of this chapter.

The example shows the mechanics of what happens in the data plane once the correct FIB and LFIB entries have been added. The rest of this topic about the MPLS VPN data plane examines how MPLS VPN LSRs build these correct entries. While reading this section, it is helpful to keep in mind a couple of details about the purpose of the inner and outer label used for MPLS VPNs:

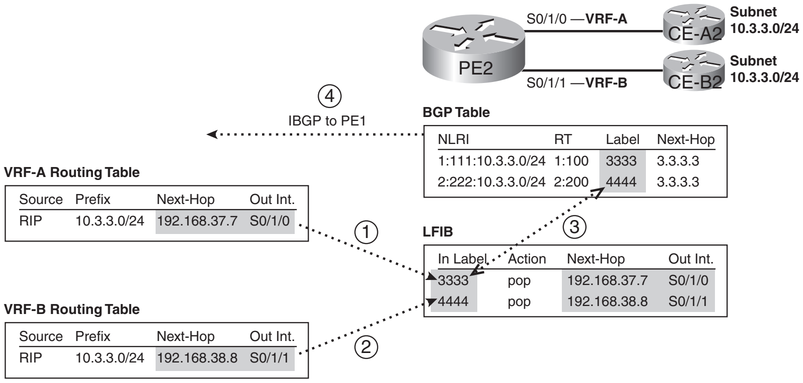
- The outer label identifies the segments of the LSP between the ingress PE and the egress PE, but it does not identify how the egress PE should forward the packet.
- The inner label identifies the egress PE's forwarding details, in particular the outgoing interface for the unlabeled packet.

Building the (Inner) VPN Label

The inner label identifies the outgoing interface out which the egress PE should forward the unlabeled packet. This inner label, called the *VPN label*, must be allocated for each route added to each customer VRF. More specifically, a customer CE will advertise routes to the PE, with the PE storing those routes in that customer's VRF. In order to prepare to forward packets to those customer subnets, the PE needs to allocate a new local label, associate the label with the prefix (and the route's next-hop IP address and outgoing interface), and store that information in the LFIB.

Figure 19-19 shows PE2's routes for 10.3.3.0/24 in both VRF-A and VRF-B and the resulting LFIB entries. The figure shows the results of PE2's process of allocating a local label for each of the two routes and then also advertising those labels using BGP. (Note that the LFIB is not a per-VRF table; the LFIB is the one and only LFIB for PE2.)

Figure 19-19 *Creating the VPN Label LFIB Entry on the Egress PE*



BGP Table

NLRI	RT	Label	Next-Hop
1:111:10.3.3.0/24	1:100	3333	3.3.3.3
2:222:10.3.3.0/24	2:200	4444	3.3.3.3

VRF-A Routing Table

Source	Prefix	Next-Hop	Out Int.
RIP	10.3.3.0/24	192.168.37.7	S0/1/0

LFIB

In Label	Action	Next-Hop	Out Int.
3333	pop	192.168.37.7	S0/1/0
4444	pop	192.168.38.8	S0/1/1

VRF-B Routing Table

Source	Prefix	Next-Hop	Out Int.
RIP	10.3.3.0/24	192.168.38.8	S0/1/1

The steps shown in the figure are as follows:

1. After adding a route for 10.3.3.0/24 to VRF-A, PE2 allocates a local label (3333) to associate with the route. PE2 then stores the local label and corresponding next hop and outgoing interface from VRF-A's route for 10.3.3.0/24 into the LIB (not shown) and LFIB.
2. PE2 repeats the logic in Step 1 for each route in each VRF, including the route in VRF-B shown at Step 2. After learning a route for 10.3.3.0/24 in VRF-B, PE2 allocates a different label value (4444), associates that route's next-hop IP address and outgoing interface with the new label, and adds the information to a new LFIB entry.
3. PE2 adds the local labels to the BGP table entry for the routes, respectively, when redistributing routes into BGP.
4. PE2 uses IBGP to advertise the routes to PE1, with the BGP Update including the VPN label.

As a result of the first two steps in the figure, if PE3 receives a labeled packet and analyzes a label value of 3333, PE2 would be able to forward the packet correctly to CE-A2. Similarly, PE2 could correctly forward a received labeled packet with label 4444 to CE-B2.

NOTE Steps 3 and 4 in Figure 19-19 do nothing to aid PE2 to forward packets; these steps were included to be referenced at an upcoming step later in this section.

Creating LFIB Entries to Forward Packets to the Egress PE

The outer label defines the LSP from the ingress PE to the egress PE. More specifically, it defines an LSP used to forward packets to the BGP next-hop address as advertised in BGP Updates. In concept, the ingress PE adds the outer label to make a request of the core of the MPLS network to "deliver this packet to the egress PE—which advertised this particular BGP next-hop address."

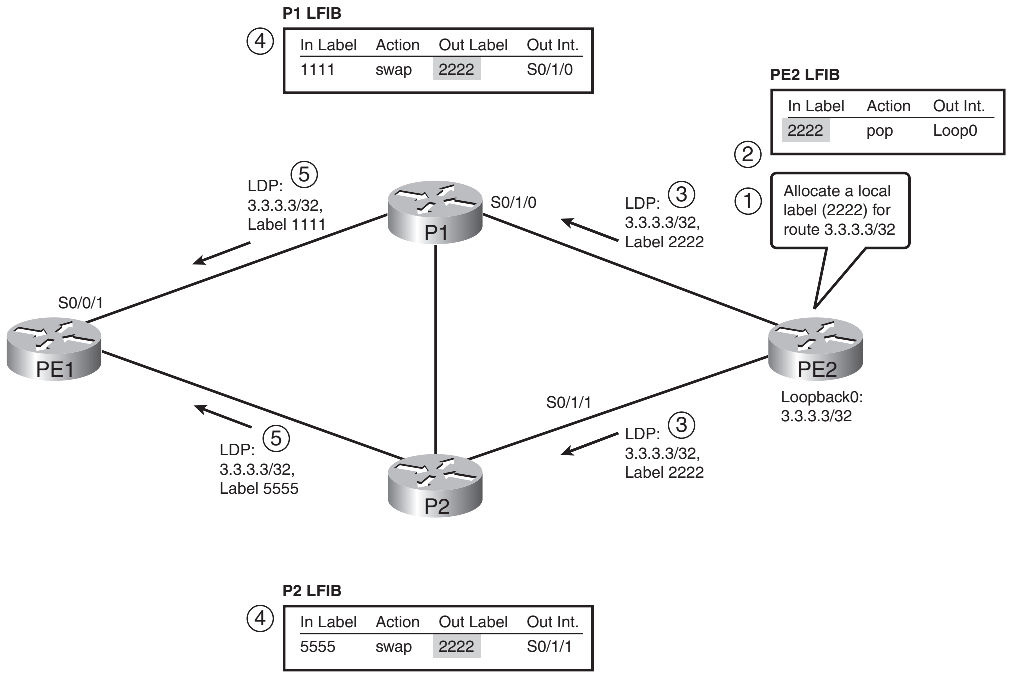
MPLS VPNs use an IGP and LDP to learn routes and labels, specifically to learn the label values to use in the outer label. To link the concepts together, it can be helpful to think of the full control plane process related to the LSP used for the outer label, particularly Step 4 onward:

1. A PE, which will be an egress PE for this particular route, learns routes from some CE.
2. The egress PE uses IBGP to advertise the routes to an ingress PE.
3. The learned IBGP routes list some next-hop IP address.
4. For MPLS VPNs to work, the PE and P routers must have advertised a route to reach the BGP next-hop addresses.
5. Likewise, for MPLS VPNs to work, the PE and P routers must have advertised labels with LDP for the routes to reach the BGP next-hop addresses.

6. Each P and PE router adds its part of the full end-to-end LSP into its LFIB, supporting the ingress PE's ability to send a packet to the egress PE.

For example, Figure 19-19 shows PE2 advertising two routes to PE1, both with BGP next-hop IP address 3.3.3.3. For MPLS to work, the collective PE and P routers need to advertise an IGP route to reach 3.3.3.3, with LDP advertising the labels, so that packets can be label switched toward the egress PE. Figure 19-20 shows the basic process; however, note that this part of the process works exactly like the simple IGP and LDP process shown for unicast IP forwarding in the first half of this chapter.

Figure 19-20 *Creating the LFIB Entries to Reach the Egress PE's BGP Next Hop*



P1 LFIB

In Label	Action	Out Label	Out Int.
1111	swap	2222	S0/1/0

PE2 LFIB

In Label	Action	Out Int.
2222	pop	Loop0

P2 LFIB

In Label	Action	Out Label	Out Int.
5555	swap	2222	S0/1/1

4
2
1
Allocate a local label (2222) for route 3.3.3.3/32
5
LDP: 3.3.3.3/32, Label 1111
P1
S0/1/0
3
LDP: 3.3.3.3/32, Label 2222
S0/0/1
PE1
PE2
Loopback0: 3.3.3.3/32
S0/1/1
3
LDP: 3.3.3.3/32, Label 2222
5
LDP: 3.3.3.3/32, Label 5555
P2
4

The steps in the figure focus on the LFIB entries for prefix 3.3.3.3/32, which matches PE2's BGP next-hop IP address, as follows. Note that the figure does not show all LDP advertisements but only those that are particularly interesting to the example.

1. PE2, upon learning a route for prefix 3.3.3.3/32, allocates a local label of 2222.
2. PE2 updates its LFIB for the local label, listing a pop action.
3. As normal, PE2 advertises to its LDP neighbors the label binding of prefix 3.3.3.3/32 with label 2222.

4. P1 and P2 both independently learn about prefix 3.3.3.3/32 with the IGP, allocate a local label (1111 on P1 and 5555 on P2), and update their LFIBs.
5. P1 and P2 advertise the binding of 3.3.3.3/32, along with their respective local labels, to their peers.

Figure 19-18 showed the FIB and LFIB entries required for forwarding a packet from CE-A1 to CE-A2, specifically into subnet 10.3.3.0/24. Figures 19-19 and 19-20, and their associated text, explained how all the LFIB entries were created. Next, the focus turns to the FIB entry required on PE1.

Creating VRF FIB Entries for the Ingress PE

The last part of the data plane analysis focuses on the ingress PE. In particular, the ingress PE uses the following logic when processing an incoming unlabeled packet:

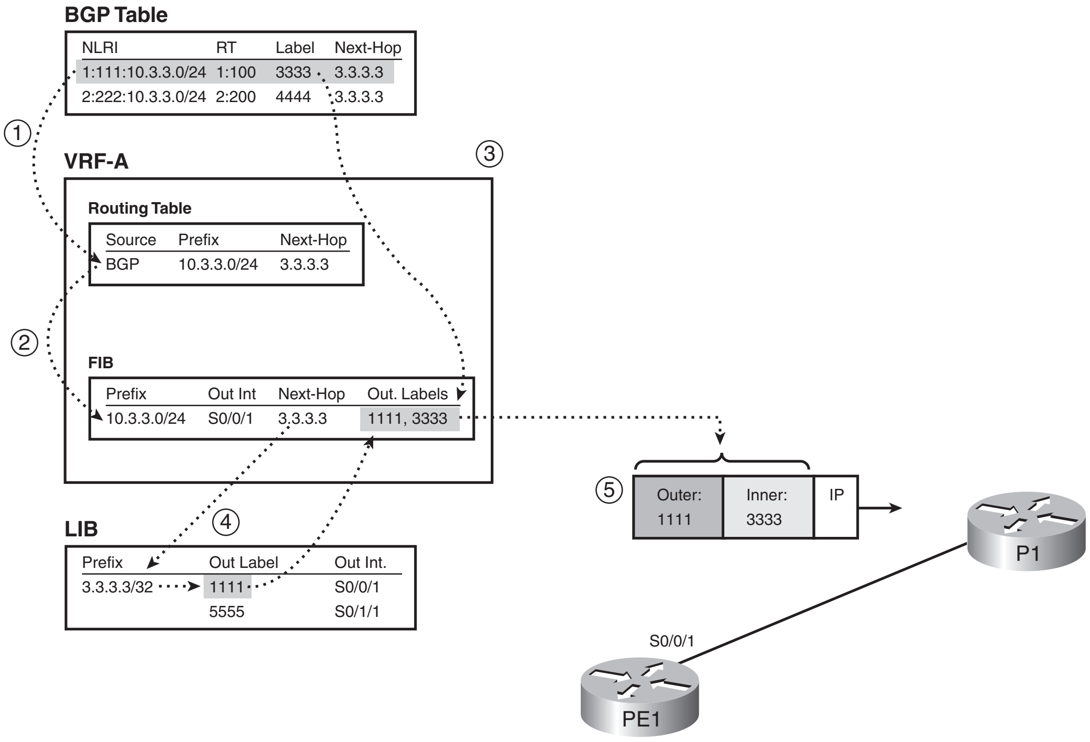
1. Process the incoming packet using the VRF associated with the incoming interface (statically configured).
2. Forward the packet using that VRF's FIB.

The FIB entry needs to have two labels to support MPLS VPNs: an outer label that identifies the LSP with which to reach the egress PE, and an inner label that identifies the egress PE's LFIB entry that includes the correct outgoing interface on the egress PE. Although it might be obvious by now, for completeness, the ingress PE learns the outer and inner label values as follows:

- The outer label is based on the LIB entry, specifically for the LIB entry for the prefix that matches the BGP-learned next-hop IP address—not the packet's destination IP address.
- The inner label is based on the BGP table entry for the route in the VRF that matches the packet's destination address.

Figure 19-21 completes the ongoing example by showing the process by which PE1 adds the correct FIB entry into VRF-A for the 10.3.3.0/24 prefix. The figure picks up the story at the point at which PE1 has learned all required BGP and LDP information, and it is ready to populate the VRF routing table and FIB.

Figure 19-21 *Creating the Ingress PE (PE1) FIB Entry for VRF-A*



BGP Table

NLRI	RT	Label	Next-Hop
1:111:10.3.3.0/24	1:100	3333	3.3.3.3
2:222:10.3.3.0/24	2:200	4444	3.3.3.3

VRF-A

Routing Table

Source	Prefix	Next-Hop
BGP	10.3.3.0/24	3.3.3.3

FIB

Prefix	Out Int	Next-Hop	Out. Labels
10.3.3.0/24	S0/0/1	3.3.3.3	1111, 3333

LIB

Prefix	Out Label	Out Int.
3.3.3.3/32	1111	S0/0/1
	5555	S0/1/1

PE1's BGP table holds the VPN label (3333), while PE1's LIB holds the two labels learned from PE1's two LDP neighbors (P1 and P2, labels 2222 and 5555, respectively). In this case, PE1's best route that matches BGP next-hop 3.3.3.3 happens to point to P1 instead of P2, so this example uses label 1111, learned from P1.

The steps in the figure are explained as follows:

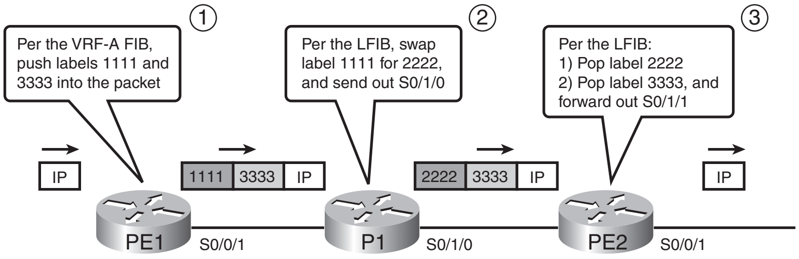
1. PE1 redistributes the route from BGP into the VRF-A routing table (based on the import RT).
2. PE1 builds a VRF-A FIB entry for the route just added to the VRF-A routing table.
3. This new FIB entry needs to include the VPN-label, which PE1 finds in the associated BGP table entry.
4. This new FIB entry also needs to include the outer label, the one used to reach the BGP next-hop IP address (3.3.3.3), so PE1 looks in the LIB for the best LIB entry that matches 3.3.3.3, and extracts the label (1111).
5. Ingress PE1 inserts the MPLS header including the two-label label stack.

At this point, when PE1 receives a packet in an interface assigned to VRF-A, PE1 will look in the VRF-A FIB. If the packet is destined for an address in prefix 10.3.3.0/24, PE1 will match the entry shown in the figure, and PE1 will forward the packet out S0/0/1, with labels 1111 and 3333.

Penultimate Hop Popping

The operation of the MPLS VPN data plane works well, but the process on the egress PE can be a bit inefficient. The inefficiency relates to the fact that the egress PE must do two lookups in the LFIB after receiving the packet with two labels in the label stack. For example, the data plane forwarding example used throughout this chapter has been repeated in Figure 19-22, with a summary description of the processing logic on each router. Note that the egress PE (PE2) must consider two entries in its LFIB.

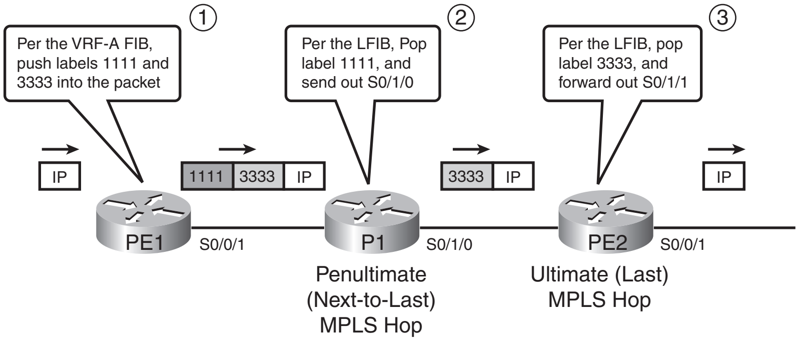
Figure 19-22 *Two LFIB Lookups Required on the Egress PE*


1
Per the VRF-A FIB, push labels 1111 and 3333 into the packet
2
Per the LFIB, swap label 1111 for 2222, and send out S0/1/0
3
Per the LFIB:
1) Pop label 2222
2) Pop label 3333, and forward out S0/1/1
IP
1111 3333 IP
2222 3333 IP
IP
PE1
S0/0/1
P1
S0/1/0
PE2
S0/0/1

To avoid this extra work on the very last (ultimate) LSR, MPLS uses a feature called *penultimate hop popping (PHP)*. (*Penultimate* simply means "1 less than the ultimate.") So the penultimate hop is not the very last LSR to process a labeled packet, but the second-to-last LSR to process a labeled packet. PHP causes the penultimate-hop LSR to pop the outer label, so that the last LSR—the ultimate hop if you will—receives a packet that only has the VPN label in it. With only this single label, the egress PE needs to look up only one entry in the LFIB. Figure 19-23 shows the revised data plane flow with PHP enabled.

Figure 19-23 *Single LFIB Lookup on Egress PE Due to PHP*

Key Topic


1
Per the VRF-A FIB, push labels 1111 and 3333 into the packet
2
Per the LFIB, Pop label 1111, and send out S0/1/0
3
Per the LFIB, pop label 3333, and forward out S0/1/1
IP
1111 3333 IP
3333 IP
IP
PE1
S0/0/1
P1
S0/1/0
PE2
S0/0/1
Penultimate (Next-to-Last) MPLS Hop
Ultimate (Last) MPLS Hop

Other MPLS Applications

This last relatively short section of the chapter introduces the general idea about the protocols used by several other MPLS applications. To that end, this section introduces and explains the concept of a *Forwarding Equivalence Class (FEC)* and summarizes the concept of an FEC as used by various MPLS applications.

Key Topic

Frankly, this chapter has already covered all the concepts surrounding the term FEC. However, it is helpful to know the term and the FEC concept as an end to itself, because it helps when comparing various MPLS applications.

Generally speaking, an FEC is a set of packets that receives the same forwarding treatment by a single LSR. For simple MPLS unicast IP forwarding, each IPv4 prefix is an FEC. For MPLS VPNs, each prefix in each VRF is an FEC—making the prefix 10.3.3.0/24 in VRF-A a different FEC from the 10.3.3.0/24 prefix in VRF-B. Alternately, with QoS implemented, one FEC might be the set of packets in VRF-A, destined to 10.3.3.0/24, with DSCP EF in the packet, and another FEC might be packets in the same VPN, to the same subnet, but with a different DSCP value.

For each FEC, each LSR needs a label, or label stack, to use when forwarding packets in that FEC. By using a unique label or set of labels for each FEC, a router has the ability to assign different forwarding details (outgoing interface and next-hop router.)

Each of the MPLS applications can be compared by focusing on the information used to determine an FEC. For example, MPLS traffic engineering (TE) allows MPLS networks to choose to send some packets over one LSP and other packets over another LSP, based on traffic loading—even though the true end destination might be in the same location. By doing so, SPs can manage the flow of data over their high-speed core networks and prevent the problem of overloading the best route as determined by a routing protocol, while barely using alternate routes. To achieve this function, MPLS TE bases the FEC concept in part on the definition of an MPLS TE tunnel.

You can also compare different MPLS applications by listing the control plane protocols used to learn label information. For example, this chapter explained how MPLS VPN uses both LDP and MP-BGP to exchange label information, whereas other MPLS applications use LDP and something else—or do not even use LDP at all. Table 19-5 lists many of the common MPLS applications, the information that determines an FEC, and the control plane protocol that is used to advertise FEC-to-label bindings.

Table 19-5 *Control Protocols Used in Various MPLS Applications*

Key Topic

Application	**FEC**	**Control Protocol Used to Exchange FEC-to-Label Binding**
Unicast IP routing	Unicast IP routes in the global IP routing table	Tag Distribution Protocol (TDP) or Label Distribution Protocol (LDP)
Multicast IP routing	Multicast routes in the global multicast IP routing table	PIM version 2 extensions
VPN	Unicast IP routes in the per-VRF routing table	MP-BGP
Traffic engineering	MPLS TE tunnels (configured)	RSVP or CR-LDP
MPLS QoS	IP routing table and the ToS byte	Extensions to TDP and LDP

Foundation Summary

Please take the time to read and study the details in the "Foundation Topics" section of the chapter, as well as review the items noted with a Key Topic icon.

Memory Builders

The CCIE Routing and Switching written exam, like all Cisco CCIE written exams, covers a fairly broad set of topics. This section provides some basic tools to help you exercise your memory about some of the broader topics covered in this chapter.

Fill in Key Tables from Memory

Appendix E, "Key Tables for CCIE Study," on the CD in the back of this book contains empty sets of some of the key summary tables in each chapter. Print Appendix E, refer to this chapter's tables in it, and fill in the tables from memory. Refer to Appendix F, "Solutions for Key Tables for CCIE Study," on the CD to check your answers.

Definitions

Next, take a few moments to write down the definitions for the following terms:

> FIB, LIB, LFIB, MPLS unicast IP routing, MPLS VPNs, LDP, TDP, LSP, LSP segment, MPLS TTL propagation, local label, remote label, label binding, VRF, RD, RT, overlapping VPN, inner label, outer label, VPN label, PHP, FEC, LSR, E-LSR, PE, CE, P, ingress PE, egress PE

Refer to the glossary to check your answers.

Further Reading

Cisco Press publishes a wide variety of MPLS books, which can be found at http://www.ciscopress.com. Additionally, you can see a variety of MPLS pages from http://www.cisco.com/go/mpls.