What You Will Learn

After completing this chapter, you will be able to do the following:

- Explain how labeled packets are forwarded
- Name the reserved MPLS labels and know what they are used for
- Determine the importance of MPLS MTU in MPLS networks
- Explain what happens to labeled packets that have TTL expiring
- Explain what happens with labeled packets that need to be fragmented

CHAPTER 3

Forwarding Labeled Packets

Chapter 2, “MPLS Architecture,” focused on what an MPLS label is and how it is used. This chapter specifically focuses on how labeled packets are forwarded. Forwarding labeled packets is quite different from forwarding IP packets. Not only is the IP lookup replaced with a lookup of the label in the label forwarding information base (LFIB), but different label operations are also possible. These operations refer to the pop, push, and swap operations of MPLS labels in the label stack.

When reading this chapter, note the existence of the reserved MPLS labels that have a special function. These reserved labels are already introduced here, because they are mentioned throughout the book.

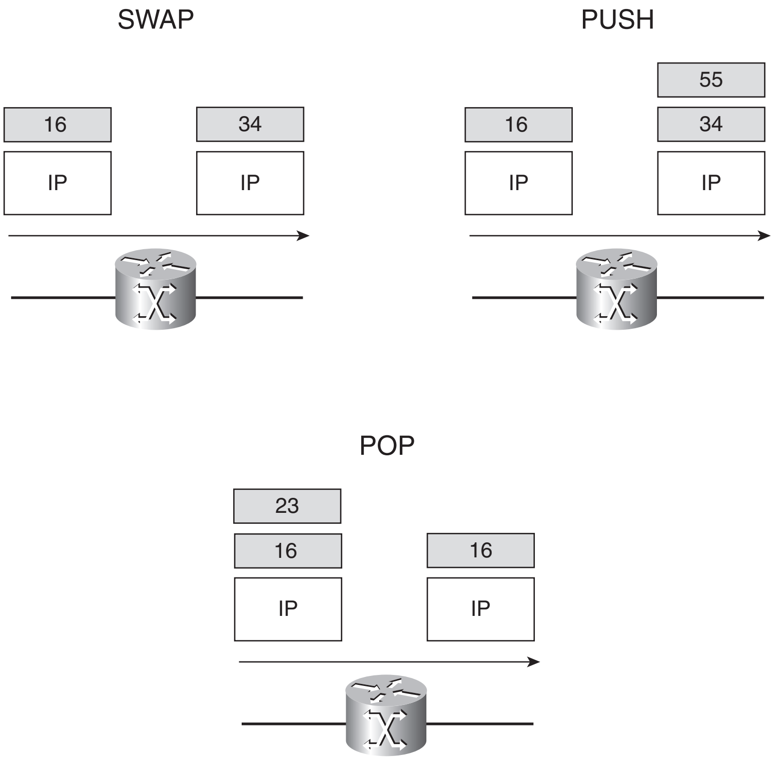
Forwarding of Labeled Packets

This section looks at how labeled packets are forwarded in MPLS networks, how forwarding labeled packets is different from forwarding IP packets, how labeled packets are load-balanced, and what a label switching router (LSR) does with a packet with an unknown label.

Label Operation

The possible label operations are swap, push, and pop. Look at Figure 3-1 to see the possible operations on labels.

Figure 3-1 *Operations on Labels*



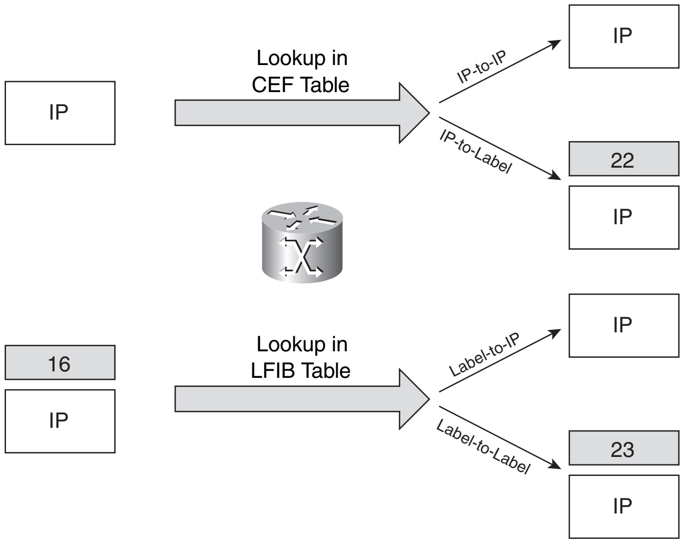
By looking at the top label of the received labeled packet and the corresponding entry in the LFIB, the LSR knows how to forward the packet. The LSR determines what label operation needs to be performed—swap, push, or pop—and what the next hop is to which the packet needs to be forwarded. The swap operation means that the top label in the label stack is replaced with another, and the push operation means that the top label is replaced with another and then one or more additional labels are pushed onto the label stack. The pop operation means that the top label is removed.

NOTE The LSR sees the 20-bit field in the top label, looks up this value in the LFIB, and tries to match it with a value in the local labels list.

IP Lookup Versus Label Lookup

When a router receives an IP packet, the lookup done is an IP lookup. In Cisco IOS, this means that the packet is looked up in the CEF table. When a router receives a labeled packet, the lookup is done in the LFIB of the router. The router knows that it receives a labeled packet or an IP packet by looking at the protocol field in the Layer 2 header. If a packet is forwarded by either Cisco Express Forwarding (CEF) (IP lookup) or by LFIB (label lookup), the packet can leave the router either labeled or unlabeled. Look at Figure 3-2 to see the difference between a lookup in the CEF table and in the LFIB.

Figure 3-2 *CEF or LFIB Lookup*



If an ingress LSR receives an IP packet and forwards it as labeled, it is called the IP-to-label forwarding case. If an LSR receives a labeled packet, it can strip off the labels and forward it as an IP packet, or it can forward it as a labeled packet. The first case is referred to as the label-to-IP forwarding case; the second is referred to as the label-to-label forwarding case.

NOTE For more information on CEF and its interaction with MPLS, refer to Chapter 6, "Cisco Express Forwarding."

Example 3-1 shows an IP-to-label forwarding case—that is, the forwarding of an IP packet by the CEF table.

Example 3-1 *Example of an Entry in the CEF table*

```
lactometer#show ip cef 10.200.254.4
10.200.254.4/32, version 44, epoch 0, cached adjacency 10.200.200.2
0 packets, 0 bytes
  tag information set, all rewrites owned
```

continues

Example 3-1 *Example of an Entry in the CEF table (Continued)*

```
    local tag: 20
    fast tag rewrite with Et0/0/0, 10.200.200.2, tags imposed {18}
  via 10.200.200.2, Ethernet0/0/0, 0 dependencies
    next hop 10.200.200.2, Ethernet0/0/0
    valid cached adjacency
    tag rewrite with Et0/0/0, 10.200.200.2, tags imposed {18}
```

IP packets that enter the LSR destined for 10.200.254.4/32 go out on interface Ethernet0/0/0 after being imposed with the label 18. The next hop of this packet is 10.200.200.2. The IP-to-label forwarding is done at the imposing LSR. In Cisco IOS, CEF switching is the only IP switching mode that you can use to label packets. Other IP switching modes, such as fast switching, cannot be used, because the fast switching cache does not hold information on labels. Because CEF switching is the only IP switching mode that is supported in conjunction with MPLS, you must turn on CEF when you enable MPLS on the router.

In Example 3-2, you can see an extract from the LFIB, by issuing the command **show mpls forwarding-table**.

Example 3-2 *Extract of the LFIB*

```
lactometer#show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Untagged    10.1.1.0/24       0          Et0/0/0    10.200.200.2
17     16          10.200.202.0/24   0          Et0/0/0    10.200.200.2
18     Pop tag     10.200.203.0/24   0          Et0/0/0    10.200.200.2
19     Pop tag     10.200.201.0/24   0          Et0/0/0    10.200.200.2
20     18          10.200.254.4/32   0          Et0/0/0    10.200.200.2
21     Pop tag     10.200.254.2/32   0          Et0/0/0    10.200.200.2
22     17          10.200.254.3/32   0          Et0/0/0    10.200.200.2
24     Untagged    l2ckt(100)        4771050    Fa9/0/0    point2point
```

The local label (or tag) is the label that this LSR assigns and distributes to the other LSRs. As such, this LSR expects labeled packets to come to it with these labels as the top ones in the label stack. If this LSR were to receive a labeled packet with the top label 22, it would swap the label with label 17 and then forward it on the Ethernet0/0/0 interface. This is an example of the label-to-label forwarding case.

If this LSR receives a packet with top label 16, it removes all labels and forwards the packet as an IP packet, because the outgoing label (tag) is Untagged. This is an example of the label-to-IP case. If the LSR receives a packet with top label 18, it removes the top label (pop one label) and forwards the packet as a labeled packet or as an IP packet. You can see in this output some examples of the swap and pop operation. Example 3-3 shows an example of a push operation. The incoming label 23 is swapped with label 20, and label 16 is pushed onto label 20.

Example 3-3 *Example of Show MPLS Forwarding-Table (Detail)*

```
lactometer#show mpls forwarding-table 10.200.254.4
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     16      [T] 10.200.254.4/32   0          Tu1        point2point

[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option

lactometer#show mpls forwarding-table 10.200.254.4 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     16          10.200.254.4/32   0          Tu1        point2point
        MAC/Encaps=14/22, MRU=1496, Tag Stack{20 16}, via Et0/0/0
        00604700881D00024A4008008847 0001400000010000
        No output feature configured
```

To see all the labels that change on an already labeled packet, you must use the show mpls **forwarding-table** [*network* {*mask* | *length*}] [**detail**] command. In Example 3-3, you can see the difference between the output of this command with and without the **detail** keyword. If the **detail** keyword is specified, you can see all the labels that change in the label stack. From left to right between { }, you see the first label, which is the swapped label (20), and then the pushed label (16) onto the swapped label. Without the **detail** keyword, you see only the pushed label (16).

The aggregate operation remains. When you perform an aggregation (or summarization) on an LSR, it advertises a specific label for the aggregated prefix, but the outgoing label in the LFIB shows "Aggregate." Because this LSR is aggregating a range of prefixes, it cannot forward an incoming labeled packet by label-swapping the top label. The outgoing label entry showing "Aggregate" means that the aggregating LSR needs to remove the label of the incoming packet and must do an IP lookup to determine the more specific prefix to use for forwarding this IP packet. Example 3-4 shows an entry in the LFIB on an egress PE router in an MPLS VPN network.

The egress LSR receiving a packet with label 23 would remove that label and perform an IP lookup on the destination IP address in the IP header.

Example 3-4 *Example of an Entry in the LFIB for an MPLS VPN Prefix*

```
singularity#show mpls forwarding-tablevrf cust-one
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     Aggregate   10.10.1.0/24[V]   0
```

You know now how the labeled packet is forwarded to a specific next hop after a label operation. The CEF adjacency table, however, determines the outgoing data link encapsulation. The adjacency table provides the necessary Layer 2 information to forward the packet to the next-hop LSR. This is explained in greater detail in Chapter 6.

Example 3-5 shows an adjacency table on an LSR. The adjacency table holds the Layer 2 information needed to switch out a frame on the outgoing data link.

Example 3-5 *Example of an Adjacency Table*

```
lactometer#show adjacency detail
Protocol Interface                 Address
IP       Ethernet0/0/0             10.200.200.2(13)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 4
                                   Encap length 14
                                   00604700881D00024A4008000800
                                   ARP
TAG      Ethernet0/0/0             10.200.200.2(9)
                                   231 packets, 22062 bytes
                                   epoch 0
                                   sourced in sev-epoch 4
                                   Encap length 14
                                   00604700881D00024A4008008847
                                   ARP
IP       Serial0/1/0               point2point(10)
                                   258 packets, 35612 bytes
                                   epoch 0
                                   sourced in sev-epoch 4
                                   Encap length 4
                                   0F000800
                                   P2P-ADJ
```

Example 3-5 *Example of an Adjacency Table (Continued)*

```
TAG      Serial0/1/0               point2point(5)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 4
                                   Encap length 4
                                   0F008847
                                   P2P-ADJ
```

To recap the label operations:

- **Pop**—The top label is removed. The packet is forwarded with the remaining label stack or as an unlabeled packet.
- **Swap**—The top label is removed and replaced with a new label.
- **Push**—The top label is replaced with a new label (swapped), and one or more labels are added (pushed) on top of the swapped label.
- **Untagged/No Label**—The stack is removed, and the packet is forwarded unlabeled.
- **Aggregate**—The label stack is removed, and an IP lookup is done on the IP packet.

Load Balancing Labeled Packets

If multiple equal-cost paths exist for an IPv4 prefix, the Cisco IOS can load-balance labeled packets, as illustrated in the Cisco IOS output of Example 3-6. You can see that the incoming/local labels 17 and 18 have two outgoing interfaces. If labeled packets are load-balanced, they can have the same outgoing labels, but they can also be different. The outgoing labels are the same if the two links are between a pair of routers and both links belong to the platform label space. If multiple next-hop LSRs exist, the outgoing label for each path is usually different, because the next-hop LSRs assign labels independently.

Example 3-6 *Example of Load Balancing Labeled Packets*

```
horizon#show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
17     Pop tag     10.200.254.3/32   252        Et1/3      10.200.203.2
       Pop tag     10.200.254.3/32   0          Et1/2      10.200.201.2
18     16          10.200.254.4/32   10431273   Et1/2      10.200.201.2
       16          10.200.254.4/32   238        Et1/3      10.200.203.2
```

If a prefix is reachable via a mix of labeled and unlabeled (IP) paths, Cisco IOS does not consider the unlabeled paths for load-balancing labeled packets. That is because in some cases, the traffic going over the unlabeled path does not reach its destination. In the case of plain IPv4-over-MPLS (MPLS running on an IPv4 network), the packets reach the destination even if they become unlabeled. The packets become unlabeled at the link where MPLS is not enabled, and become labeled again at the next link where MPLS is enabled. At the place where the packets become unlabeled, an IP lookup has to occur. Because the network is running IPv4 everywhere, it should be able to deliver the packet to its destination without a label. However, in some scenarios, as with MPLS VPN or Any Transport over MPLS (AToM), a packet that becomes unlabeled in the MPLS network at a certain link does not make it to its final destination.

In the example of MPLS VPN, the MPLS payload is an IPv4 packet, but the P routers do not normally have the VPN routing tables, so they cannot route the packet to its destination. In the case of AToM, the MPLS payload is a Layer 2 frame; therefore, if the packet loses its label stack on a P router, the P router does not have the Layer 2 forwarding tables present to forward the frame further. This is why in an MPLS network labeled packets are not load-balanced over an IP and a labeled path. In general, the intelligence to forward the MPLS payload is on the edge LSRs (or PEs) only. Therefore, a P router cannot—in most cases—forward a packet that becomes unlabeled.

Example 3-7 shows load balancing via two labeled paths. Then Label Distribution Protocol (LDP) is disabled over one of the two outgoing links, and that link is removed as a next hop in the LFIB. The command **no mpls ip** on an interface disables LDP on that interface.

Example 3-7 *Changing One Path to Unlabeled*

```
horizon#show mpls forwarding-table 10.200.254.4
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
18     18          10.200.254.4/32   56818      Et1/2      10.200.201.2
       18          10.200.254.4/32   160        Et1/3      10.200.203.2
horizon#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
horizon(config)#interface ethernet 1/3
horizon(config-if)#no mpls ip
horizon(config-if)#^Z
horizon#horizon#show mpls forwarding-table 10.200.254.4
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
18     18          10.200.254.4/32   57270      Et1/2      10.200.201.2
```

Unknown Label

In normal operation, an LSR should receive only a labeled packet with a label at the top of the stack that is known to the LSR, because the LSR should have previously advertised that label. However, it is possible for something to go wrong in the MPLS network and the LSR to start receiving labeled packets with a top label that the LSR does not find in its LFIB. The LSR can theoretically try two things: strip off the labels and try to forward the packet, or drop the packet. The Cisco LSR drops the packet. This is the right thing to do, because this LSR did not assign the top label, and it does not know what kind of packet is behind the label stack. Is it an IPv4, IPv6 packet, a Layer 2 frame, or something else? The LSR can try to figure that out by performing an inspection of the MPLS payload. But then the same problem as described in the previous section occurs: The LSR on which the packet or frame becomes unlabeled is likely not able to look up the destination of the packet or frame. Even if the LSR tries to forward the packet, it is not guaranteed that the packet will not get dropped at a router downstream. The only right thing to do is to drop an incoming packet with an unknown top label.

Reserved Labels

Labels 0 through 15 are reserved labels. An LSR cannot use them in the normal case for forwarding packets. An LSR assigns a specific function to each of these labels. Label 0 is the explicit NULL label, whereas label 3 is the implicit NULL label. Label 1 is the router alert label, whereas label 14 is the OAM alert label. The other reserved labels between 0 and 15 have not been assigned yet.

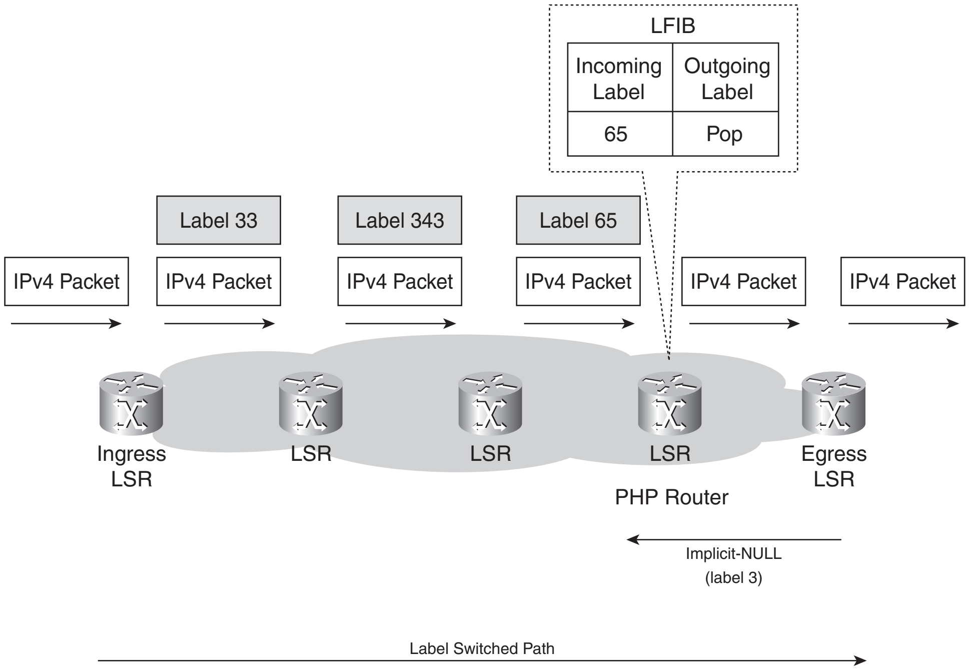
Implicit NULL Label

The implicit NULL label is the label that has a value of 3. An egress LSR assigns the implicit NULL label to a FEC if it does not want to assign a label to that FEC, thus requesting the upstream LSR to perform a pop operation. In the case of a plain IPv4-over-MPLS network, such as an IPv4 network in which LDP distributes labels between the LSRs, the egress LSR—running Cisco IOS—assigns the implicit NULL label to its connected and summarized prefixes. The benefit of this is that if the egress LSR were to assign a label for these FECs, it would receive the packets with one label on top of it. It would then have to do two lookups. First, it would have to look up the label in the LFIB, just to figure out that the label needs to be removed; then it would have to perform an IP lookup. These are two lookups, and the first is unnecessary.

The solution for this double lookup is to have the egress LSR signal the last but one (or penultimate) LSR in the label switched path (LSP) to send the packets without a label. The egress LSR signals the penultimate LSR to use implicit NULL by not sending a regular label, but by sending the special label with value 3. The result is that the egress LSR receives an IP packet and only needs to perform an IP lookup to be able to forward the packet. This enhances the performance on the egress LSR.

The use of implicit NULL at the end of an LSP is called *penultimate hop popping* (PHP). The LFIB entry for the LSP on the PHP router shows a "Pop Label" as the outgoing label. Figure 3-3 shows penultimate hop popping.

Figure 3-3 *Penultimate Hop Popping*



NOTE PHP is the default mode in Cisco IOS. In the case of IPv4-over-MPLS, Cisco IOS only advertises the implicit NULL label for directly connected routes and summarized routes.

The use of implicit NULL is widespread and not confined only to the example in Figure 3-3. It could be that the packets have two or three or more labels in the label stack. Then the implicit NULL label used at the egress LSR would signal the penultimate hop router to pop one label and send the labeled packet with one label less to the egress LSR. Then the egress LSR does not have to perform two label lookups. The use of the implicit NULL label does not mean that all labels of the label stack must be removed. Only one label is popped off. In any case, the use of the implicit NULL label prevents the egress LSR from having to perform two lookups. Although the label value 3 signals the use of the implicit NULL label, the label 3 will never be seen as a label in the label stack of an MPLS packet. That is why it is called the *implicit* NULL label.

Explicit NULL Label

The use of implicit NULL adds efficiency when forwarding packets. However, it has one downside: The packet is forwarded with one label less than it was received by the penultimate LSR or unlabeled if it was received with only one label. Besides the label value, the label also holds the Experimental (EXP) bits. When a label is removed, the EXP bits are also removed. Because the EXP bits are exclusively used for quality of service (QoS), the QoS part of the packet is lost when the top label is removed. In some cases, you might want to keep this QoS information and have it delivered to the egress LSR. Implicit NULL cannot be used in that case.

The explicit NULL label is the solution to this problem, because the egress LSR signals the IPv4 explicit NULL label (value 0) to the penultimate hop router. The egress LSR then receives labeled packets with a label of value 0 as the top label. The LSR cannot forward the packet by looking up the value 0 in the LFIB because it can be assigned to multiple FECs. The LSR just removes the explicit NULL label. After the LSR removes the explicit NULL label, another lookup has to occur, but the advantage is that the router can derive the QoS information of the received packet by looking at the EXP bits of the explicit NULL label.

NOTE The explicit NULL label for IPv6 has the value 2.

You can copy the EXP bits value to the precedence or DiffServ bits when performing PHP and thus preserve the QoS information. Or, if the label stack has multiple labels and the top label is popped off, you can copy the EXP bits value to the EXP field of the new top label. However, Chapter 12, “MPLS and Quality of Service,” gives you two examples where this is not wanted; thus, the use of the explicit NULL label is warranted.

Router Alert Label

The Router Alert label is the one with value 1. This label can be present anywhere in the label stack except at the bottom. When the Router Alert label is the top label, it alerts the LSR that the packet needs a closer look. Therefore, the packet is not forwarded in hardware, but it is looked at by a software process. When the packet is forwarded, the label 1 is removed. Then a lookup of the next label in the label stack is performed in the LFIB to decide where the packet needs to be switched to. Next, a label action (pop, swap, push) is performed, the label 1 is pushed back on top of the label stack, and the packet is forwarded. Refer to Chapter 14, “MPLS Operation and Maintenance,” for more details on the Router Alert label.

Example 3-8 shows the output of **debug mpls packet** on a router for a labeled packet with the Router Alert label on it.

Example 3-8 *Debug MPLS Packet Showing Label 1*

```
00:39:14: MPLS: Et1/1: recvd: CoS=6, TTL=255, Label(s)=1/21
00:39:14: MPLS: Et1/3: xmit: CoS=6, TTL=254, Label(s)=1/18

00:38:13: MPLS turbo: Se4/0: rx: Len 76 Stack {1 6 255} {20 6 255} - ipv4 data
00:38:13: MPLS les: Se4/0: rx: Len 76 Stack {1 6 255} {20 6 255} - ipv4 data
```

Example 3-8 shows two possible formats in the output. Both formats have the labels sorted from left to right or topmost label to bottommost label. The first format is the old format, with the slash separating the labels. The second format is the new format with the *{label EXP TTL}* format.

OAM Alert Label

The label with value 14 is the Operation and Maintenance (OAM) Alert label as described by the ITU-T Recommendation Y.1711 and RFC 3429. OAM is basically used for failure detection, localization, and performance monitoring. This label differentiates OAM packets from normal user data packets. Cisco IOS does not use label 14. It does perform MPLS OAM, but not by using label 14. Chapter 14 covers MPLS OAM in greater detail.

Unreserved Labels

Except for the reserved labels of 0 through 15, you can use all the label values for normal packet forwarding. Because the label value has 20 bits, the labels from 16 through 1,048,575 (2^{20} – 1) are used for normal packet forwarding. In Cisco IOS, the default range is 16 through 100,000. This is more than enough for labeling all the IGP prefixes you have, but if you want to label the BGP prefixes, this number might be insufficient. You can change the label range with the **mpls label range** *min max* command. Example 3-9 shows how to change the default mpls label range.

Example 3-9 *Changing the MPLS Label Range*

```
event#show mpls label range
Downstream Generic label region: Min/Max label: 16/100000

event#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
event(config)#mpls label range ?
<16-1048575>  Minimum label value
event(config)#mpls label range 16 ?
  <16-1048575>  Maximum label value
event(config)#mpls label range 16 1048575

event#show mpls label range
Downstream Generic label region: Min/Max label: 16/1048575
```

TTL Behavior of Labeled Packets

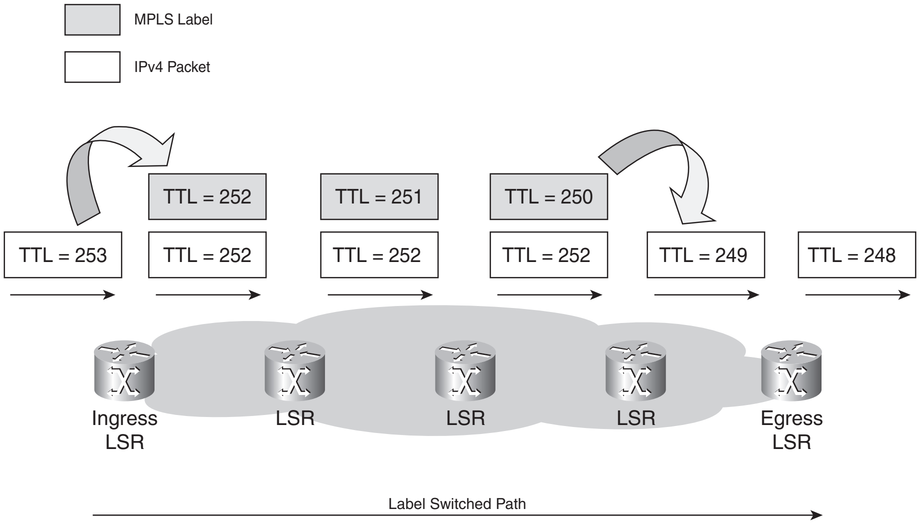
Time To Live (TTL) is a well-known mechanism thanks to IP. In the IP header is a field of 8 bits that signifies the time that a packet still has before its life ends and is dropped. When an IP packet is sent, its TTL is usually 255 and is then decremented by 1 at each hop. If the TTL reaches 0, the packet is dropped. In such a case, the router that dropped the IP packet for which the TTL reached 0 sends an Internet Control Message Protocol (ICMP) message type 11 and code 0 (time exceeded) to the originator of the IP packet.

With the introduction of MPLS, labels are added to IP packets. This calls for a mechanism in which the TTL is propagated from the IP header into the label stack and vice versa. This ensures that packets do not live forever when entering and leaving the MPLS cloud, if there is a routing loop.

TTL Behavior in the Case of IP-to-Label or Label-to-IP

In MPLS, the usage of the TTL field in the label is the same as the TTL in the IP header. When an IP packet enters the MPLS cloud—such as on the ingress LSR—the IP TTL value is copied (after being decremented by 1) to the MPLS TTL values of the pushed label(s). At the egress LSR, the label is removed, and the IP header is exposed again. The IP TTL value is copied from the MPLS TTL value in the received top label after decrementing it by 1. In Cisco IOS, however, a safeguard guards against possible routing loops by not copying the MPLS TTL to the IP TTL if the MPLS TTL is greater than the IP TTL of the received labeled packet. If the MPLS TTL would be copied to the IP header, the smaller IP TLL value would be overwritten by a newer but higher value. If the IP packet would be injected into the MPLS cloud again—such as the result of a routing loop—the packet could live forever because the TTL would never reach 0. Figure 3-4 shows the default behavior of copying or propagating the TTL between the IP header and the MPLS labels and vice versa.

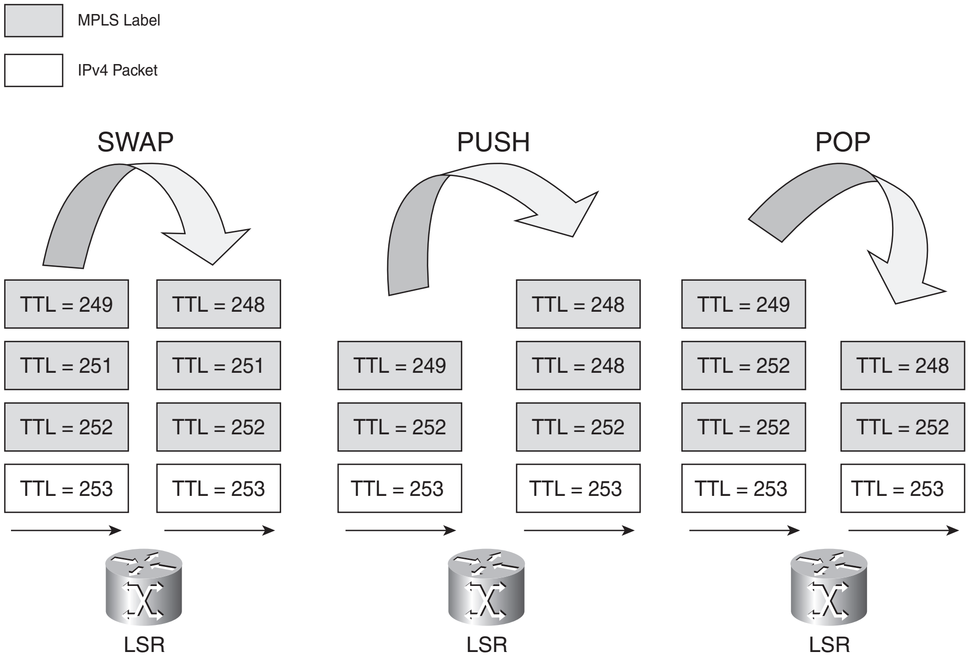
Figure 3-4 *Propagation Behavior of TTL Between IP Header and MPLS Labels*



TTL Behavior in the Case of Label-to-Label

If the operation that is performed on the labeled packet is a swap, the TTL of incoming label –1 is copied to the swapped label. If the operation that is performed on the labeled packet is to push one or more labels, the received MPLS TTL of the top label –1 is copied to the swapped label and all pushed labels. If the operation is pop, the TTL of the incoming label –1 is copied to the newly exposed label unless that value is greater than the TTL of the newly exposed label, in which case the copy does not happen. Figure 3-5 shows examples of TTL propagation in the case of Label-to-Label operation for a swap, push, and pop operation.

Figure 3-5 *TTL Propagation in Label-to-Label Operation in the Case of a Swap, Push, and Pop Operation*



The intermediate LSR does not change the TTL field in underlying labels or the TTL field in the IP header. An LSR only looks at or only changes the top label in the label stack of a packet.

NOTE The TTL behavior of labeled packets described here refers to the TTL operation in Cisco IOS.

TTL Expiration

When a labeled packet is received with a TTL of 1, the receiving LSR drops the packet and sends an ICMP message "time exceeded" (type 11, code 0) to the originator of the IP packet. This is the same behavior that a router would exhibit with an IP packet that had an expiring TTL. However, the ICMP message is not immediately sent back to the originator of the packet because an interim LSR might not have an IP path toward the source of the packet. The ICMP message is forwarded along the LSP the original packet was following.

Figure 3-6 shows a router sending the ICMP message "time exceeded" to the originator of the packet in the case of an IP network.

Figure 3-6 *ICMP "Time Exceeded" Sent Back by a Router in an IP Network*

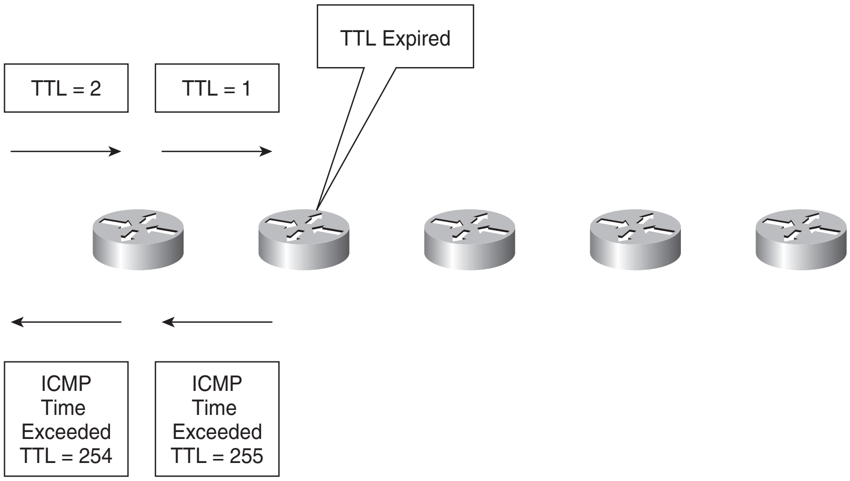
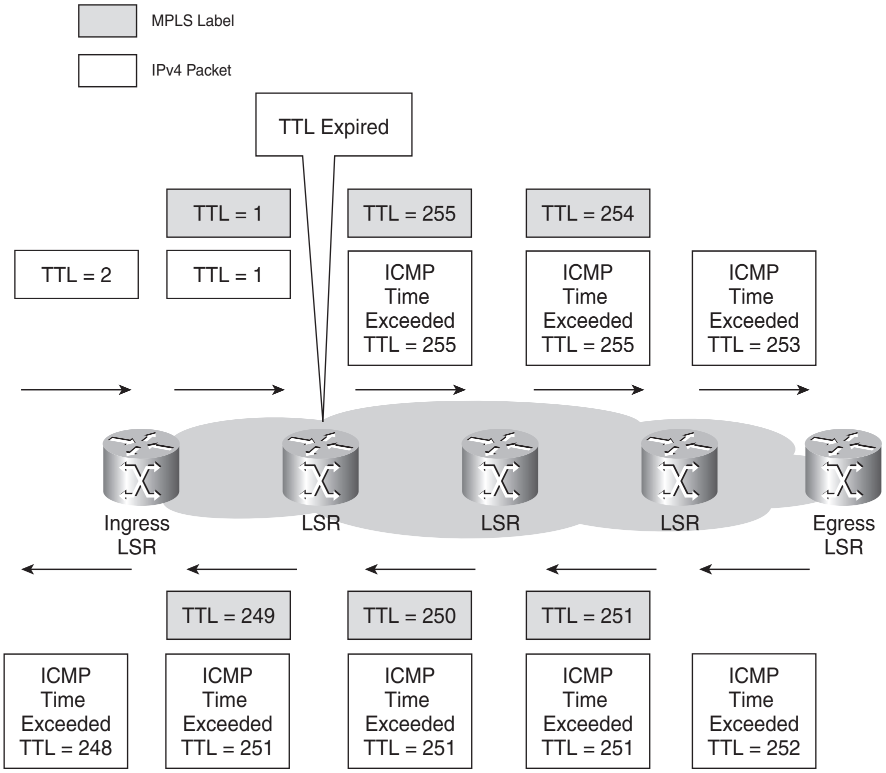


Figure 3-7 shows an LSR forwarding the ICMP "time exceeded" message along the LSP of the original packet.

Figure 3-7 *ICMP "Time Exceeded" Sent by a Router in an MPLS Network*


MPLS Label
IPv4 Packet
TTL Expired
TTL = 1
TTL = 255
TTL = 254
TTL = 2
TTL = 1
ICMP Time Exceeded TTL = 255
ICMP Time Exceeded TTL = 255
ICMP Time Exceeded TTL = 253
Ingress LSR
LSR
LSR
LSR
Egress LSR
TTL = 249
TTL = 250
TTL = 251
ICMP Time Exceeded TTL = 248
ICMP Time Exceeded TTL = 251
ICMP Time Exceeded TTL = 251
ICMP Time Exceeded TTL = 251
ICMP Time Exceeded TTL = 252

The reason for this forwarding of the ICMP message along the LSP that the original packet with the expiring TTL was following is that in some cases the LSR that is generating the ICMP message has no knowledge of how to reach the originator of the original packet. Equally so, an intermediate LSR closer to the originator of the packet might not have that knowledge. One such case is a network with MPLS VPN. In this scenario, the P router does not have the knowledge to send back the ICMP messages to the originator of the VPN packet, because the P router does not have a route to directly return the ICMP message. (In general, the P routers do not hold the VPN routing tables.) Hence, the P router builds the ICMP message and forwards the packet along the LSP, in the hope that the ICMP message reaches a router at the end of the LSP that can return the packet to the originating routing. In the case of MPLS VPN, the ICMP message is returned by the egress PE or the CE that is attached to that PE, because these routers certainly have the route to correctly return the packet.

It is important that the P router—where the TTL expires—notes what the MPLS payload is. The P router checks whether the payload is an IPv4 (or IPv6) packet. If it is, it can generate the ICMP "time exceeded" message and forward it along the LSP. However, if the payload is not an IPv4 (or IPv6) packet, the P router cannot generate the ICMP message. Therefore, the P router drops the packet in all cases, except if it is an IPv4 (or IPv6) packet. A case in which the LSR drops a packet with the TTL expiring is AToM. The MPLS payload in the case of AToM is a Layer 2 frame and not an IP packet. Hence, if the TTL in the top label of an AToM packet expires at a P router, the only action that the P router can undertake is to drop the packet, because an IP lookup is not possible. The packet is also dropped if the payload is an IPv6 packet. However, if the P router runs newer Cisco IOS code—which understands the IPv6 protocol—that router can generate the ICMP IPv6 time exceeded packet. Whether the P router actually has an IPv6 route pointing to the originator of the packet is irrelevant. This is so because the ICMP message is always forwarded along the LSP of the packet with the expiring TTL.

NOTE Chapter 13, "Troubleshooting MPLS Networks," has a more detailed description of what happens when the TTL expires. Tracerouting—which relies on expiring TTL to function properly—in an MPLS network is explained there, too.

MPLS MTU

Maximum transmission unit (MTU) is a well-known parameter in the IP world. It indicates the maximum size of the IP packet that can still be sent on a data link, without fragmenting the packet. Data links in MPLS networks also have a specific MTU, but for labeled packets. Take the case of an IPv4 network implementing MPLS. All IPv4 packets have one or more labels. This does imply that the labeled packets are slightly bigger than the IP packets, because for every label, four bytes are added to the packet. So, if *n* is the number of labels, *n* * 4 bytes are added to the size of the packet when the packet is labeled.

This section explains that an MPLS MTU parameter pertains to labeled packets. Furthermore, it explains what giant and baby giant frames are and how to ensure that Ethernet switches can handle them. Finally, a new parameter is introduced: MPLS Maximum Receive Unit. This parameter is used in the LFIB to keep track of how big labeled packets can be and still be forwarded without needing to fragment them.

MPLS MTU Command

The interface MTU command in Cisco IOS specifies how big a Layer 3 packet can be without having to fragment it when sending it on a data link. For the Ethernet encapsulation, for example, MTU is by default set to 1500. However, when *n* labels are added, *n* * 4 bytes are added to an already maximum sized IP packet of 1500 bytes. This would lead to the need to fragment the packet.

Cisco IOS has the **mpls mtu** command that lets you specify how big a labeled packet can be on a data link. If, for example, you know that all packets that are sent on the link have a maximum of two labels and the MTU is 1500 bytes, you can set the MPLS MTU to 1508 (1500 + 2 * 4). Thus, all labeled packets of size 1508 bytes (labels included) can be sent on the link without fragmenting them. The default MPLS MTU value of a link equals the MTU value. Look at Example 3-10 to see how you can change the MPLS MTU on an interface in Cisco IOS.

Example 3-10 *Changing MPLS MTU*

```
london#show mpls interfaces fastEthernet 2/6 detail
Interface FastEthernet2/6:
        IP labeling enabled
        LSP Tunnel labeling not enabled
        BGP labeling not enabled
        MPLS not operational
        MTU = 1500
london#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
london(config)#interface FastEthernet2/6
london(config-if)#mpls mtu 1508
london(config-if)#^Z
london#
london#show mpls interfaces fastEthernet 2/6 detail
Interface FastEthernet2/6:
        IP labeling enabled
        LSP Tunnel labeling not enabled
        BGP labeling not enabled
        MPLS not operational
        MTU = 1508
```

Giant and Baby Giant Frames

When a packet becomes labeled, the size increases slightly. If the IP packet was already at the maximum size possible for a certain data link (full MTU), it becomes too big to be sent on that data link because of the added labels. Therefore, the frame at Layer 2 becomes a giant frame. Because the frame is only slightly bigger than the maximum allowed, it is called a baby giant frame.

Take the example of Ethernet: The payload can be a maximum of 1500 bytes. However, if the packet is a maximum sized packet and labels are added, the packet becomes slightly too big to be sent on the Ethernet link. It is possible to close one eye and allow frames that are bigger (perhaps by just a few bytes) to be sent on the Ethernet link, even though it is not the correct thing according to the Ethernet specifications, which say that such frames should be dropped. This is, of course, possible only if the Ethernet hardware in the router and all switches in the Ethernet network support receiving and sending baby giant frames.

On Ethernet data links on LSRs, you can set the MPLS MTU to 1508 bytes to allow IP packets with a size of 1500 bytes with two labels to be received and forwarded. If, however, the hardware of the router does not support this, or if an Ethernet switch exists in between, dropping baby giant frames, you can lower the MPLS MTU parameter on the LSRs. When you set the MPLS MTU to 1500, all the IP packets with a size of 1492 bytes are still forwarded, because the size of the labeled packet then becomes 1500 (1492 plus 8) bytes at Layer 3. However, all IP packets sized between 1493 through 1500 bytes (or more) are fragmented. Because of the performance impact of fragmentation, you should use methods to avoid it, such as path MTU discovery.

> **NOTE** In some Cisco IOS releases, you cannot configure the MPLS MTU to be bigger than the interface MTU.

Giant Frames on Switches

You can also see giant and baby giant frames on Layer 2 switches because the maximum Ethernet frame has increased by as many bytes as are in the label stack. Configuration might be needed on the Ethernet switches to allow them to switch giant and baby giant frames. Example 3-11 shows examples on how to enable jumbo Ethernet frames on an Ethernet switch.

Example 3-11 *Allowing Jumbo Frames on Ethernet Switches*

```
Cluster#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Cluster(config)#system jumbomtu ?
  <1500-9216>  Jumbo mtu size in Bytes, default is 9216

donquijote-msfc#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
donquijote-msfc(config)#int vlan 1
donquijote-msfc(config-if)#mtu ?
  <64-9216>  MTU size in bytes

Lander#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Lander(config)#system mtu ?
  <1500-2000>  MTU size in bytes
```

MPLS Maximum Receive Unit

Maximum receive unit (MRU) is a parameter that Cisco IOS uses. It informs the LSR how big a received labeled packet of a certain FEC can be that can still be forwarded out of this LSR without fragmenting it. This value is actually a value per FEC (or prefix) and not just per interface. The reason for this is that labels can be added to or removed from a packet on an LSR.

Think of the example of a router in which all the interfaces have an MTU of 1500 bytes. This means that the biggest IP packet that can be received and transmitted on all interfaces is 1500 bytes. Imagine that the packets can be labeled by adding a maximum of two labels. (Typically, MPLS VPN and AToM networks label the packets respectively the frames with two labels.) Also assume that the MPLS MTU is set to 1508 on all links to accommodate for the extra 8 bytes (2 times 4 bytes) for the labels. A labeled packet that is transmitted on any of the links can now be 1508 bytes. If, however, the operation on the incoming packet were POP, the packet could have been 4 bytes or 1 label bigger (thus 1512 bytes) when it was received, because one label would have been popped off before transmitting the packet. If the label operation were a push, however, and one label was added, the incoming packet could only have been 1504 bytes, because 4 bytes or one label would have been added—making the packet 1508 bytes—before switching the packet out.

As you can see, the label operation plays a role in determining the MRU. Because the label operation is determined per FEC or prefix, the MRU can change per FEC or prefix. Notice how in Example 3-12, the MRU changes per prefix according to the specific label operation performed on the packets. The LFIB shows you the value of the MRU per prefix.

Example 3-12 *Example of MRU*

```
lactometer#show mpls forwarding-table 10.200.254.2 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
21     Pop tag     10.200.254.2/32   0          Et0/0/0    10.200.200.2
        MAC/Encaps=14/14, MRU=1512, Tag Stack{}
        00604700881D00024A4008008847
        No output feature configured

lactometer#show mpls forwarding-table 10.200.254.3 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     17          10.200.254.3/32   0          Et0/0/0    10.200.200.2
        MAC/Encaps=14/18, MRU=1508, Tag Stack{17}
        00604700881D00024A4008008847 00011000
        No output feature configured

lactometer#show mpls forwarding-table 10.200.254.4 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     18          10.200.254.4/32   0          Tu1        point2point
        MAC/Encaps=14/22, MRU=1504, Tag Stack{20 18}, via Et0/0/0
        00604700881D00024A4008008847 0001400000012000
        No output feature configured
```

The MRU for the prefix 10.200.254.2/32 is 1512. The packet received can be 1512 bytes, because one label is popped off before it is forwarded. The MRU for prefix 10.200.254.3/32 is 1508. The size of the packet does not change, because only the top label is swapped. The MRU for prefix 10.200.254.4/32 is 1504. The packet received can be only 1504 bytes because one extra label is pushed onto the label stack before the packet is forwarded; therefore, the packet size increases by 4 bytes. The "Tag Stack" shows that one label is pushed onto the label stack after the incoming label is swapped.

Fragmentation of MPLS Packets

If an LSR receives a labeled packet that is too big to be sent out on a data link, the packet should be fragmented. This is similar to fragmenting an IP packet. If a labeled packet is received and the LSR notices that the outgoing MTU is not big enough for this packet, the LSR strips off the label stack, fragments the IP packet, puts the label stack (after the pop, swap, or push operation) onto all fragments, and forwards the fragments. Only if the IP header has the Don't Fragment (DF) bit set does the LSR not fragment the IP packet, but it drops the packet and returns an ICMP error message "Fragmentation needed and do not fragment bit set" (ICMP type 3, code 4) to the originator of the IP packet. As with the ICMP message "time exceeded" (type 11, code 0), which is sent when the TTL expires of a labeled packet, the "Fragmentation needed and do not fragment bit set" ICMP message is sent, using a label stack that is the outgoing label stack for the packet that caused the ICMP message to be created. This means that the ICMP message travels further down the LSP until it reaches the egress LSR of that LSP. Then it is returned to the originator of the packet with the DF bit set.

In general, fragmentation causes a performance impact and should be avoided. A good method to avoid fragmentation is using the Path MTU Discovery method as described in the next section.

Path MTU Discovery

One method to avoid fragmentation is Path MTU Discovery, which most modern IP hosts perform automatically. In that case, the IP packets sent out have the "Don't Fragment" (DF) bit set. When a packet encounters a router that cannot forward the packet without fragmenting it, the router notices that the DF bit is set, drops the packet, and sends an ICMP error message "Fragmentation needed and do not fragment bit set" (ICMP type 3, code 4) to the originator of the IP packet. The originator of the IP packet then lowers the size of the packet and retransmits the packet. If a problem still exists, the host can lower the size of the packet again. This continues until no ICMP message is received for the IP packet. The size of the last IP packet successfully sent is then used as maximum packet size for all subsequent IP traffic between the specific source and destination; hence, it is the MTU of the path.

NOTE Path MTU Discovery is not guaranteed to work in all cases; sometimes the ICMP message does not make it back to the originator. Possible causes for the ICMP message not making it to the originator of the packet are firewalls, access lists, and routing problems.

Summary

In this chapter, you have learned how a packet is forwarded in an MPLS network. You have seen that CEF can label an incoming IP packet. An incoming labeled packet is forwarded by looking up the top label value in the LFIB and finding the label operation and next hop to forward the packet to. You have learned that several specially reserved labels (implicit NULL label, explicit NULL label, and Router Alert label) exist and what their function is. MPLS has its own MTU, which is important because the labeled packet is slightly bigger than the unlabeled one. That is because only a few labels add to the packet. You have seen that MPLS has a TTL field in the labels that is used in the same way as it is used for IP packets, but there is a specific behavior in copying the IP TTL to the label TTL fields and vice versa. Finally, you have seen that MPLS also supports fragmenting labeled packets.

Chapter Review Questions

1. What does the push operation do on a labeled packet?
2. Which Cisco IOS command do you use to see what the swapped label is and which labels are pushed onto a received packet for a certain prefix?
3. What does the outgoing label entry of "Aggregate" in the LFIB of a Cisco IOS LSR mean?
4. What label value signals the penultimate LSR to use penultimate hop popping (PHP)?
5. What are the value and the function of the Router Alert label?
6. Why does an LSR forward the ICMP message "time exceeded" along the LSP of the original packet with the TTL expiring instead of returning it directly?
7. Is using Path MTU Discovery a guarantee that there will be no MTU problems in the MPLS network?
8. Why is MTU or MRU such an important parameter in MPLS networks?