MPLS Introduction

Why MPLS?

Agenda

- IP Routing /Forwarding Limitations
- Limitations of IP over ATM
- Limitations of IP based Traffic Engineering
- MPLS concepts
- MPLS vs. IP over ATM
- MPLS-TE vs. IP-TE
- MPLS Benefits

IP Routing/Forwarding Limitations

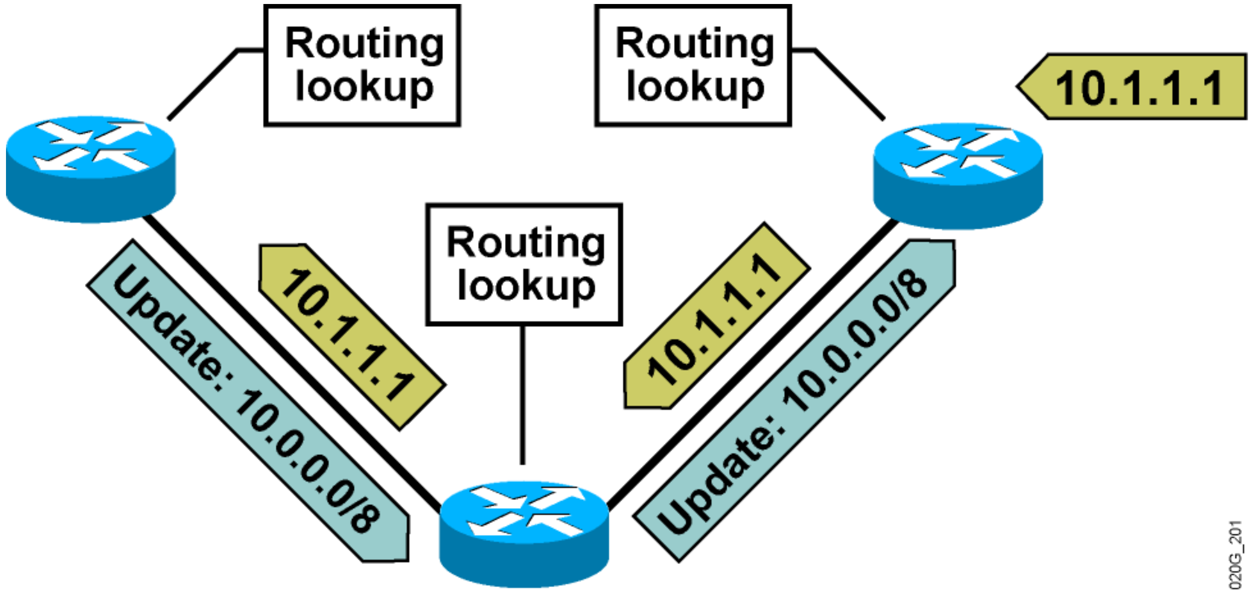
- IP Routing is based on:

    Routing protocols, which are used to distribute Layer 3 routing information.

    Forwarding is based on the Layer 3 destination address

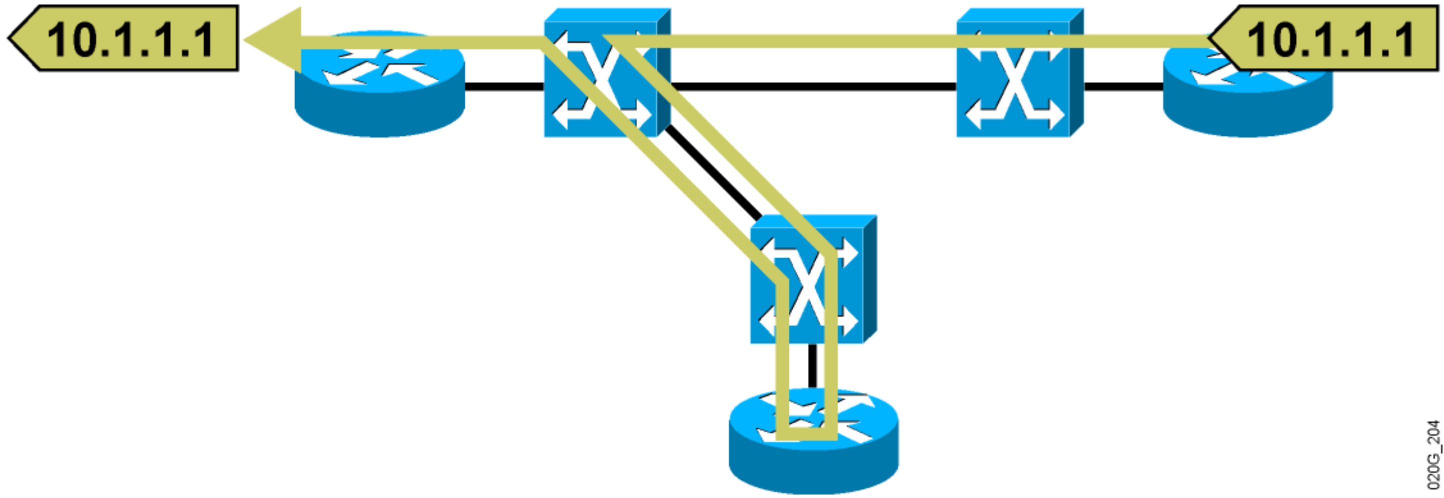
    Routing lookups are performed on every hop.

IP Limitations (contd.)



Destination-based routing lookup is needed on every hop.

IP Limitations:
IP over ATM

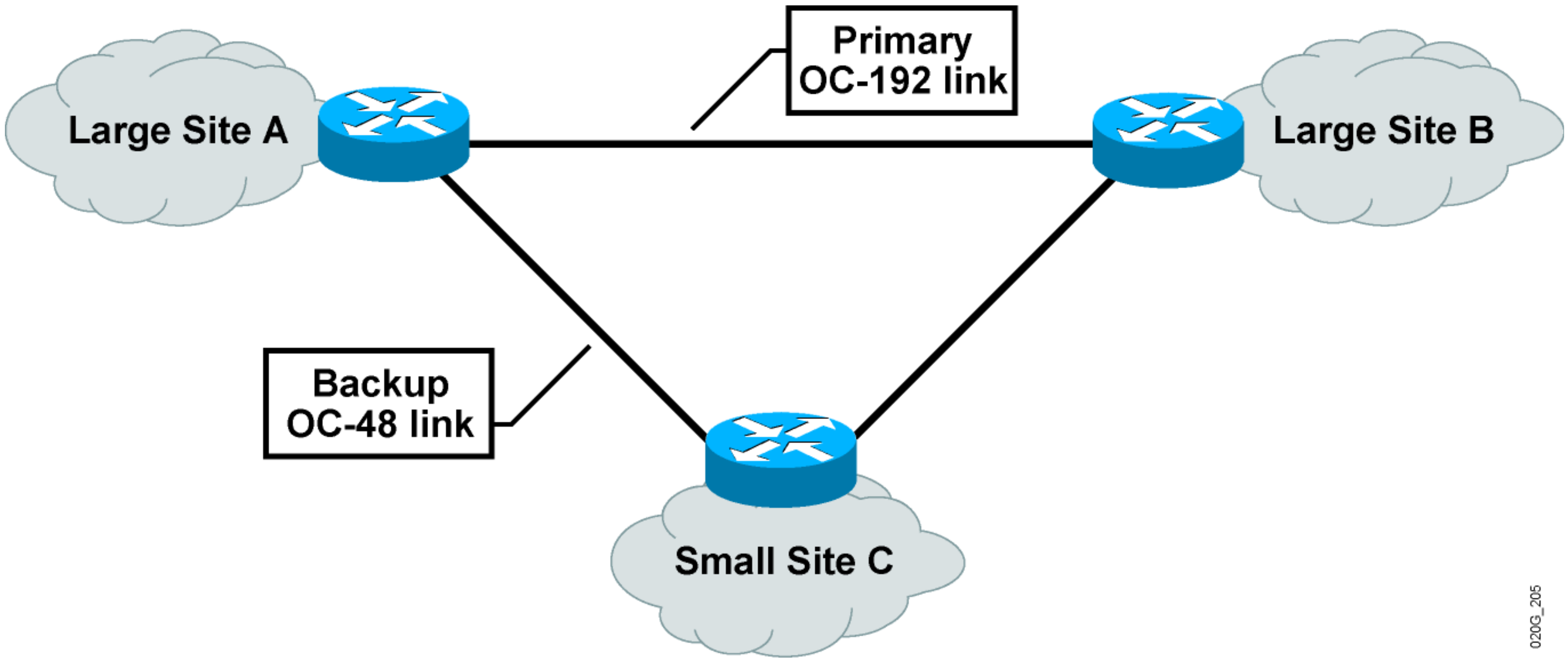


Layer 2 devices have no knowledge of Layer 3 routing information—virtual circuits must be manually established.

Layer 2 topology may be different from Layer 3 topology, resulting in suboptimal paths

Even if the two topologies overlap, the hub-and-spoke topology is usually used because of easier management.

IP Limitations: Traffic Engineering



Based on Routing Protocol forwarding, all traffic between Site A and Site B takes the OC-192 link, even if this link is congested. The alternate A-C-B link may be underutilized

Destination-based routing does not provide any mechanism for unequal cost balancing (except for variance in EIGRP)

Policy-based routing can be used to forward packets based on other parameters, but this is not a scalable solution.

How MPLS helps

MPLS (Multi Protocol Label Forwarding) uses Labels to forward packets.

Labels usually correspond to IP destination networks (equal to traditional IP forwarding).

Labels can also correspond to other parameters, such as ATM VC (in case of ATM over MPLS), VLAN id (in case of Ethernet over MPLS) or even QoS parameters

Now packet forwarding is no longer strictly tied to IP destination address

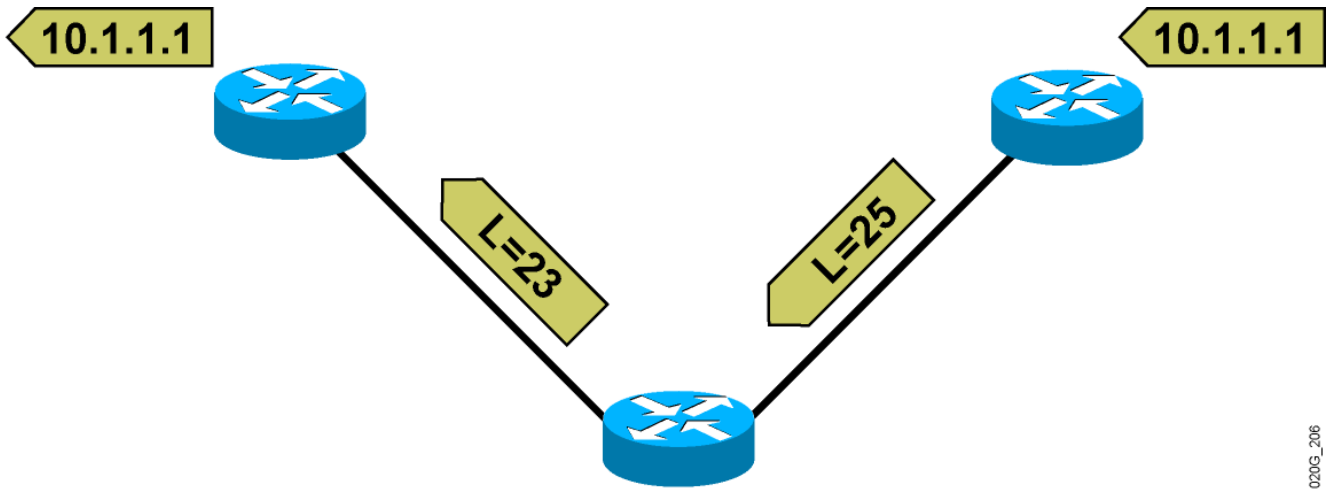
MPLS is an IETF standard based on RFC 3031, 3032

How MPLS helps (contd.)

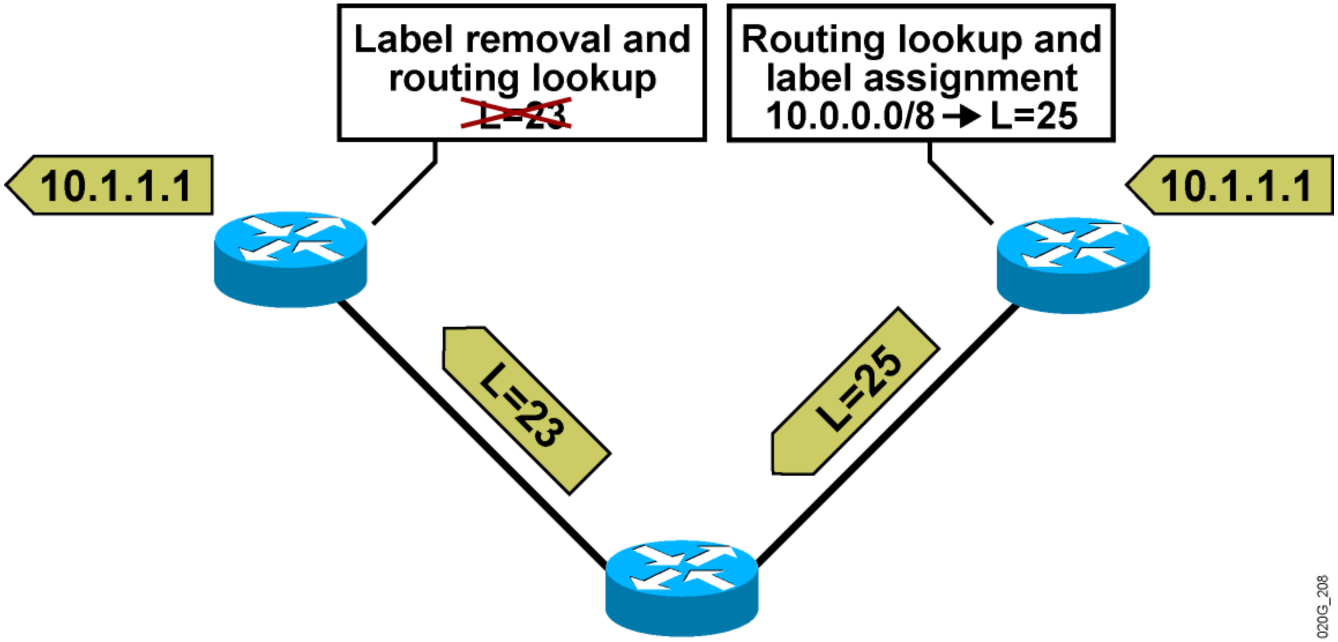
Labels assigned to packets can be based on:

- **Destination prefix**
- **Traffic Engineering tunnel**
- **VPN-ID, ATM VC, VLAN ID**
- **Class of Service**

Basic MPLS Concepts: Label based forwarding

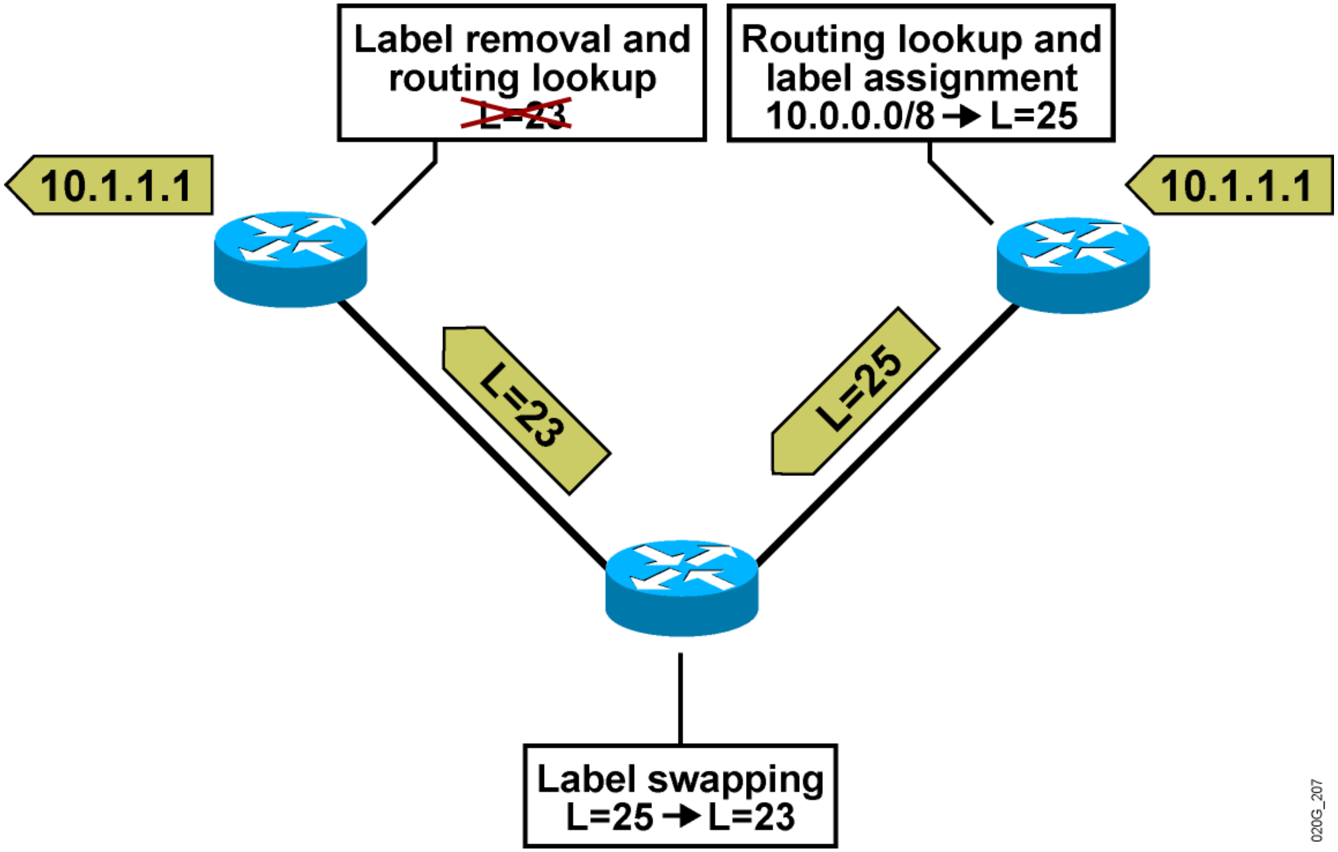


Basic MPLS Concepts:
Label based forwarding (contd.)



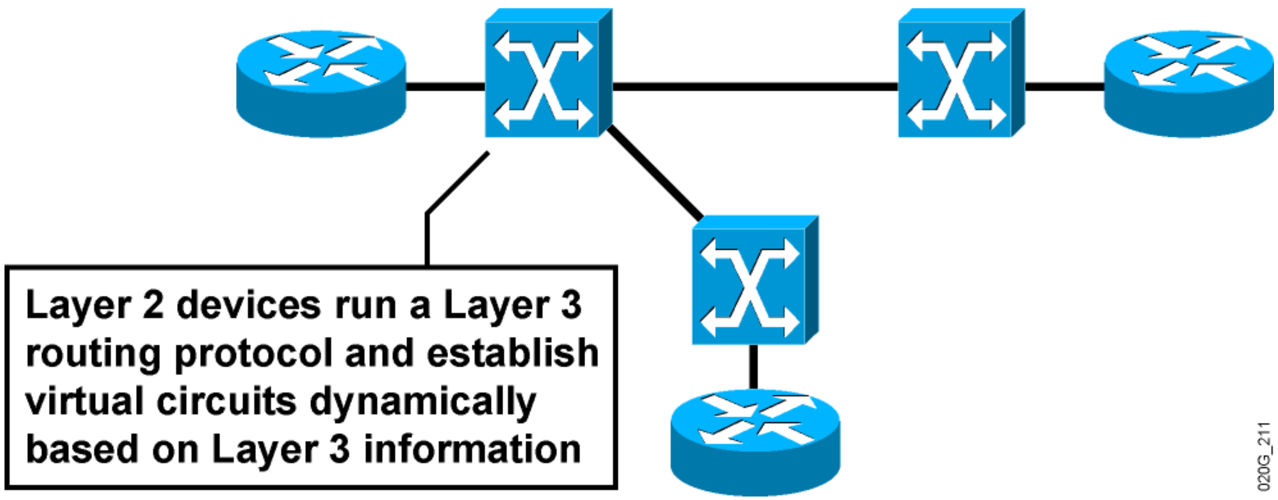
- **Only edge routers perform a routing (Layer 3) lookup.**

Basic MPLS Concepts: Label based forwarding (contd.)



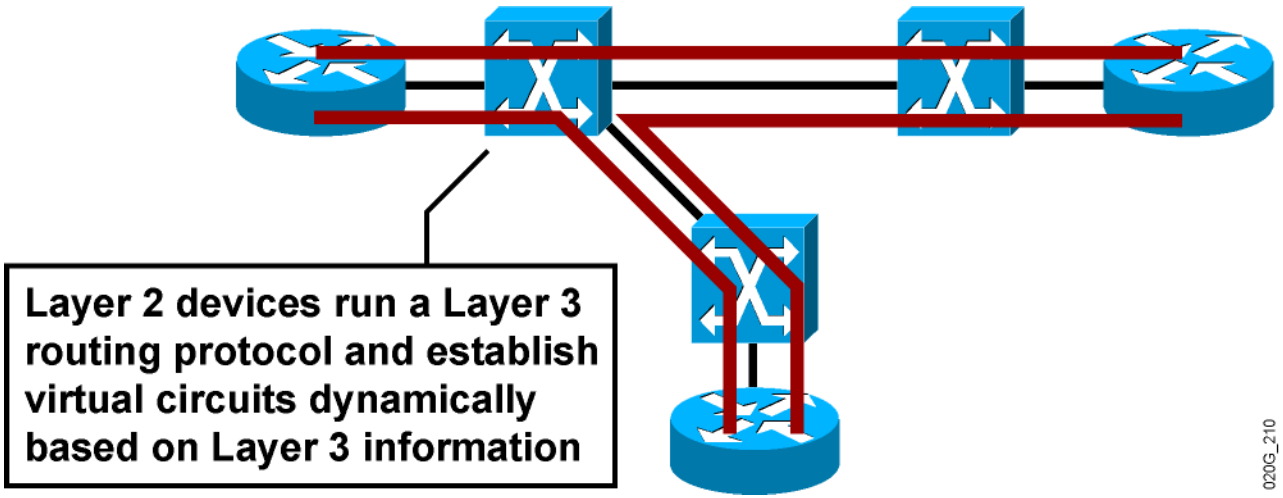
- Core routers forward packets based on MPLS label lookups
- Core router can be any device capable of doing label forwarding, so we might as well use a switch, if needed

MPLS versus IP over ATM



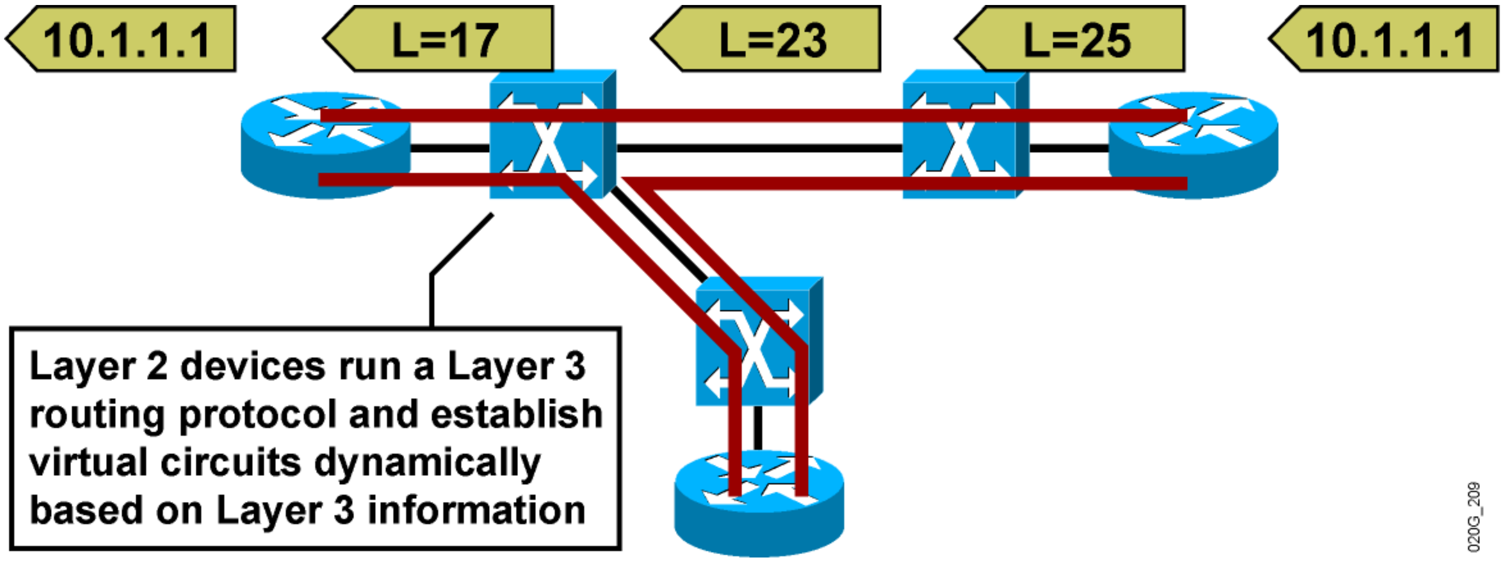
Layer 2 devices are IP-aware and run a routing protocol.

MPLS Versus IP over ATM (contd.)



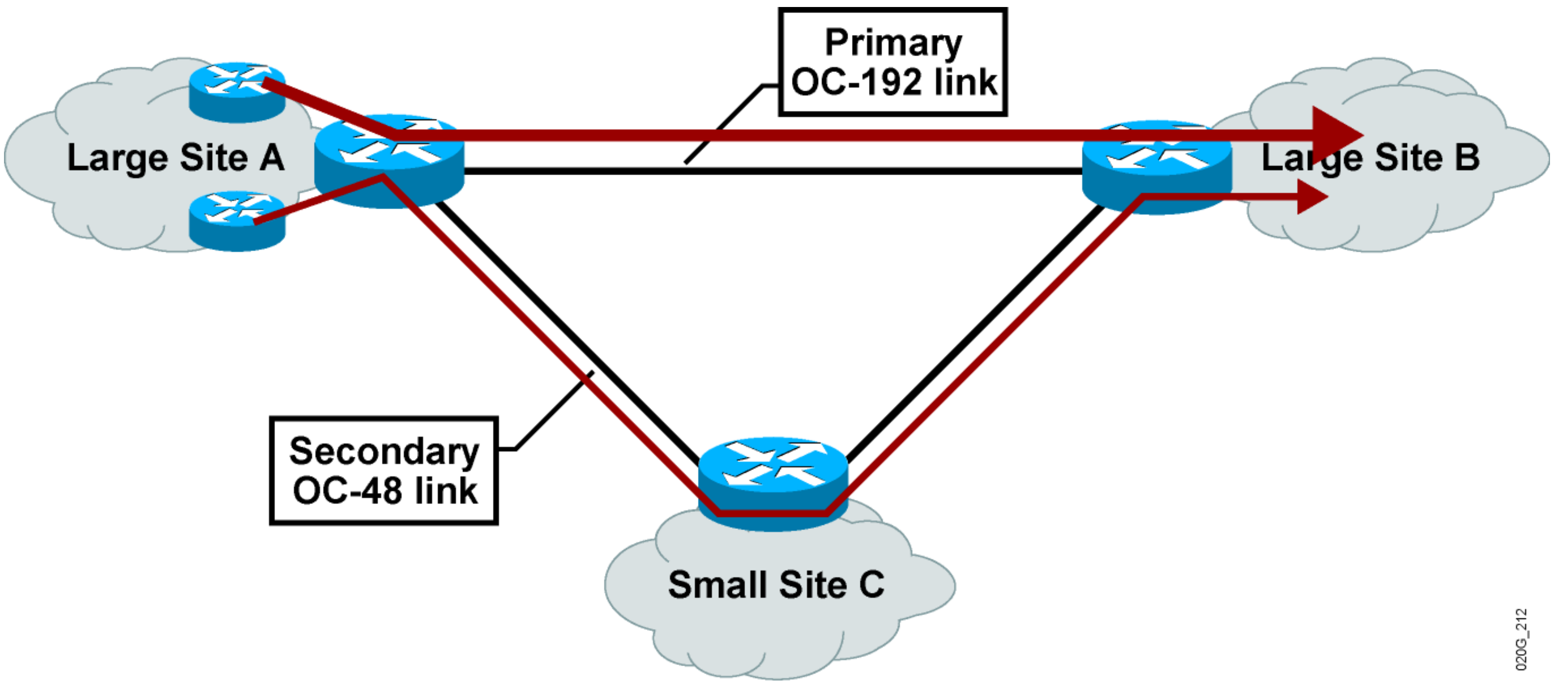
- **There is no need to manually establish virtual circuits.**

MPLS Versus IP over ATM



- Layer 2 devices are IP-aware and run a routing protocol.
- There is no need to manually establish virtual circuits.
- MPLS provides a virtual full-mesh topology.

Traffic Engineering with MPLS



Traffic can be forwarded based on labels via the primary and secondary links

Load sharing across unequal paths can be achieved.

We shall cover MPLS Traffic Engineering in detail in later modules

MPLS Benefits

- Separates Control plane & the forwarding plane
- Only ingress router needs to look up the network layer & make routing decision. Other LSRs only swap labels
- <u>Source Based routing</u>: e.g. explicit routes in MPLS-TE
- <u>Scalability</u>: Hierarchy of Routing (via label stacking)
- <u>AnyThing over MPLS (AToM):</u> Labels are common binding between different Layer 2 technologies like ATM, Ethernet.

Summary

Forwarding based on IP Routing only requires a layer 3 lookup at each router

MPLS forwards packets based on labels.

MPLS separates the control plane and forwarding plane cisco

MPLS Basics

Basic MPLS Concepts

Agenda

MPLS Architecture

MPLS Terminology

MPLS Labels

Label Switch Routers

LFIB and outgoing labels

MPLS and BGP

Label Distribution Protocols

Summary

MPLS Architecture

- MPLS has two major components:

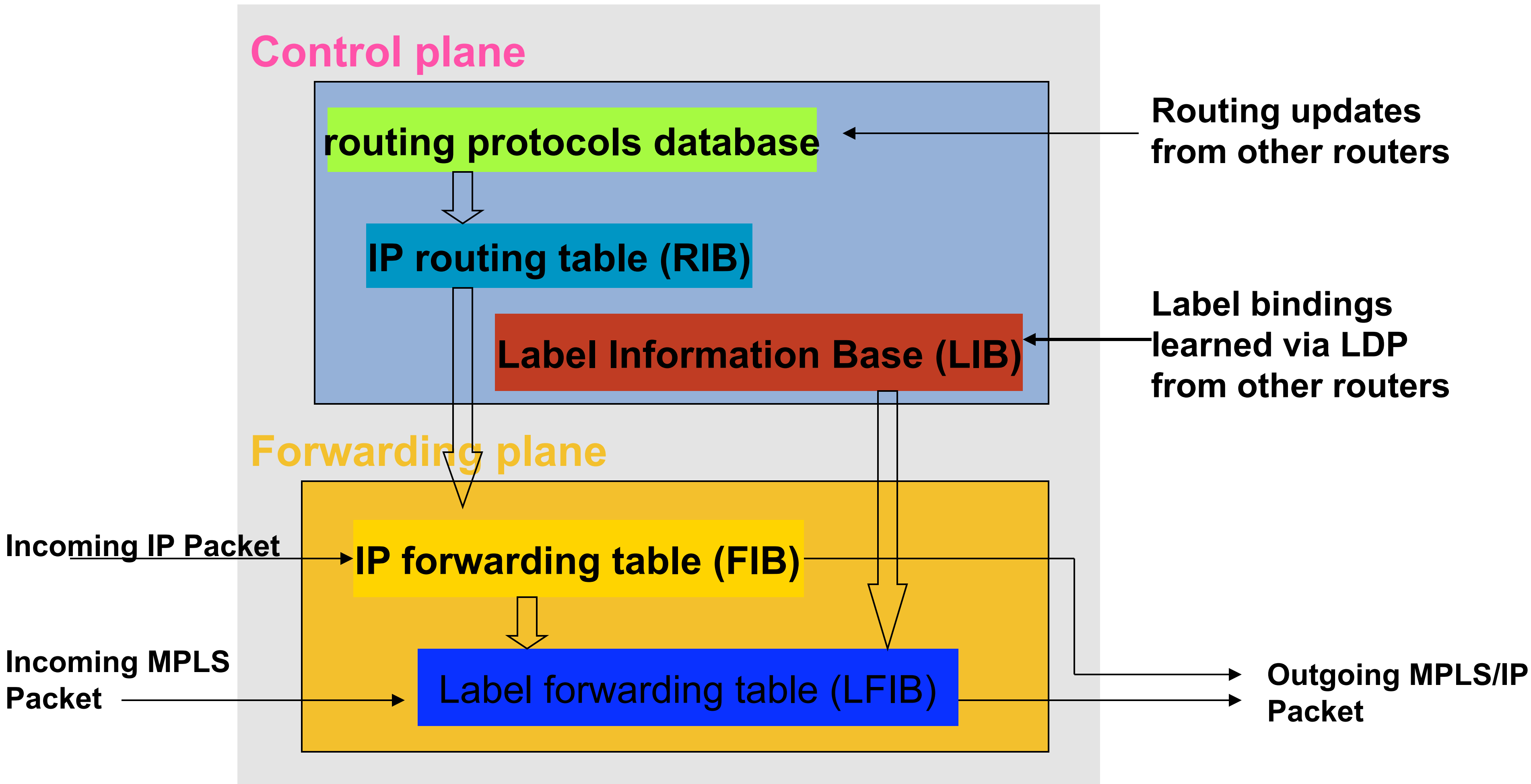
  Control plane: Exchanges Layer 3 routing information and labels

  Data plane: Forwards packets based on labels

  Control plane contains complex mechanisms to exchange routing information, such as OSPF, EIGRP, IS-IS, and BGP, and to exchange labels, such as TDP, LDP, BGP, and RSVP.

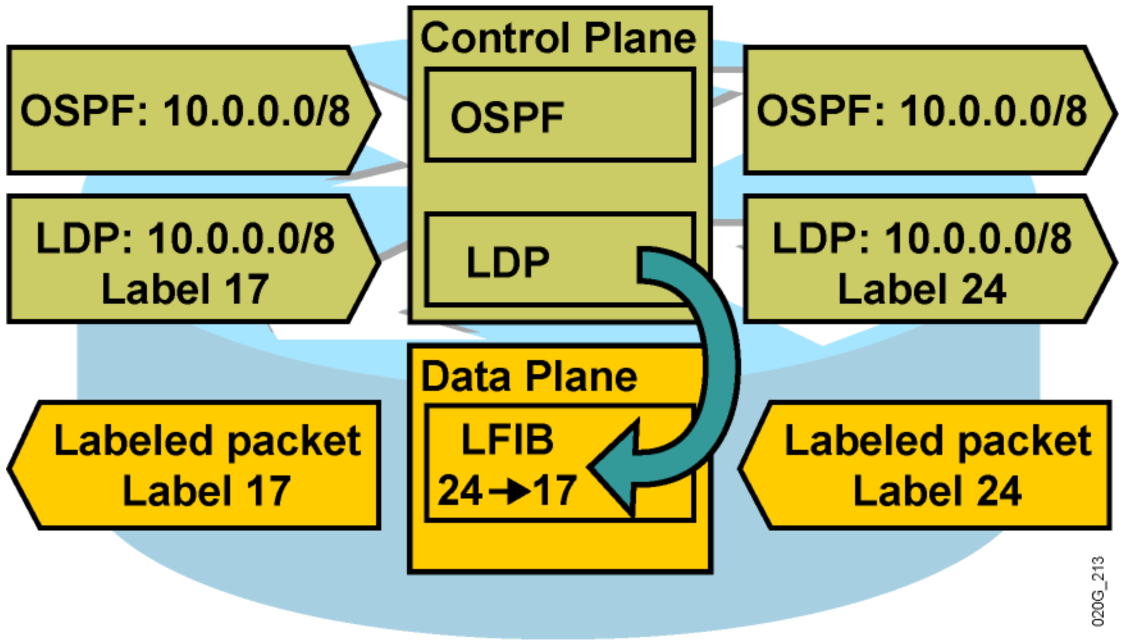
  Data plane forwards packets based on CEF (LFIB)

MPLS Architecture



Population of RIB/FIB/LIB/LFIB in an MPLS router

MPLS Architecture (Cont.)



Control plane and Data plane in action

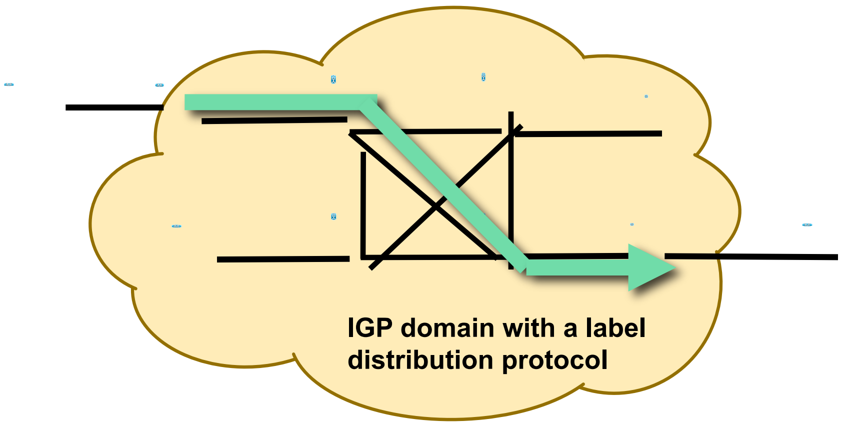
MPLS Terminology: FEC

- FEC (Forwarding Equivalence Class)

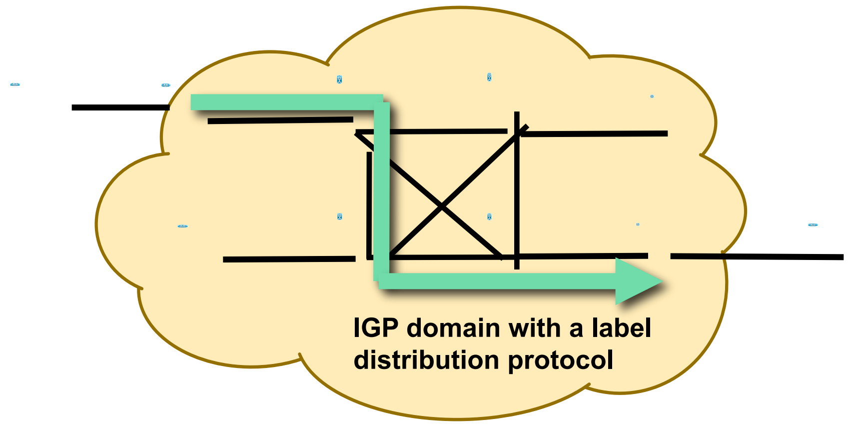
  Group of IP packets forwarded in the same manner (e.g. over same forwarding path)

- A FEC can represent a: Destination IP prefix, VPN ID, ATM VC, VLAN ID, Traffic Engineering tunnel, Class of Service.

MPLS Terminology:
Label Switch Path (LSP)



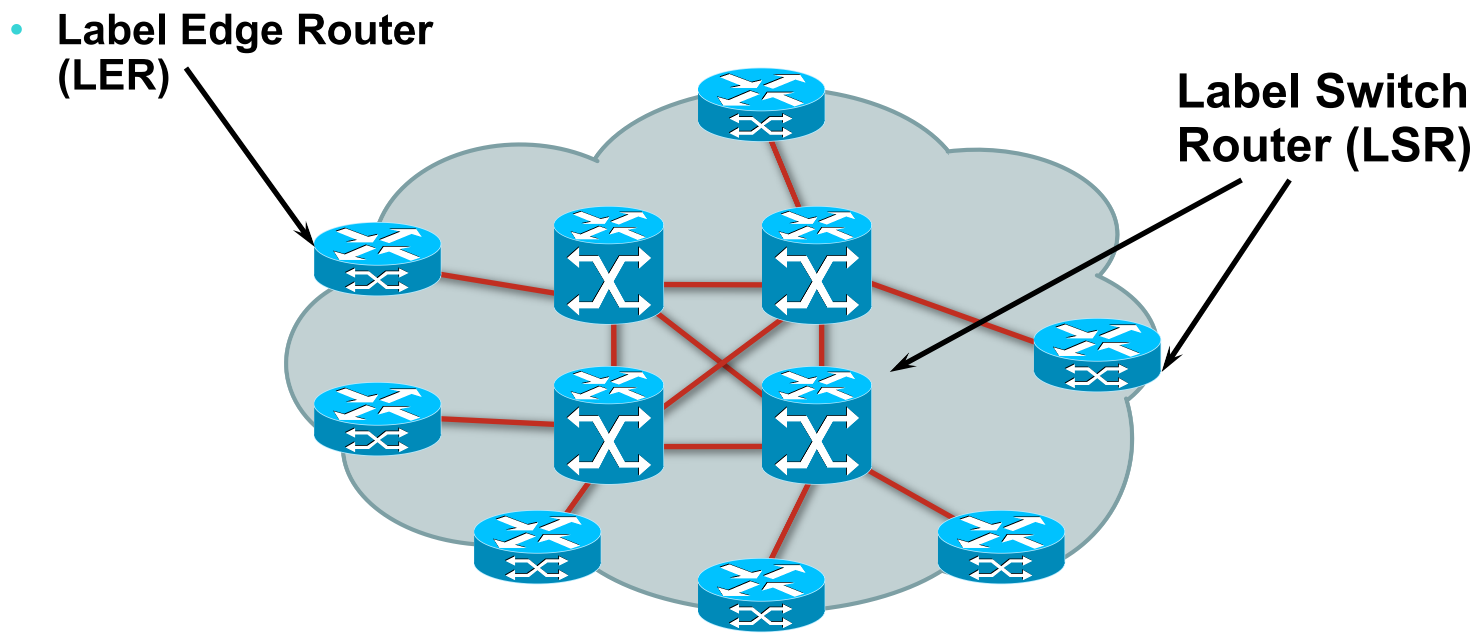
LSP follows IGP shortest path



LSP diverges from IGP shortest path

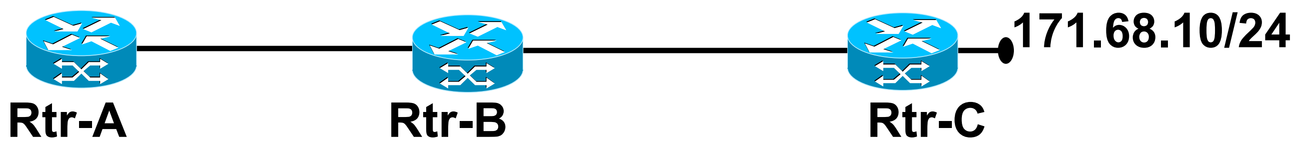
- **LSPs are derived from IGP routing information**
- **LSPs may diverge from IGP shortest path**
  - LSP tunnels (explicit routing) with TE
- **LSPs are unidirectional**
  - Return traffic takes another LSP

MPLS terminology:
LSR, LER



- LSR (Label Switch Router) is any netwrok router/switch running MPLS label switching
- LER (Label Edge Router) is an edge LSR. Also referred to as PE (Provider Edge) router

MPLS terminology: Upstream and Downstream LSRs



- Rtr-C is the downstream neighbor of Rtr-B for destination 171.68.10/24
- Rtr-B is the downstream neighbor of Rtr-A for destination 171.68.10/24
- LSRs know their downstream neighbors through the IP routing protocol

  Next-hop address is the downstream neighbor

MPLS Labels

MPLS uses a 32-bit label field that is inserted between Layer 2 and Layer 3 headers
(frame-mode).

MPLS with ATM uses the VPI, VCI fields of the ATM header as the label (cell-mode).

MPLS Labels:
Label Format (Shim header)

Label = 20 bits
COS/EXP = Class of Service, 3 bits
S = Bottom of Stack, 1 bit
TTL = Time to Live (Loop detection)

MPLS Labels:
Special Label values

- <u>SPECIAL LABEL VALUES</u>
- 0 – IPv4 Explicit Null
- 1 – Router Alert
- 2 – IPv6 Explicit Null
- 3 – Implicit Null

MPLS Labels:
Frame Mode Label Encapsulation

PPP Header
(Packet over SONET/SDH)

PPP Header	Shim Header	Layer 3 Header

LAN MAC Label Header

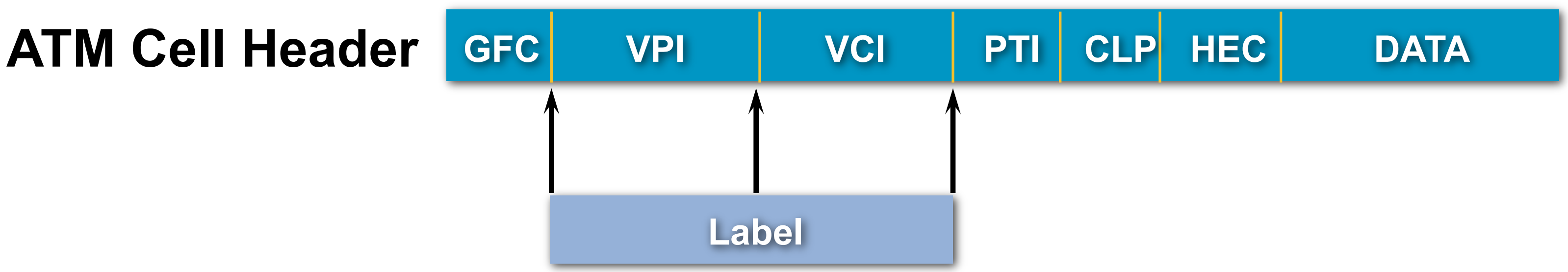
MAC Header	Shim Header	Layer 3 Header

- Shim header is used with Ethernet, 802.3, or PPP frames
- Sits between the Layer 2 and Layer 3 header
- L2 frame has

  ethertype=0x8847 to indicate frame carrying MPLS unicast packet

  ethertype=0x8848 to indicate frame carrying MPLS unicast packet

MPLS Labels:
Cell mode Label Encapsulation



- ATM switches forward cells, not packets

- In case of label stack:
- First level label could be in VPI
- Second level label could be in VCI

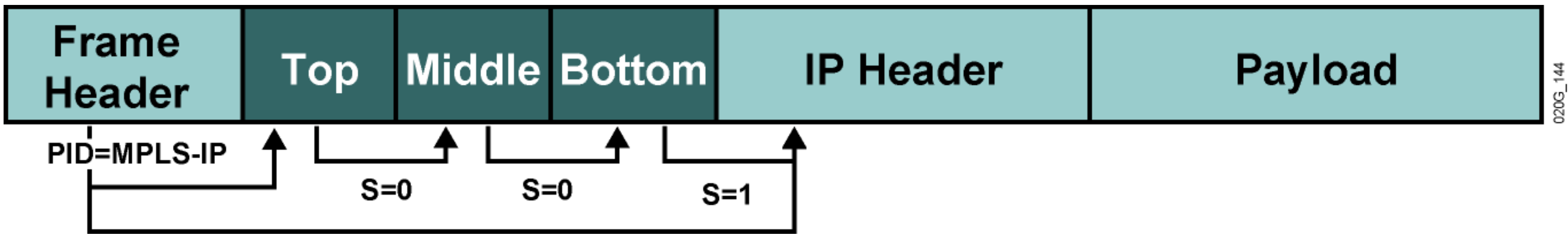
MPLS Labels: Label Assignment

- Labels have local significance

  Each LSR binds his own label mappings

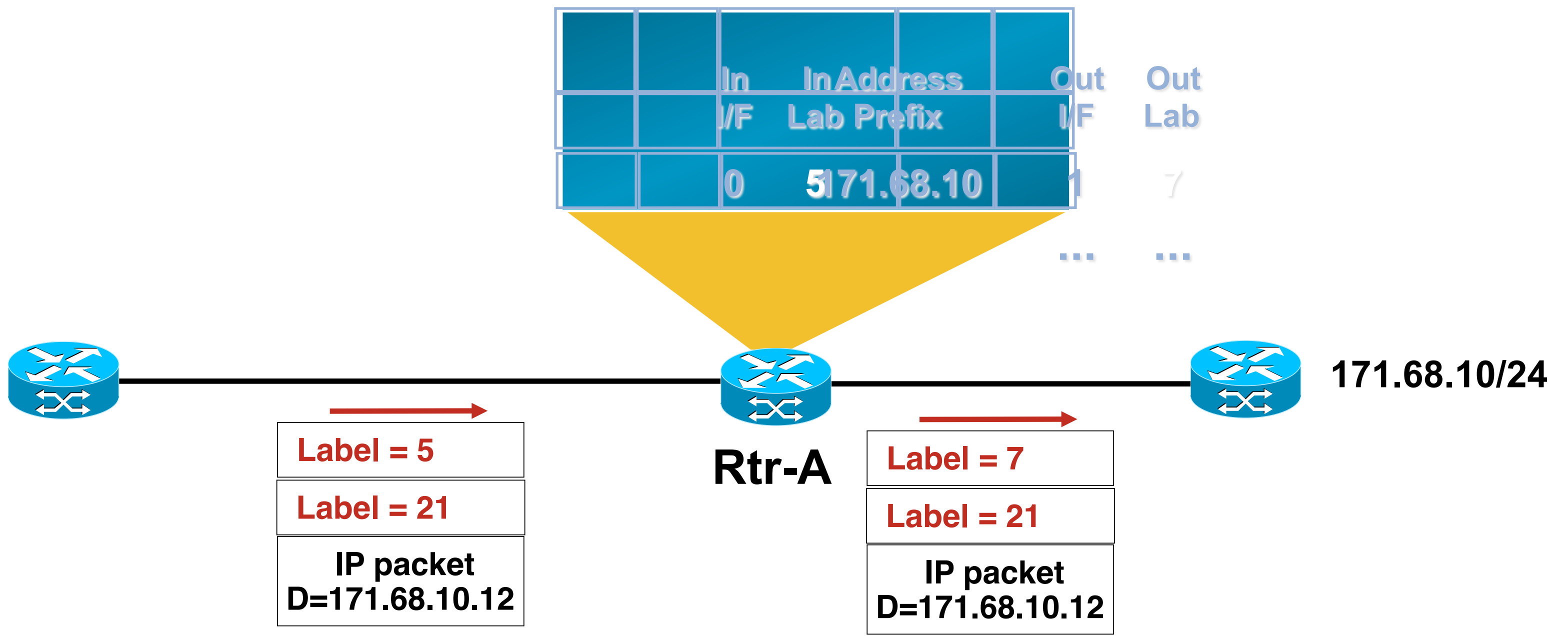
- LSR assigns labels to prefixes learnt in the routing table
- < Label, prefix, prefix mask > are exchanged between adjacent LSRs

MPLS Labels:
Label Stack



- An MPLS packet may have more than one label
- Frame Mode can handle a stack of two or more labels, depending on the platform
- Bottom most label has the S-bit set to 1
- ATM cells can have a stack of labels in the VPI, VCI fields
- LSRs label switch packets based ONLY on the label at the top of the stack

MPLS Labels:
Label Stack (Cont..)



- Rtr-A forwards the labelled packet based on the label at the top of the label stack

MPLS Labels: Label Stack (Cont..)

The following scenarios may produce more than one label:

- MPLS VPNs (two labels: The top label points to the egress router and the second label identifies the VPN.)
- MPLS TE with Fast Reroute (two or more labels: The top label is for the backup tunnel and the second label points to the primary tunnel destination.)
- MPLS VPNs combined with MPLS TE (three or more labels.)

MPLS Labels:
Label Distribution modes

- **Unsolicited**

  Downstream LSR advertises Label Binding to all adjacent LSRs, irrespective of whether they demand the Label binding or not

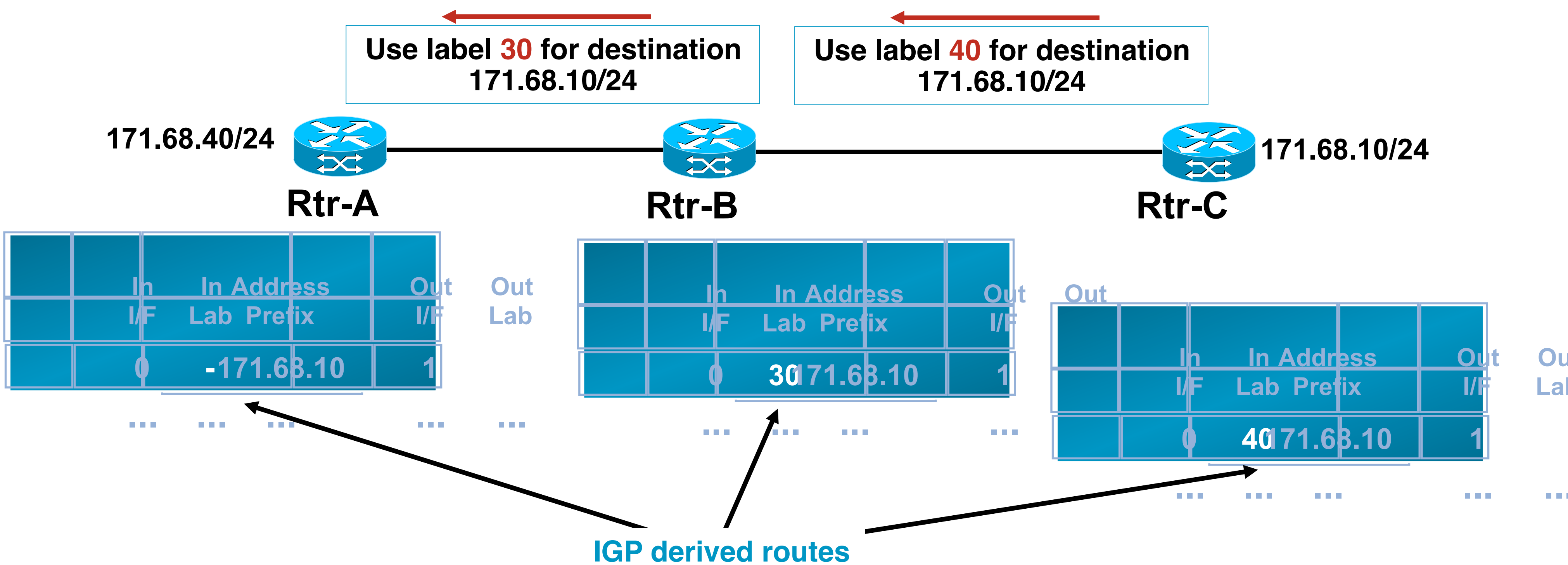
  Example: LDP, MP-iBGP

- **On-demand**

  Downstream LSR advertises Label Binding to those adjacent LSRs, who demand the Label binding

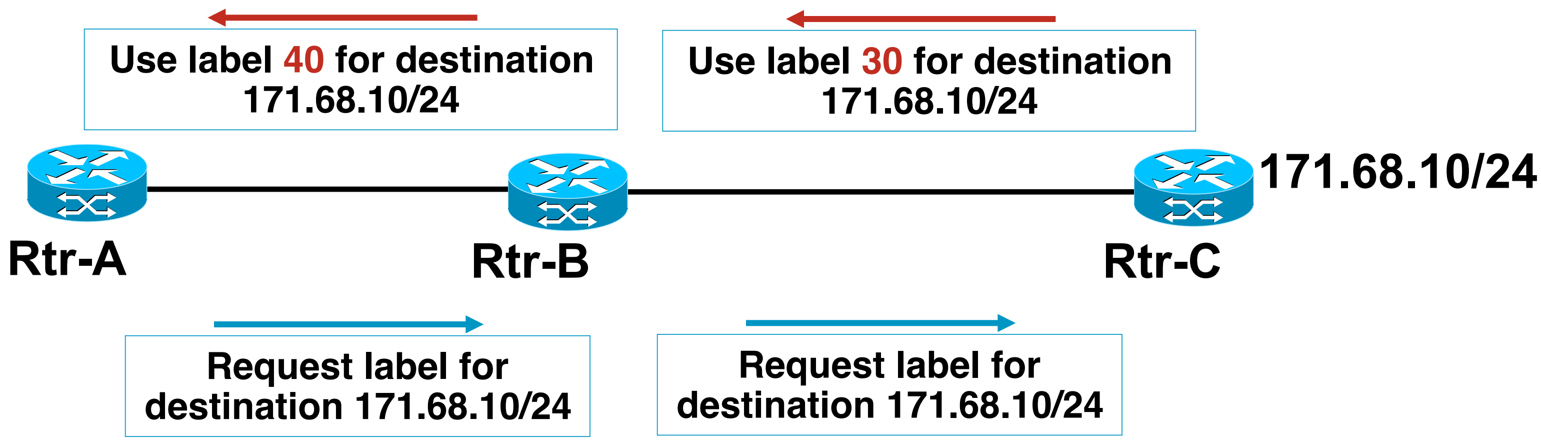
  Example: RSVP-TE, ATM

MPLS Labels:
Unsolicited example



- LSRs distribute labels to the upstream neighbors

MPLS Labels:
Downstream on-demand example



- Upstream LSRs request labels to downstream neighbors
- Downstream LSRs distribute labels upon request

MPLS Labels:
Control modes

- **Independent LSP control**

  LSR binds a Label to a FEC independently, whether or not the LSR has received a Label the next-hop for the FEC

  The LSR then advertises the Label to its neighbor

  Example: LDP

- **Ordered LSP control**

  LSR only binds and advertise a label for a particular FEC if:

    it is the egress LSR for that FEC or

    it has already received a label binding from its next-hop

  Example: RSVP-TE

MPLS Labels: Retention modes

- **Liberal retention**
  - **LSR retains labels from all neighbors**
    - In case, the next-hop LSR disappears, LSR already has the Out Label for the next best next-hops
    - Quick convergence
    - Requires more memory and label space
    - Example: LDP
- **Conservative retention**
  - **LSR retains labels only from next-hops neighbors**
    - LSR discards all labels for FECs which are not routing next-hops
    - Free memory and label space
    - Example: ATM cell mode

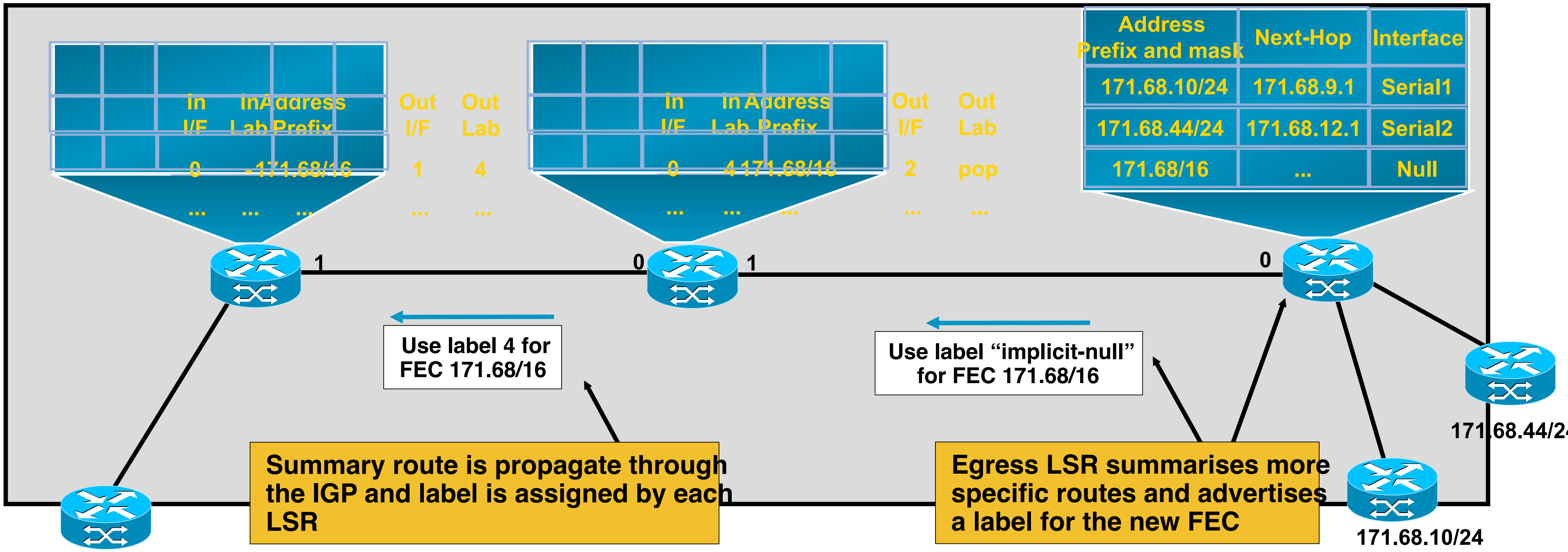
MPLS Labels:
Penultimate Hop Popping

- The label at the top of the stack is removed (popped) by the upstream neighbor of the egress LSR
- The egress LSR requests the “popping” through the label distribution protocol

  Egress LSR advertises *implicit-null* label

- One lookup is saved in the egress LSR

MPLS Labels: Penultimate Hop Popping Example

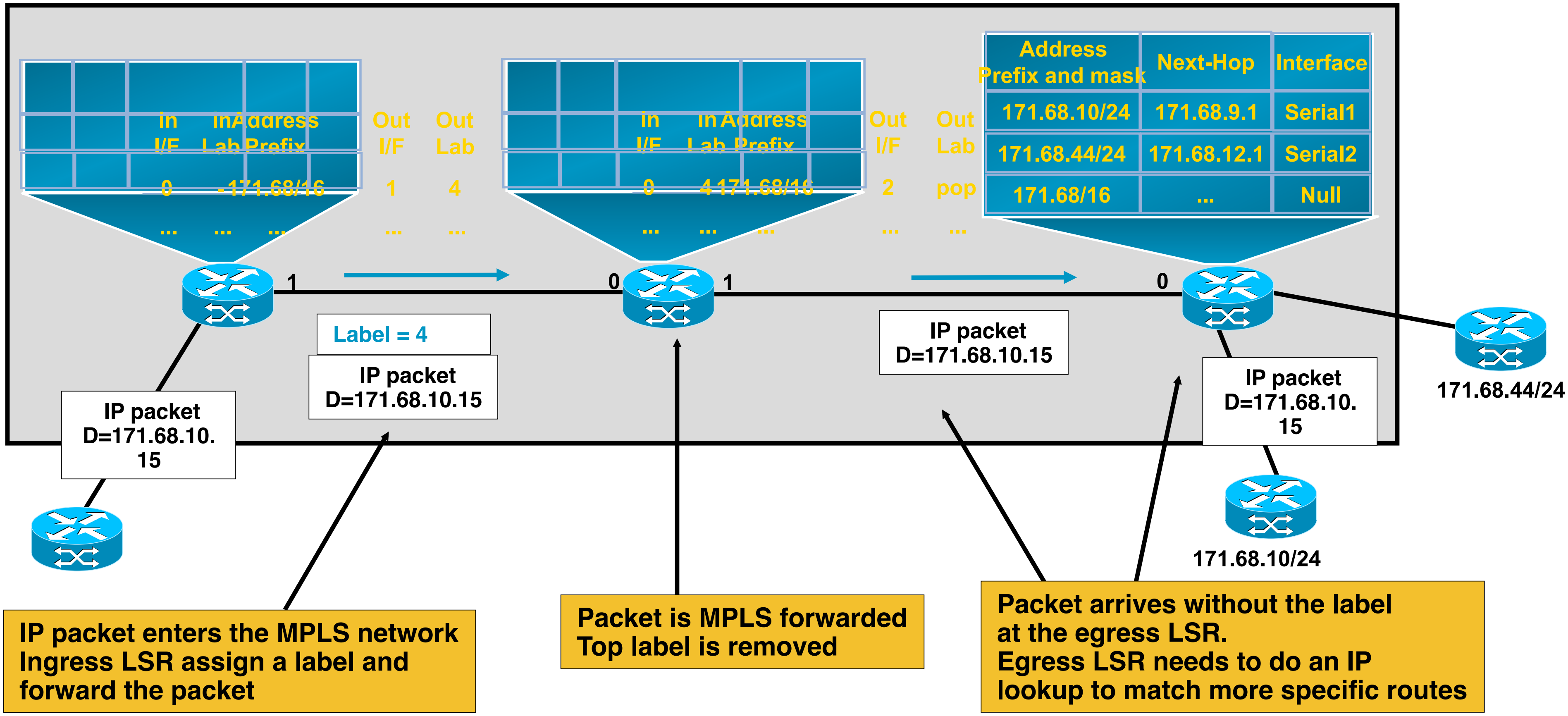


Egress LSR needs to do an IP lookup for finding more specific route

Egress LSR need NOT to receive a labelled packet

labelled will have to be popped anyway

MPLS Labels:
Penultimate Hop Popping Example (contd.)



Label Switch Routers: Architecture of LSRs

- LSRs, regardless of the type, perform these functions:
  - Exchange routing information
  - Exchange labels
  - Forward packets or cells

The first two functions are part of the control plane.

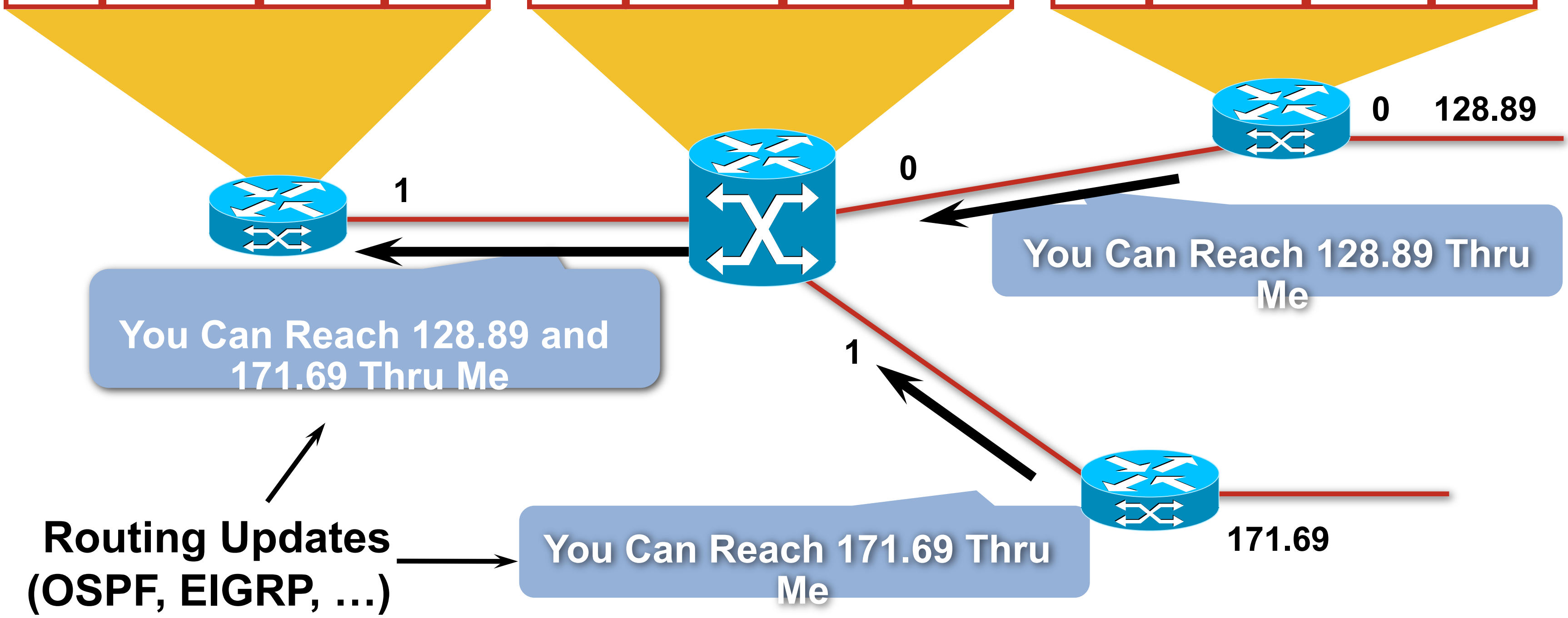
The last function is part of the data plane.

Label Switch Routers: Exchanging Routing updates

In Label	Address Prefix	Out I'face	Out Label
	128.89	1	
	171.69	1	
	...	...	

In Label	Address Prefix	Out I'face	Out Label
	128.89	0	
	171.69	1	
	...	...	

In Label	Address Prefix	Out I'face	Out Label
	128.89	0	
	...	...	

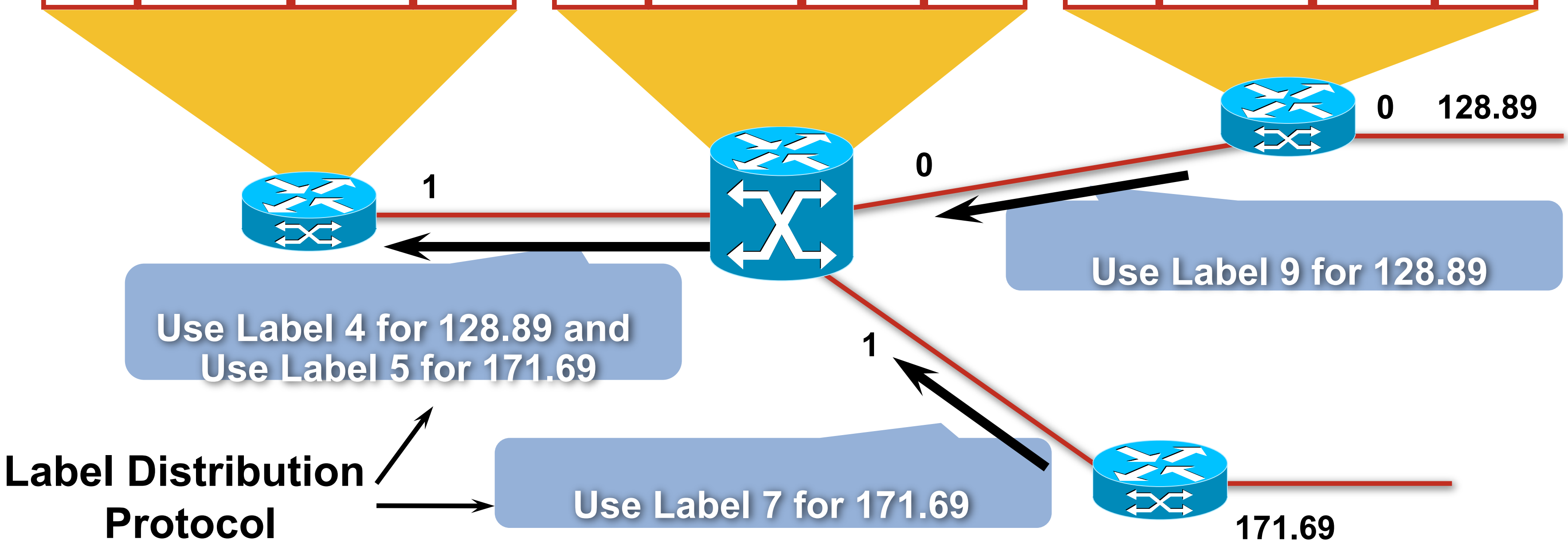


Label Switch Routers: Exchanging and Assigning Labels

In Label	Address Prefix	Out I'face	Out Label
-	128.89	1	4
-	171.69	1	5
...	...	...	...

In Label	Address Prefix	Out I'face	Out Label
4	128.89	0	9
5	171.69	1	7
...	...	...	...

In Label	Address Prefix	Out I'face	Out Label
9	128.89	0	-
...	...	...	...



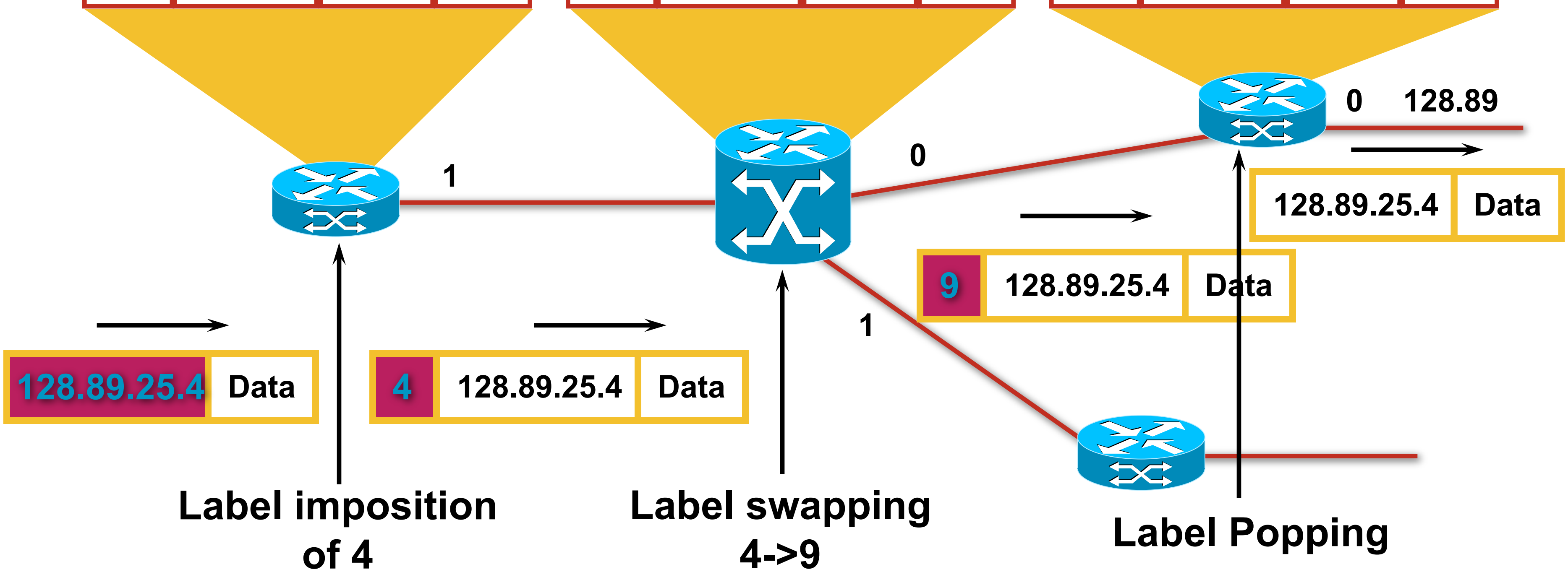
- In Label is the local label generated by the LSR
- Out Label is the remote label advertised by the adjacent LSR, which is the IGP next hop

Label Switch Routers: Forwarding Packets

In Label	Address Prefix	Out I'face	Out Label
-	128.89	1	4
-	171.69	1	5
...	...	...	...

In Label	Address Prefix	Out I'face	Out Label
4	128.89	0	9
5	171.69	1	7
...	...	...	...

In Label	Address Prefix	Out I'face	Out Label
9	128.89	0	-
...	...	...	...



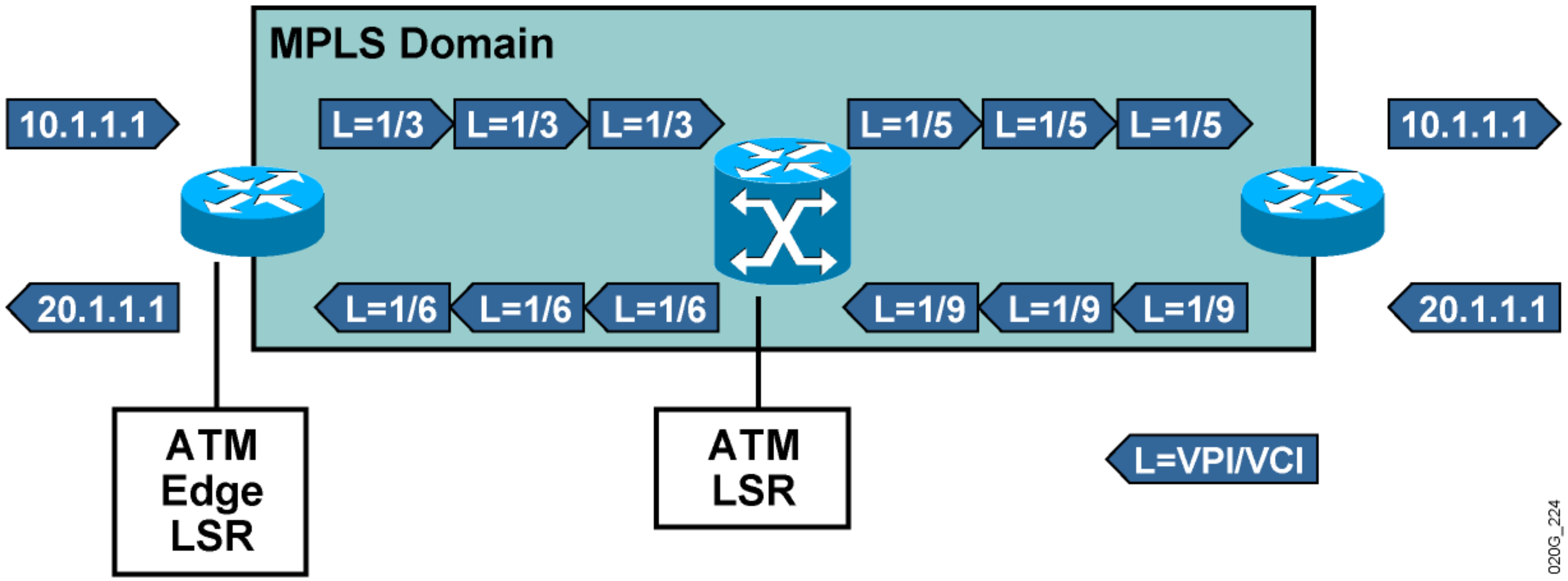
Label Switch Routers: Label functions

An LSR can perform the following functions:

- Insert (impose) a label or a stack of labels on ingress
- Swap a label with a next-hop label or a stack of labels in the core
- Remove (pop) a label on egress

ATM LSRs can swap a label with only one label (VPI/VCI fields change).

Label Switch Routers: Cell Mode



ATM LSR can forward only cells.

ATM edge LSR segments packets into cells and forwards them into an MPLS ATM domain, or reassembles cells into packets and forwards them out of an MPLS ATM domain.

MPLS Forwarding Plane: LFIB and Outgoing Labels

```
RSP-PE-SOUTH-5#show mpls forwarding-table 10.13.1.11
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
59     46          10.13.1.11/32     0          Se10/0/0   point2point
RSP-PE-SOUTH-5#
```

- Outgoing label tells what treatment the packet is going to get. It could also be -
  - I. Pop - Pops the topmost label
  - II. Untagged - Untag the incoming MPLS packet
  - III. Aggregate - Untag and then do a FIB lookup
  - IV. 0 - Nullify the top label (first 20bits)
- Label values 0-15 are reserved.

MPLS Forwarding Plane: Outgoing label types

```
PE1#sh mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     2002        10.13.1.22/32     0          Et0/0      10.13.1.5
       2002        10.13.1.22/32     0          Et1/0      10.13.1.9
17     2001        10.13.1.62/32     0          Et0/0      10.13.1.5
       2001        10.13.1.62/32     0          Et1/0      10.13.1.9
18     Pop tag     10.13.1.101/32    0          Et1/0      10.13.1.9
       Pop tag     10.13.1.101/32    0          Et0/0      10.13.1.5
19     Pop tag     10.13.2.4/30      0          Et1/0      10.13.1.9
       Pop tag     10.13.2.4/30      0          Et0/0      10.13.1.5
20     Untagged    5.5.5.5/32[V]     0          Se2/0      point2point
21     Pop tag     10.13.21.4/30     0          Et1/0      10.13.1.9
       Pop tag     10.13.21.4/30     0          Et0/0      10.13.1.5
22     Pop tag     10.13.22.4/30     0          Et1/0      10.13.1.9
       Pop tag     10.13.22.4/30     0          Et0/0      10.13.1.5
23     Aggregate   0.0.0.0/0[V]      0
24     Aggregate   200.1.61.4/30[V]  0
26     Untagged    30.30.30.1/32[V]  0          Se2/0      point2point
PE1#
```

V means it is a VPN prefix

MPLS Forwarding Plane: Outgoing label types (cont.)

- Untagged

  Convert the incoming MPLS packet to an IP packet and forward it.

- Pop

  Pop the top label from the label stack present in an incoming MPLS packet and forward it as an MPLS packet. If there was only one label in the stack, then forward it as an IP packet. SAME as imp-null label.

- Aggregate

  Convert the incoming MPLS packet to an IP packet and then do a FIB lookup for it to find out the outgoing interface.

- 0 (zero)

  Same as exp-null label. Simplify fills 0 in the first 20 bits of label; helps to preserve the EXP value of the top label.

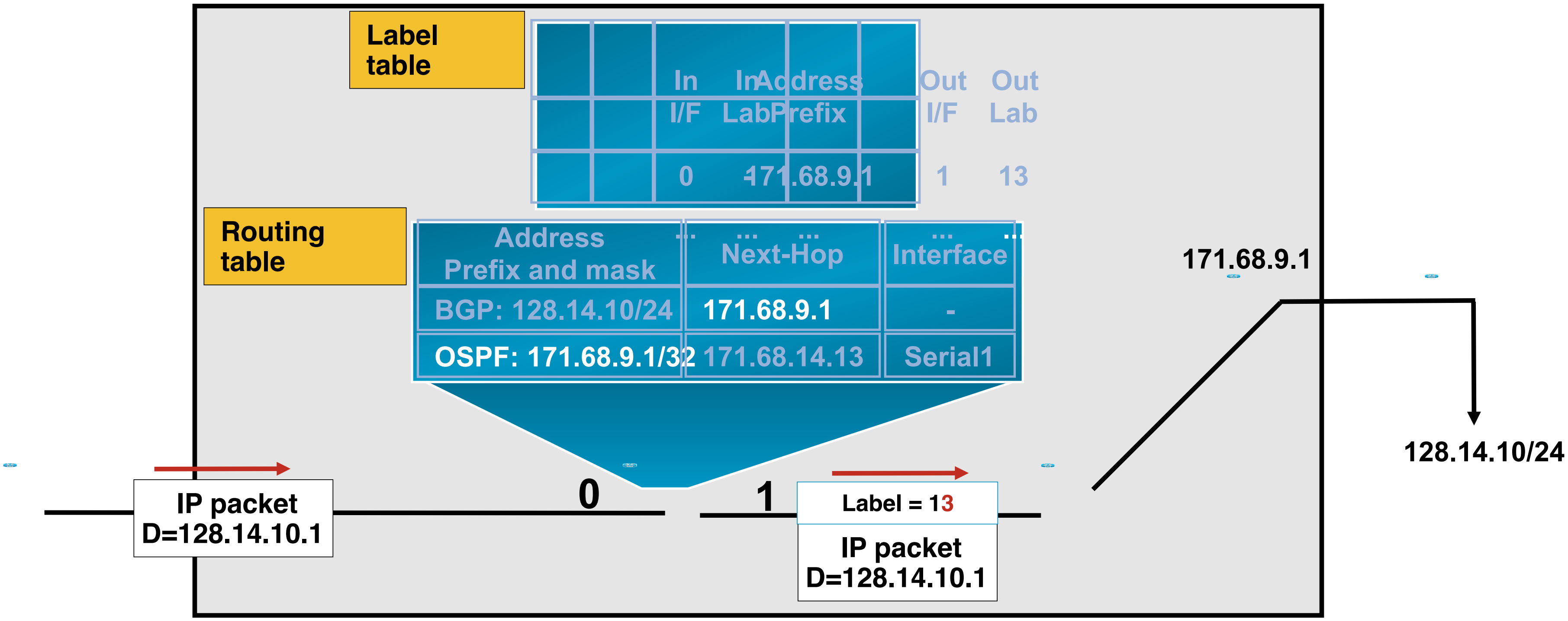
MPLS and BGP

- Labels are assigned to FECs which are derived from IP routing protocols (IGP)
- Labels are NOT assigned to BGP routes
- BGP routes use recursive routing to find next-hop reachability
- Labels are assigned to BGP next-hops
- This saves CPU/Memory, label space and stability on core LSRs

  Core LSRs are preserved from BGP instability

- We can assign labels to BGP learnt routes based on RFC 3107

MPLS and BGP (cont.)



Ingress LSR receives IP packet

Destination is given by BGP

BGP has next-hop known in the IGP

Label is available for BGP next-hop, through IGP route

Packet will traverse the core using IGP (BGP next-hop) label

Label Distribution Protocols

Several protocols for label exchange

- TDP/LDP
- RSVP
- BGP

Summary

MPLS LSRs have separate control planes and forwarding planes

Labels can be in Shim header or as part of ATM header

Labels have adevrtisement modes, retention modes & control modes

All LSRs perform three functions:

- Exchange routing information
- Exchange labels
- Forward packets or cells (depending on type) based on labels

There are several label distribution protocols

LDP Introduction

LDP Basics and Session establishment

Agenda

- LDP Concepts
- LDP Identifier
- LDP PDU
- LDP Messages
- LDP Session Establishment
- LDP Sessions between ATM LSRs
- Targeted LDP sessions
- Summary

LDP Concepts

- Label Distribution Protocol
- LDP works between adjacent/non-adjacent peers
- LDP sessions are established between peers
- LDP messages sent in the form of TLVs

  <Type, Length, Value>
- Standardized via RFC 3036

TDP/LDP Transport

- Uses TCP for reliable transport
- Well-known TCP port

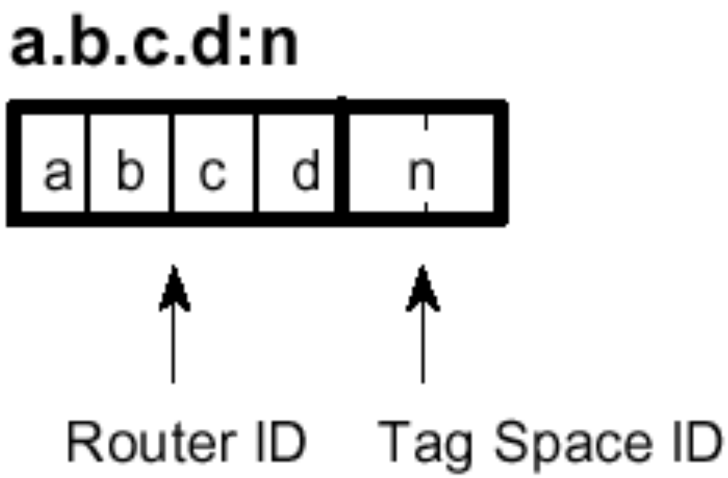
  LDP (port 646)

  TDP (port 711)
- LSR with higher LDP router-id opens a connection to port 646 of other LSR
- Design Choice:

  One TDP/LDP session per TCP connection

LDP Identifier

- Identifies tag space
- 6 bytes (4 bytes =>IP address, 2 bytes =>Label space ID)

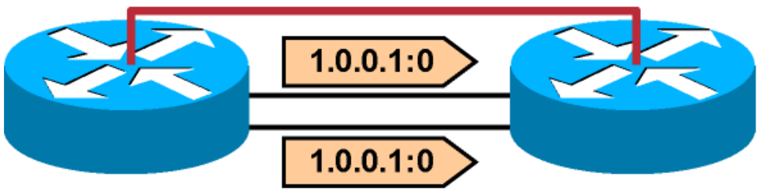


LDP Identifier:
Label Space

- LSRs establish one LDP session per label space.
  - Per-platform label space requires only one LDP session, even if there are multiple parallel links between a pair of LSRs.
- Per-platform label space is announced by setting the label space ID to 0, for example:
  - LDP ID = 1.0.0.1:0
- A combination of frame-mode and cell-mode MPLS, or multiple cell-mode links, results in multiple LDP sessions.

Label Space and number of LDP sessions

Frame Mode



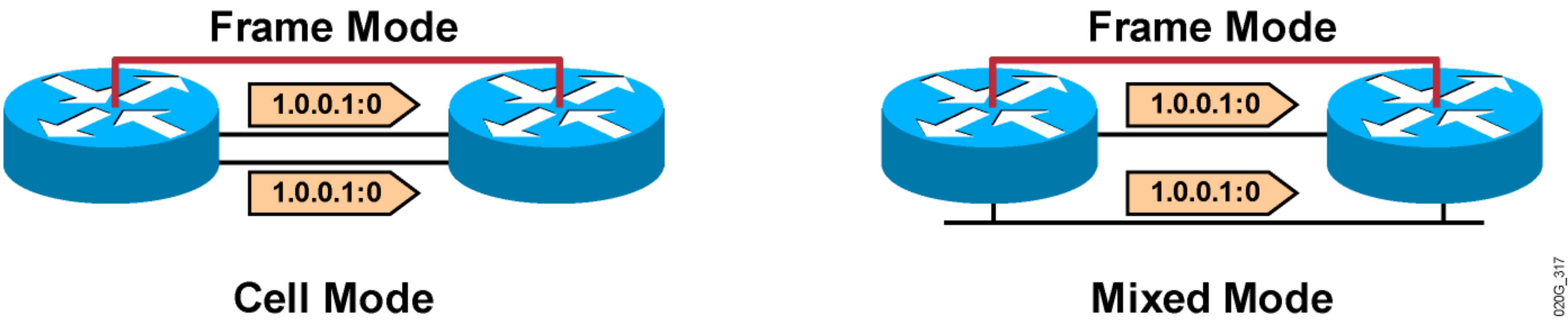
Frame Mode

Cell Mode

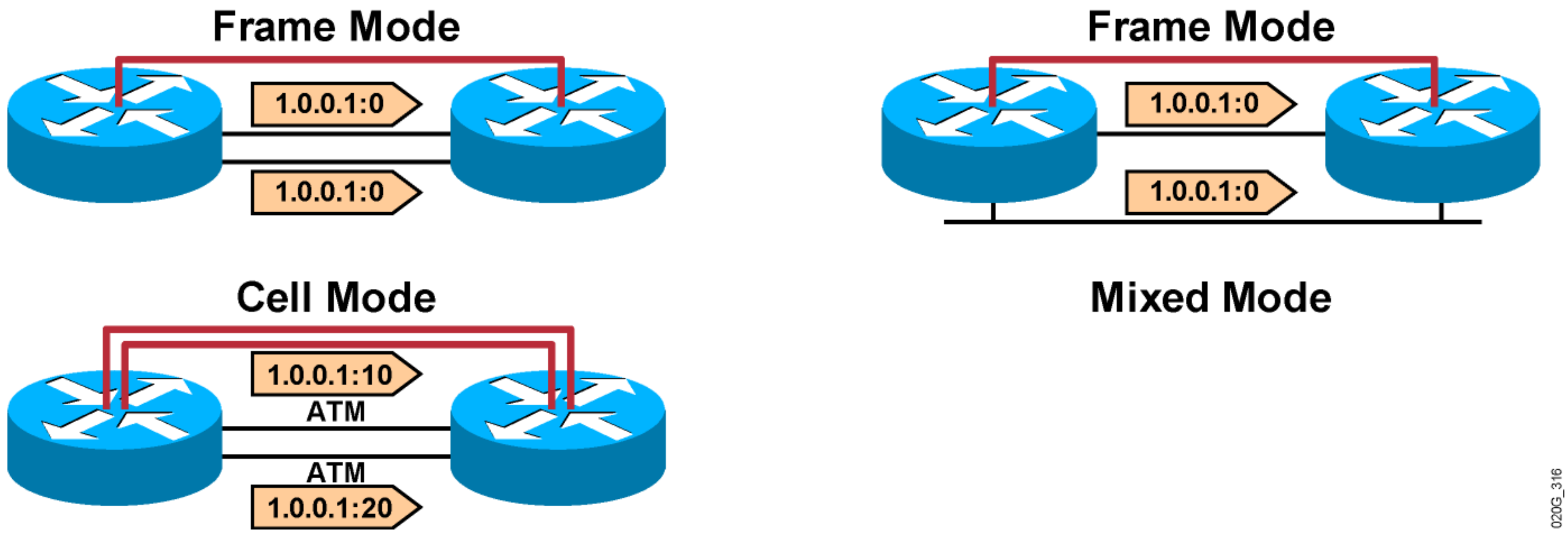
Mixed Mode

020G_318

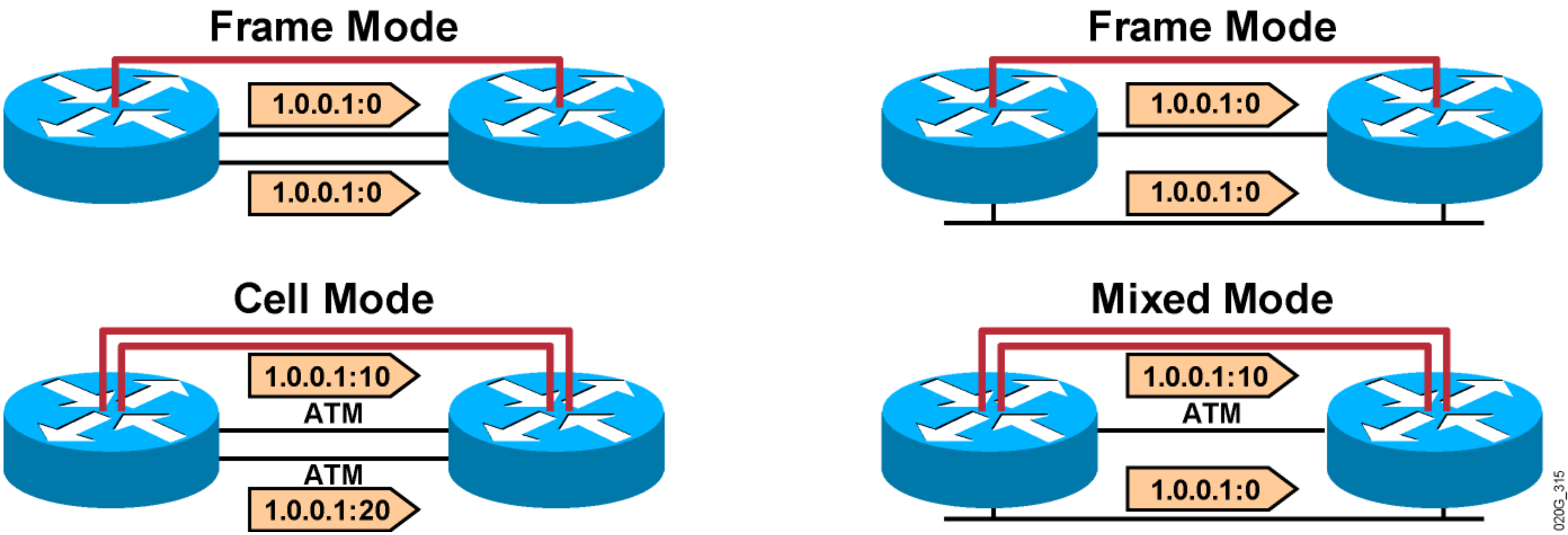
Label Space and number of LDP sessions (Cont.)



Label Space and number of LDP sessions (Cont.)



Label Space and number of LDP sessions (Cont.)



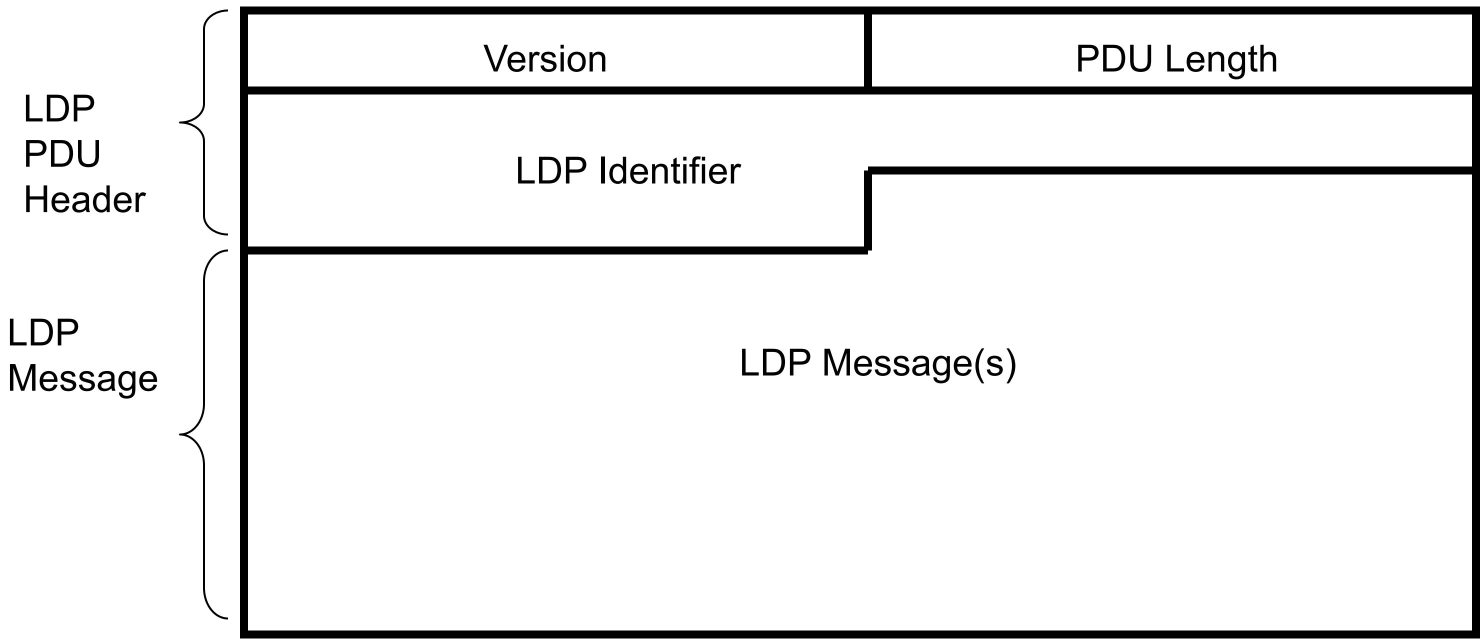
One LDP session is established for each announced LDP identifier (router ID + label space).

The number of LDP sessions is determined by the number of different label spaces.

The bottom right example is not common, because ATM LSRs do not use Ethernet for packet forwarding, and frame-mode MPLS across ATM uses per-platform label space.

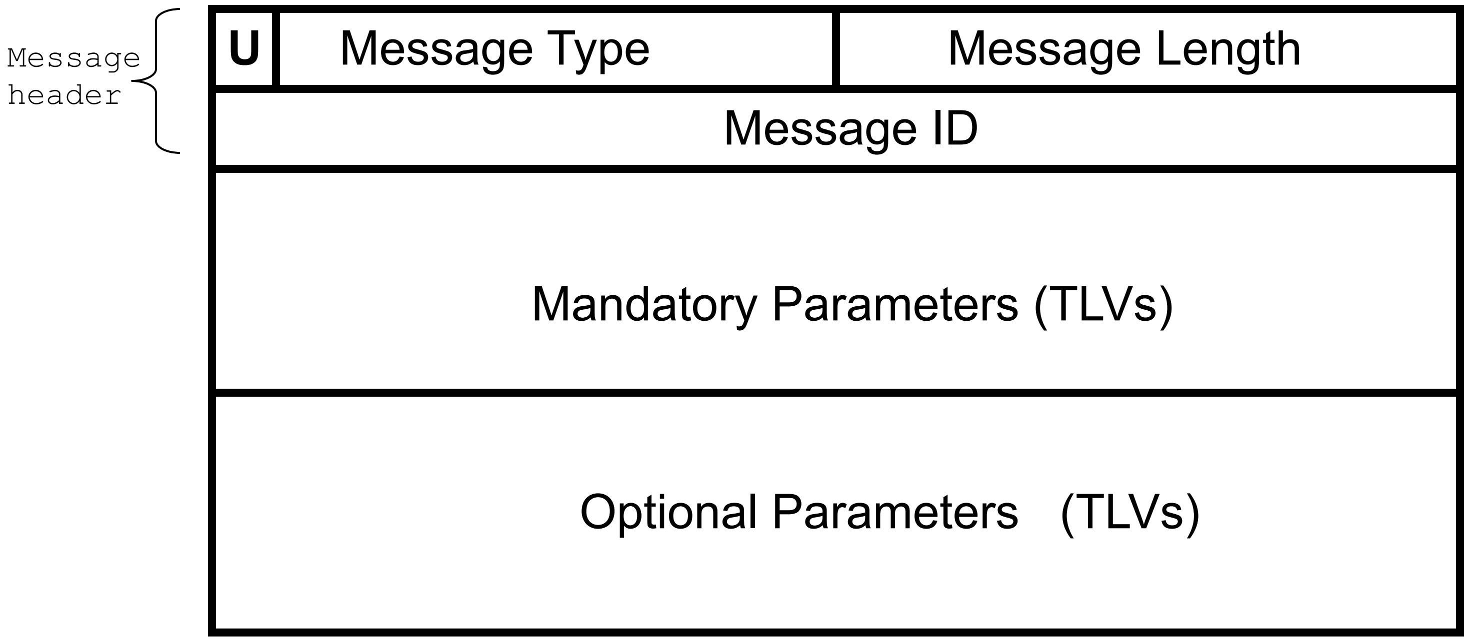
LDP Protocol Data Units

- All LDP information is sent in the form of PDUs over the TCP connection



•**Version** => LDP version. Current LDP version is 1
•**PDU Length** (excludes Version and PDU Length fields) => total length of PDU in bytes.
•**LDP Identifier** => discussed earlier
•**LDP Messages** => one or more LDP messages

LDP Message

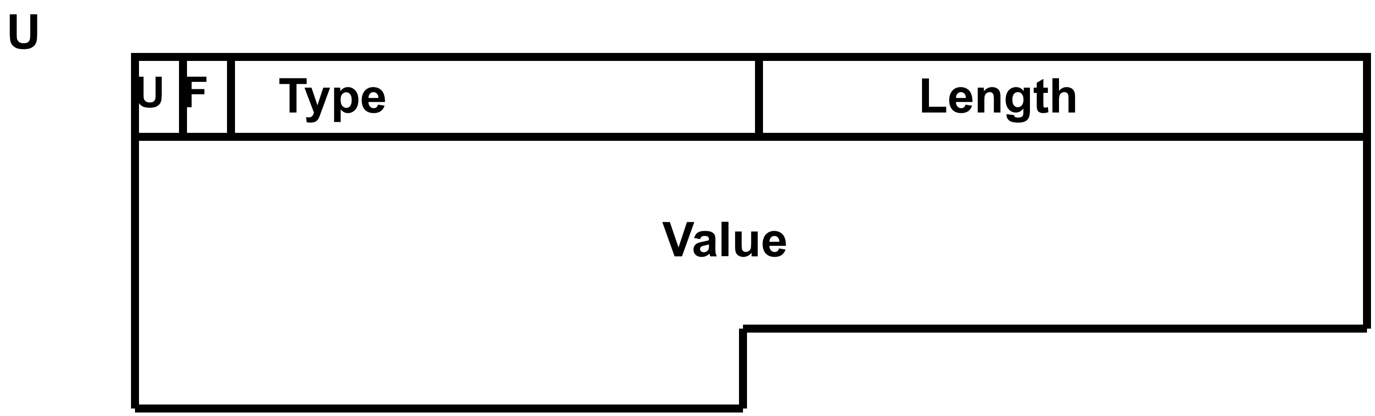


U bit is the Unknown Message bit. If the received message is of unknown type, then if:

U=0, send Notification Message to the originator of this message

U=1, silently ignore the unknown message

LDP Message TLVs



U bit is the Unknown TLV bit. If the received TLV is of unknown type, then if:

U=0, send Notification Message to the originator of this message and ignore the entire message

U=1, silently ignore the unknown TLV and process the rest of the message

F bit is the Forward unknown TLV bit. F bit is only applicable when the U=1

F=0,the unknown TLV is not forwarded with its LDP message

F=1,the unknown TLV is forwarded with its LDP message

LDP Messages Types

- **DISCOVERY messages**
- **ADJACENCY messages deal with initialization, keepalive & shutdown of sessions**
- **LABEL ADVERTISEMENT messages deal with label binding, requests, withdrawal & release**
- **NOTIFICATION messages provide advisory information & signal errors**

Discovery Message

- Used to discover and maintain the presence of new peers using HELLO messages
- Hello packets (UDP) sent to all-routers multicast address (224.0.0.2)
- Direct unicast hello is sent to non-adjacent neighbors
- Once session is established, HELLO messages serve as link integrity messages
- Session is bi-directional

Adjacency Messages

- **INITIALIZATION**
  - Two LSRs negotiate on various parameters & options
  - These include keepalive timer values, Label ranges, Unsolicited vs. On-demand label advertisement, Ordered vs. Independent mode, Liberal vs. Conservative Label retention
- **KEEPALIVE**
  - LDP message that indicates that neighbor is alive

Label Advertisement related messages

- **LABEL RELEASE**

  An LSR releases a Label Binding that it previously got from it's LDP peer. Used in Conservative Label Retention mode

- **LABEL REQUEST**

  Used by an upstream LSR to request a Label binding for a prefix from the downstream LDP peer. Used in downstream on-demand mode

- **LABEL ABORT REQUEST**

  Send to abort the LABEL REQUEST message

- **LABEL MAPPING**

  Are the TLV object containing <Label, prefix> information

- **LABEL WITHDRAWAL**

  Used to revoke a previously advertised label binding

Notification message

- NOTIFICATION

  Used for Error Notification and Advisory

LDP Session Establishment

LDP establishes a session by performing the following:

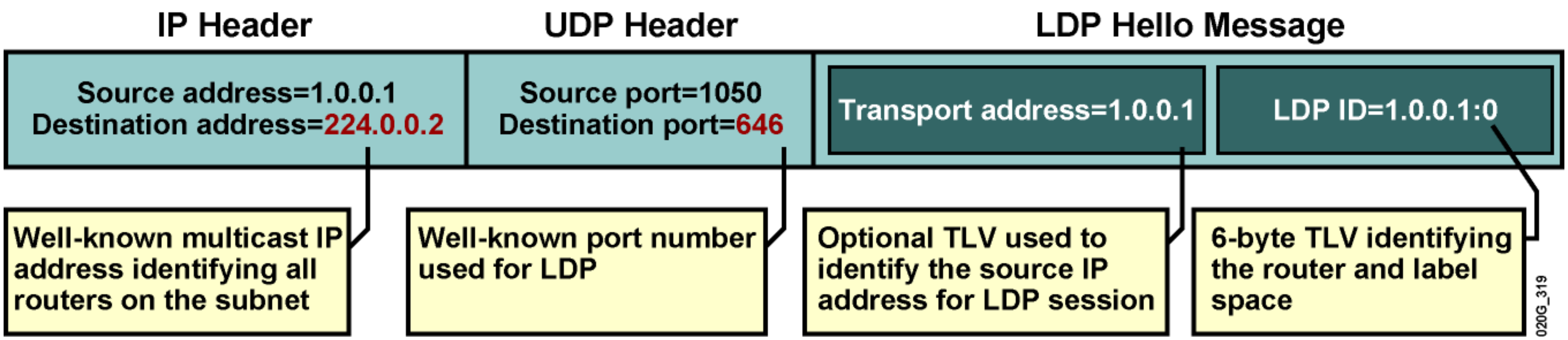
- Hello messages are periodically sent on all interfaces that are enabled for MPLS.
- If there is another router connected to that interface, that it also has MPLS enabled, it will respond by trying to establish a session with the source of the hello messages.

UDP is used for hello messages. It is targeted at “all routers on this subnet” multicast address (224.0.0.2).

TCP is used to establish the session.

Both TCP and UDP use well-known LDP port number 646 (711 for TDP).

LDP Hello Message



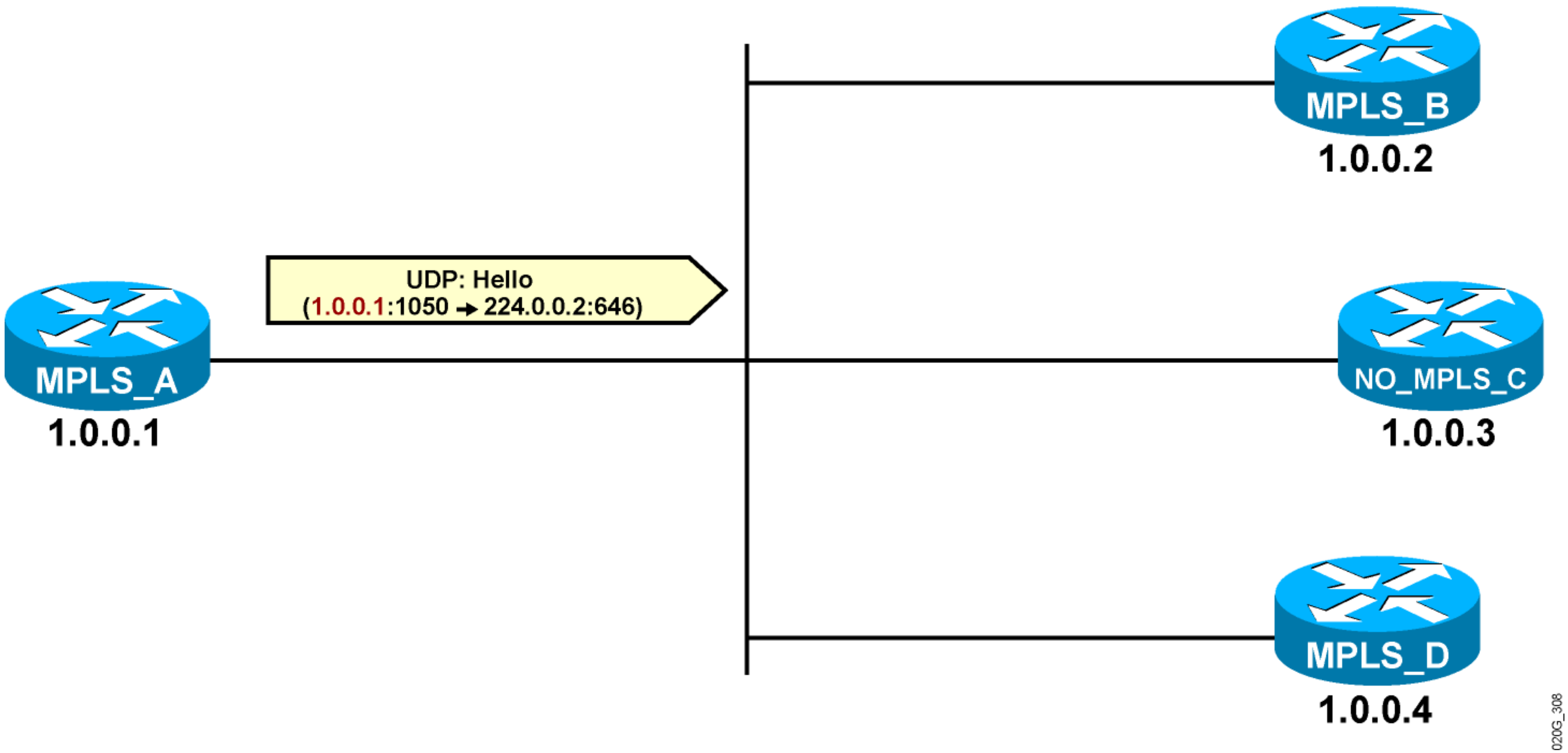
Hello messages are targeted at all routers reachable through an interface.

LDP uses well-known (UDP and TCP) port number 646.

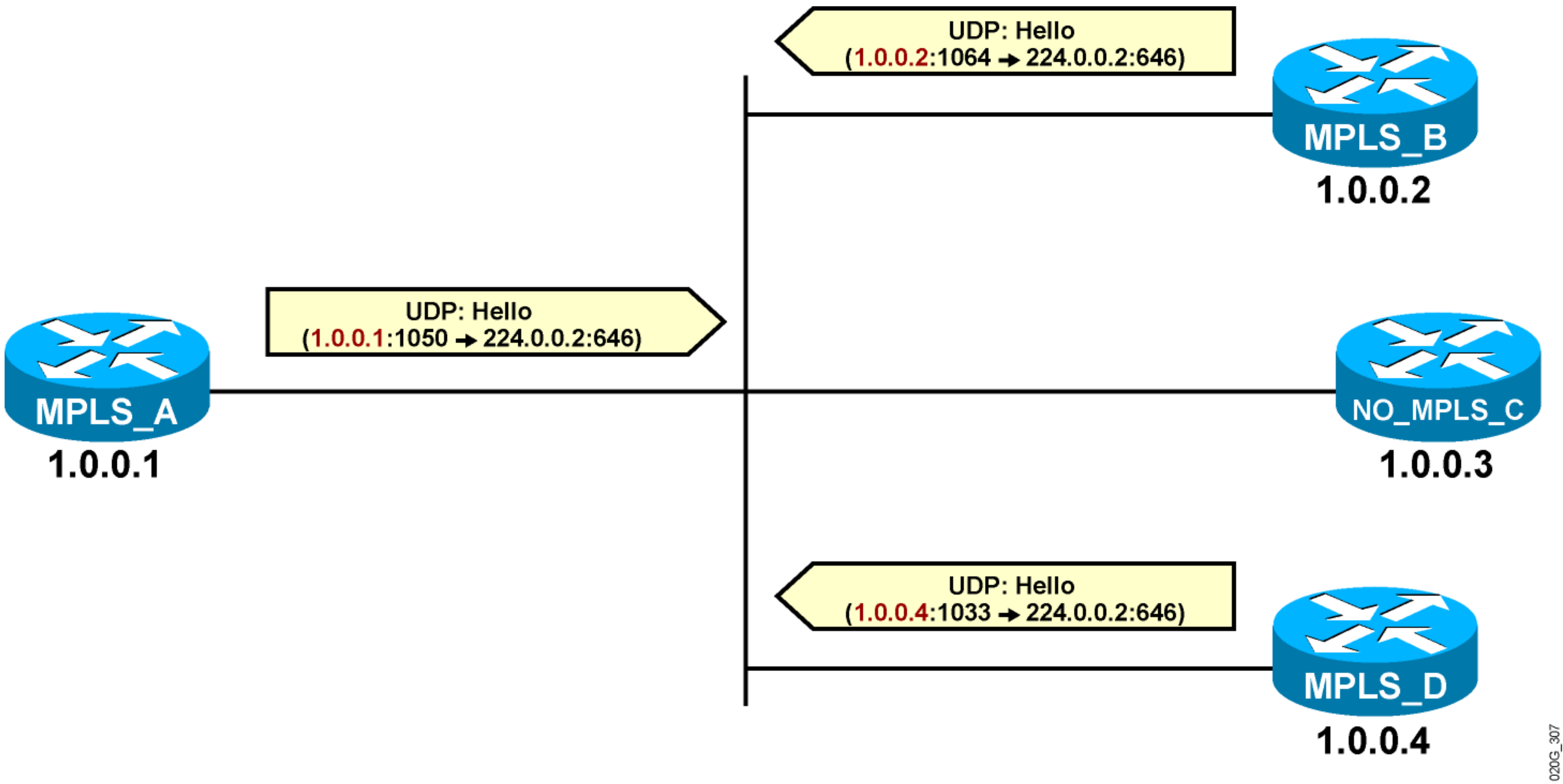
The source address used for an LDP session can be set by adding the transport address TLV to the hello message.

A 6-byte LDP identifier (TLV) identifies the router (first four bytes) and label space (last two bytes).

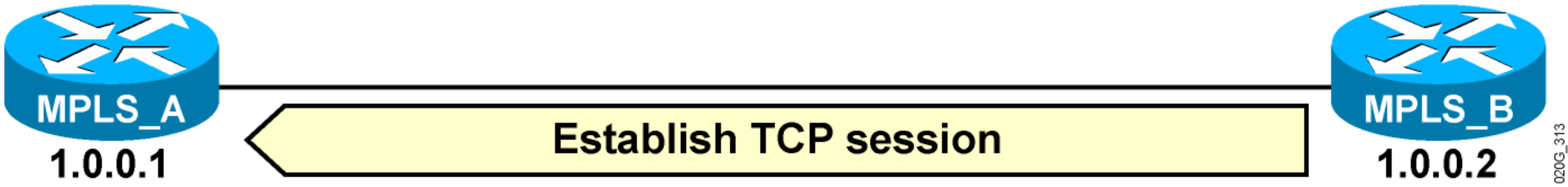
LDP Neighbor Discovery



LDP Neighbor Discovery

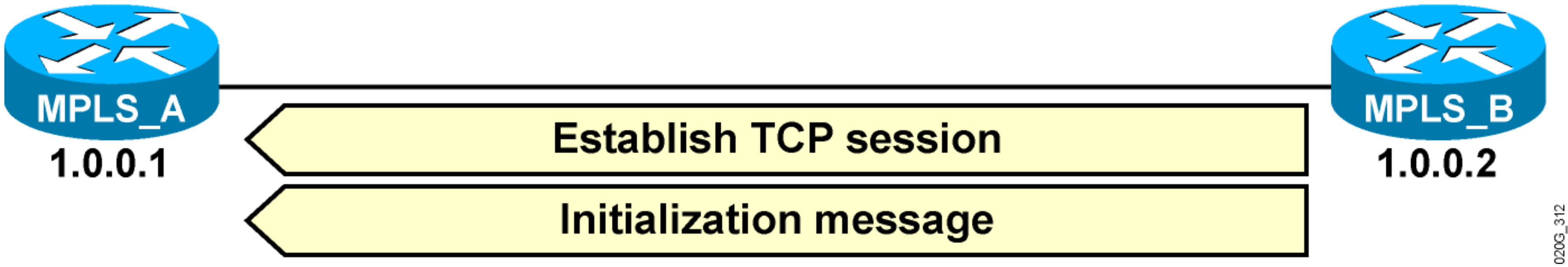


LDP Session: Transport Connection



- Once LDP peers receive hellos, they establish a TCP connection
- Peer with higher LDP router-id is active LSR and the peer with lower LDP router-id is the passive LSR
- Active LSR tries to open a TCP connection to the well-known LDP port number 646 of the passive LSR, while the passive LSR waits for the active LSR to initiate the connection

LDP Session: Session Initialization

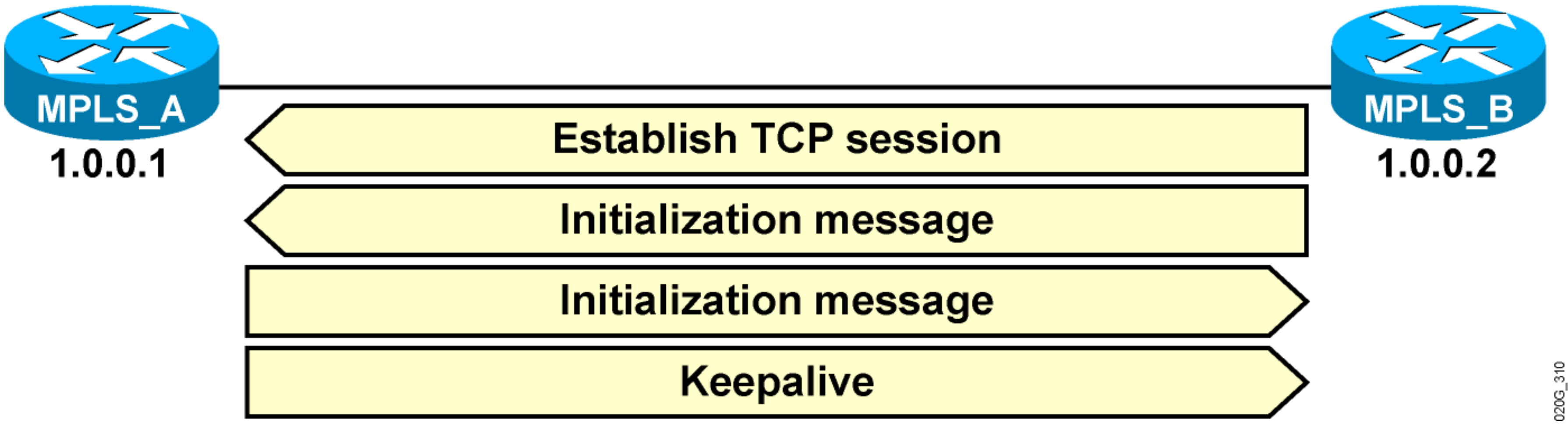


Active LDP peer (1.0.0.2) sends Initialization message to passive LDP peer

Initialization message contains important parameters:

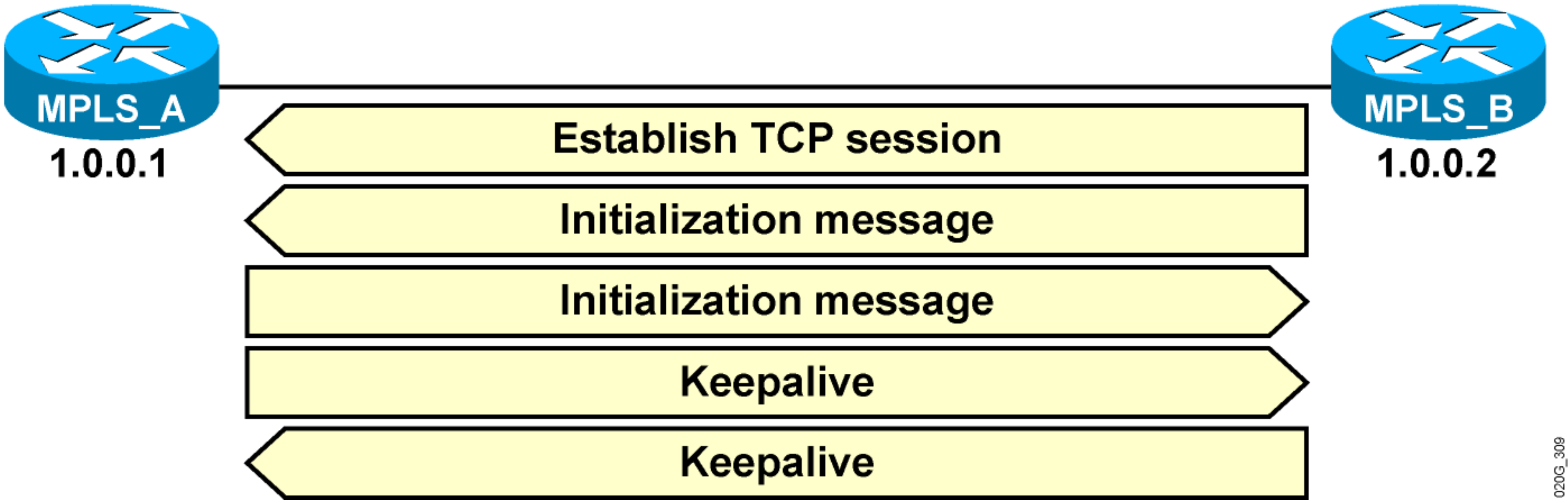
- Session keepalive time (default=180 sec)
- Label distribution method: Downstream unsolicited
- Max PDU length
- Receiver’s LDP Identifier
- Whether Loop Detection is enabled
- Some optional parameters

LDP Session:
Session Initialization (cont.)



- Passive LDP peer sends Initialization message and/or keepalive message to active LDP peer if Initialization message parameters are acceptable
- Passive LDP peer could also send Error Notification & close the LDP connection if something was unacceptable

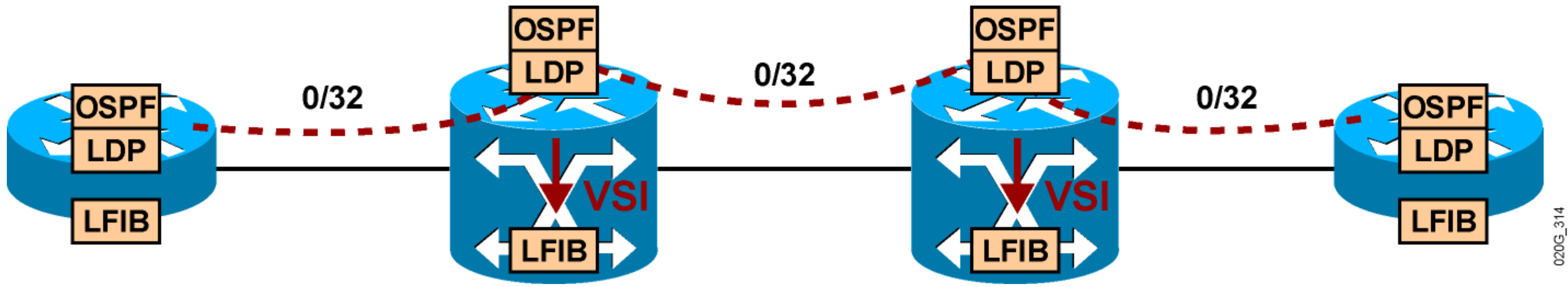
LDP Session: Session Initialization (cont.)



Active LDP peer sends keepalive to passive LDP peer & the LDP session is up

The session is ready to exchange label mappings after receiving the first keepalive.

LDP Sessions Between ATM LSRs



An IP adjacency between ATM LSRs is established through the control virtual circuit (0/32).

The control virtual circuit is used for LDP as well as for IP routing protocols.

VSI protocol is used to populate the ATM switching matrix (LFIB) in the data plane of some ATM switches (Cisco implementation).

Targeted LDP Sessions

LDP neighbor discovery of nonadjacent neighbors differs from normal discovery only in the addressing of hello packets:

- Hello packets use unicast IP addresses instead of multicast addresses.

When a neighbor is discovered, the mechanism to establish a session is the same.

Summary

TCP is used to establish LDP sessions between neighbors.

LDP uses PDUs to carry messages

LDP hello messages contain an identifier field that uniquely identifies the neighbor and the label space.

Per-platform label space requires only one LDP session.

Routers that have the higher IP address must initiate the TCP session.

LDP session negotiation is a three-step process.

LDP sessions between ATM LSRs use the control VPI/VCI, which by default is 0/32.

Nonadjacent neighbor discovery is accomplished by using unicast IP addresses instead of multicast.

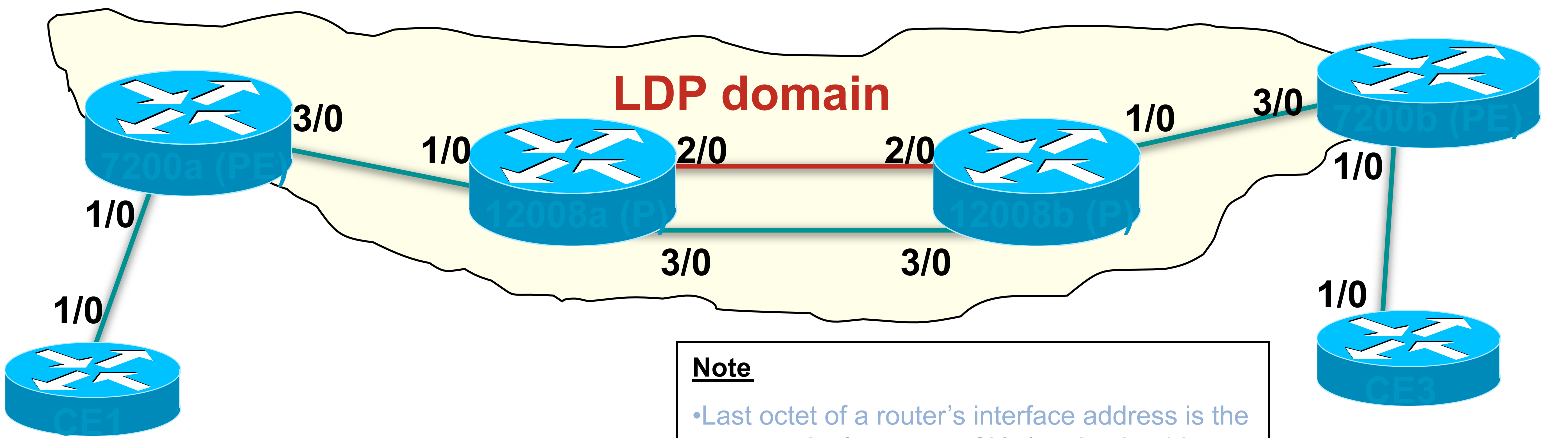
LDP Configuration

Configuring and monitoring LDP

Agenda

- Configuration
- Verifying Your Configuration
- Monitoring LDP

Network Topology



Note

•Last octet of a router's interface address is the same as the last octet of it's loopback address

•All pt-pt links are /24

	7200a	CE1	12008a	12008b	7200b	CE3
7200a	4.4.4.4 (loop)	10.0.20.0	10.0.3.0			
CE1	10.0.20.0	100.100.100.100(lo0)				
12008a	10.0.3.0		5.5.5.5 (lo0)	10.0.4.0 (2/0) 10.0.5.0 (3/0)		
12008b			10.0.5.0 (3/0) 10.0.4.0 (2/0)	11.11.11.11(lo0)	10.0.17.0	
7200b				10.0.17.0	12.12.12.12 (lo0)	10.0.22.0
CE3					10.0.22.0	30.30.30.30(lo0)

Configuring LDP

Global

ip cef <distributed>

mpls label protocol <ldp | tdp | both>

tag-switching tdp router-id Loopback0

mpls ldp explicit-null (optional)

no mpls ip propagate-ttl (optional)

Interface

mpls ip **or** **tag-switching ip** (enables this interface for MPLS forwarding)

mpls label protocol ldp

(optional, if you want to run LDP on this interface only, while other interfaces don't run LDP or run another label protocol such as TDP)

Configuring Conditional Label Distribution

Router(config)#

```
tag-switching advertise-tags for net-acl [ to tdp-acl ]
```

- By default, labels for all destinations are announced to all LDP/TDP neighbors.
- This command enables you to selectively advertise some labels to some LDP/TDP neighbors.
- Conditional label advertisement only works over frame-mode interfaces.
- Parameters:
  - Net-ACL – the IP ACL that selects the destinations for which the labels will be generated.
  - TDP-ACL – the IP ACL that selects the TDP neighbors that will receive the labels.

Conditional Label Distribution Example

- The customer is already running IP infrastructure.
- MPLS is only needed to support MPLS/VPN services.
  - Labels should only be generated for loopback interfaces (BGP next-hops) of all routers.
  - All loopback interfaces are in one contiguous address block (192.168.254.0/24).

Conditional Label Distribution Router Configuration

- Enable conditional label advertisment

```
no tag-switching advertise-tags
!
! Configure conditional advertisments
!
tag-switching advertise-tags for 90 to 91
!
access-list 90 permit ip 192.168.254.0 0.0.0.255
access-list 91 permit ip any
```

Agenda

- Configuration
- Verifying Your Configuration
- Monitoring LDP

Verifying your configuration

```
hostname mpls-7200a
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
!
interface Ethernet3/0
 tag-switching ip
```

Agenda

- Configuration
- Verifying Your Configuration
- Monitoring LDP

Monitoring LDP

- show mpls interface <x> detail
- show mpls ldp discovery
- show mpls ldp neighbor
- show mpls ip/ldp binding <prefix> <prefix-length>
- show mpls forwarding-table <prefix> <prefix-length>
- sh ip cef <prefix>
- show mpls ldp parameters

Show mpls interface

```
mpls-7200a#sh mpls interface
Interface          IP          Tunnel   Operational
Ethernet3/0          Yes (ldp)    No        Yes
```

```
mpls-7200a#sh mpls interface ethernet3/0 detail
Interface Ethernet3/0:
        IP labeling enabled (ldp)
        ........<snip>.......
          Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
          MTU = 1500
```


Show mpls interface (contd..)

- “sh mpls interface [detail]”

  Lists whether MPLS is enabled and the application that enabled MPLS on the interface

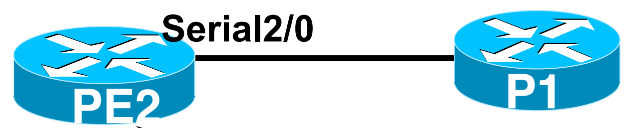
```
PE2#sh mpls interface
Interface              IP            Tunnel   Operational
Serial2/0              Yes (ldp)     No       Yes
PE2#
```

MPLS Enabled

LDP Enabled

```
PE2#sh mpls interface ser2/0 detail
Interface Serial2/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508
PE2#
```

MPLS MTU




```
!
interface Serial2/0
 description To P1 ser2/0
 ip address 10.13.2.6/30
 mpls label protocol ldp
 tag-switching ip
 tag-switching mtu 1508
!
```

Show mpls interface (contd..)

- This slide is to show that BGPipv4+label (or MP-eBGP) is another application that can enable MPLS; *what's different here -*

```
RSP-PE-SOUTH-6#sh mpls int
Interface              IP            Tunnel   Operational
Fddi1/0/0              Yes (ldp)     No       Yes
ATM1/1/0.108           No            No       Yes
RSP-PE-SOUTH-6#
```

MPLS is Operational.

LDP not enabled

```
RSP-PE-SOUTH-6#sh mpls int ATM1/1/0.108 detail
Interface ATM1/1/0.108:
        IP labeling not enabled
        LSP Tunnel labeling not enabled
        BGP tagging enabled
        Tagging operational
        Optimum Switching Vectors:
          IP to MPLS Feature Vector
          MPLS Feature Vector
        Fast Switching Vectors:
          IP to MPLS Fast Feature Switching Vector
          MPLS Feature Vector
        MTU = 4470
RSP-PE-SOUTH-6#
```

LDP not enabled

BGP+Label Enabled

MPLS MTU

LDP discovery/adjacency: commands and debugs

- `show mpls ldp discovery`
- `debug mpls ldp transport`
- `debug mpls ldp session io`

LDP discovery

Interface eth3/0 configured with LDP

```
mpls-7200a#sh mpls ldp discovery
 Local LDP Identifier:
    4.4.4.4:0
    Discovery Sources:
    Interfaces:
        Ethernet3/0 (ldp): xmit/recv
            LDP Id: 5.5.5.5:0
```

My LDP id

we are transmitting & receiving LDP messages

Neighbor's LDP id

"debug mpls ldp transport events"

- Should give information regarding whether the HELLOS are advertised/received

LDP adjacency debugs

LDP discovery, connection setup and shutdown events

```
mpls-7200a#debug mpls ldp transport events
debugging for LDP discovery and connection setup / shutdown events
2d11h: ldp: Send ldp hello; Ethernet3/0, src/dst 10.0.3.4/224.0.0.2, inst_id 0
2d11h: ldp: Rcvd ldp hello; Ethernet3/0, from 10.0.3.5 (5.5.5.5:0), intf_id 0, opt 0xC
shutting neighbor
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold time expired
2d11h: ldp:Discovery hold timer expired for adj 0x17D45A0, 5.5.5.5:0,will close conn
2d11h: ldp: Discovery hold timer expired for adj 0x17D45A0; 5.5.5.5:0
2d11h: ldp:     adj_addr/adj_xport_addr: 10.0.3.5/5.5.5.5
2d11h: ldp: LDP ptcl SM; close xport request for adj 0x0
2d11h: ldp: Close LDP transport conn for adj 0x17D45A0
2d11h: ldp: Closing ldp conn 4.4.4.4:646 <-> 5.5.5.5:11012, adj 0x17D45A0
2d11h: ldp: Adj 0x17D45A0; state set to closed
2d11h: ldp: Send ldp hello; Ethernet3/0, src/dst 10.0.3.4/224.0.0.2, inst_id 0
```

LDP session i/o debug

LDP session I/O, excluding periodic Keep Alives

```
mpls-7200a#debug mpls ldp session io <all>
```

bringing neighbor down

```
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold
time expired
2d11h: ldp: Sent notif msg to 5.5.5.5:0 (pp 0x17A0870)
.............
```

bringing neighbor up

```
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Up, new
adjacency
2d11h: ldp: Rcvd init msg from 5.5.5.5 (pp 0x0)
2d11h: ldp: Sent init msg to 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Sent keepalive msg to 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Rcvd keepalive msg from 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Sent address msg to 5.5.5.5:0 (pp 0x186CB38)
2d11h: ldp: Sent label mapping msg to 5.5.5.5:0 (pp 0x186CB38)
...............
```

LDP neighbor

```
mpls-7200a#sh mpls ldp neighbor
    Peer LDP Ident: 5.5.5.5:0; Local LDP Ident 4.4.4.4:0
       TCP connection: 5.5.5.5.11000 - 4.4.4.4.646
       State: Oper; Msgs sent/rcvd: 268/264; Downstream Up time: 03:41:45
       LDP discovery sources:
          Ethernet3/0, Src IP addr: 10.0.3.5
       Addresses bound to peer LDP Ident:
          10.0.3.5        10.0.4.5        10.0.5.5        5.5.5.5
```

LDP neighbor (contd..)

- LDP session is a TCP session (port = 646)
- Multiple links between two routers still mean single LDP session.

```
PE1#sh mpls ldp neighbor
    Peer LDP Ident: 10.13.1.101:0; Local LDP Ident 10.13.1.61:0
        TCP connection: 10.13.1.101.11031 - 10.13.1.61.646
        State: Oper; Msgs sent/rcvd: 58/60; Downstream
        Up time: 00:39:27
        LDP discovery sources:
          Ethernet0/0, Src IP addr: 10.13.1.5
          Ethernet1/0, Src IP addr: 10.13.1.9
        Addresses bound to peer LDP Ident:
          10.13.1.9       10.13.1.5       10.13.2.5     10.13.1.101
PE1#
```

LDP_ID

Unsolicited Label Distribution*

Interfaces on which peer is discovered

Peer's Connected int

```
PE1#sh tcp brief| i 646
43ABB020  10.13.1.101.11031        10.13.1.61.646         ESTAB
PE1#
```


LDP binding commands

- "sh mpls ip binding detail"

  Lists all prefixes with labels & LDP neighbors

- "sh mpls ip binding <*prefix*> <mask> det"

  Lists ACLs (if any), *prefix* bindings, and LDP neighbors. <u>Notice "Advertised to:" field</u>.

- "sh mpls ip binding advertisement-acls"

  Lists LDP filter, if there is any, on the first line. Prefixes followed by "Advert acl(s):" are advertised via LDP, others are not.

LIB information

```
mpls-7200a#sh mpls ip binding 12.12.12.12 32
  12.12.12.12/32
        in label:     21
        out label:    19        lsr: 5.5.5.5:0        in use

mpls-7200a#sh mpls ldp binding 12.12.12.12 32
  tib entry: 12.12.12.12/32, rev 48
        local binding:  tag: 21
        remote binding: tsr: 5.5.5.5:0, tag: 19
```


LDP binding related debugs

```
mpls-7200a#debug mpls ldp bindings

shutting neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold
time expired

2d11h: tagcon: tibent(5.5.5.5/32): label imp-null from 5.5.5.5:0 removed

2d11h: tagcon: route_tag_change for: 5.5.5.5/32

        inlabel 16, outlabel withdrwn, nexthop lsr 5.5.5.5:0, reason response to
find_route_tags

2d11h: tagcon: Deassign peer id; 5.5.5.5:0: id 0

2d11h: tagcon: tc_iprouting_table_change: 5.5.5.5/255.255.255.255, event 0x2

2d11h: tagcon: rib change: 5.5.5.5/255.255.255.255; event 0x2; ndb attrflags
0x1000000;

ndb->pdb_index/pdb->index 0x3/0x3

2d11h: tagcon: rib change: 5.5.5.5/255.255.255.255; event 0x2; ndb attrflags
0x1000000;

ndb->pdb_index/pdb->index 0x3/undef
```

LDP Advertisement related debugs

```
mpls-7200a#debug mpls ldp advertisements

shutting neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down,
hold time expired

2d11h: tagcon: Deassign peer id; 5.5.5.5:0: id 0

activating neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Up,
new adjacency

2d11h: tagcon: Assign peer id; 5.5.5.5:0: id 0

2d11h: tagcon: peer 5.5.5.5:0 (pp 0x17AFAE0): advertise 4.4.4.4

2d11h: tagcon: Advertise labels: Clear LDP_CTX_TCB_FLAGS_ENULL_RECFG

2d11h: tagcon: peer 5.5.5.5:0 (pp 0x17AFAE0): advertise 4.4.4.4/32, label 3
(imp-null) (#32)
```

LFIB information

```
show mpls forwarding-table <prefix>
  <prefix-length>

sh ip cef <prefix> internal
```

Looking at LFIB

Looking at LFIB on 12008a

```
mpls-12008a#sh mpls forwarding 12.12.12.12 32 detail
Local   Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag     tag or VC   or Tunnel Id      switched   interface
19      19          12.12.12.12/32    498        Et2/0       10.0.4.11
         MAC/Encaps=14/18, MTU=1500, Tag Stack{19}
         AABBCC000502AABBCC0004028847 00013000
         No output feature configured
 Per-destination load-sharing, slots: 0 2 4 6 8 10 12 14
        19          12.12.12.12/32    498        Et3/0       10.0.5.11
         MAC/Encaps=14/18, MTU=1500, Tag Stack{19}
         AABBCC000503AABBCC0004038847 00013000
         No output feature configured
 Per-destination load-sharing, slots: 1 3 5 7 9 11 13 15
```

Ethertype=
8847
Label Value in
MPLS shim=
13 Hex=19 dec

Destination MAC=
AABBCC000502
Source MAC=
AABBCC000402

CEF command

```
mpls-12008a#sh ip cef 12.12.12.12 internal

12.12.12.12/32, version 24, epoch 0, per-
destination sharing

0 packets, 0 bytes

  tag information set, local tag: 19

  via 10.0.4.11, Ethernet2/0, 0
dependencies

    traffic share 1

    next hop 10.0.4.11, Ethernet2/0

    valid adjacency

    tag rewrite with Et2/0, 10.0.4.11, tags
imposed: {19}

  via 10.0.5.11, Ethernet3/0, 0
dependencies

    traffic share 1

    next hop 10.0.5.11, Ethernet3/0

    valid adjacency

    tag rewrite with Et3/0, 10.0.5.11, tags
imposed: {19}
```

0 packets, 0 bytes switched through the prefix

........................(contd..)

tmstats: external 0 packets, 0 bytes

internal 0 packets, 0 bytes

Load distribution: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 (refcount 1)

Hash	OK	Interface	Address	Packets	Tags imposed
1	Y	Ethernet2/0	10.0.4.11	0	{19}
2	Y	Ethernet3/0	10.0.5.11	0	{19}
3	Y	Ethernet2/0	10.0.4.11	0	{19}
4	Y	Ethernet3/0	10.0.5.11	0	{19}
5	Y	Ethernet2/0	10.0.4.11	0	{19}
6	Y	Ethernet3/0	10.0.5.11	0	{19}
7	Y	Ethernet2/0	10.0.4.11	0	{19}
8	Y	Ethernet3/0	10.0.5.11	0	{19}
9	Y	Ethernet2/0	10.0.4.11	0	{19}
10	Y	Ethernet3/0	10.0.5.11	0	{19}
11	Y	Ethernet2/0	10.0.4.11	0	{19}
12	Y	Ethernet3/0	10.0.5.11	0	{19}
13	Y	Ethernet2/0	10.0.4.11	0	{19}
14	Y	Ethernet3/0	10.0.5.11	0	{19}
15	Y	Ethernet2/0	10.0.4.11	0	{19}
16	Y	Ethernet3/0	10.0.5.11	0	{19}

Monitoring LDP: LDP parameters

```
mpls-7200a#sh mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 100000
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 180 sec; interval: 5 sec
Downstream on Demand max hop count: 255
TDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```


Forwarding traffic down the LSP

```
mpls-7200a#sh mpls forwarding-table 12.12.12.12
Local   Outgoing    Prefix             Bytes tag  Outgoing    Next Hop
tag     tag or VC   or Tunnel Id       switched   interface
21      19          12.12.12.12/32     0          Et3/0       10.0.3.5
Note: Bytes tag switched this will increment if packets are being tag
switched using this entry
```

```
mpls-12008a#sh mpls forwarding-table label 19
Local   Outgoing    Prefix             Bytes tag  Outgoing    Next Hop
tag     tag or VC   or Tunnel Id       switched   interface
19      19          12.12.12.12/32     498        Et2/0       10.0.4.11
        19          12.12.12.12/32     1176       Et3/0       10.0.5.11
```

```
mpls-12008b#sh mpls forwarding-table labels 19
Local   Outgoing    Prefix             Bytes tag  Outgoing    Next Hop
tag     tag or VC   or Tunnel Id       switched   interface
19      Pop tag     12.12.12.12/32     4176       Et1/0       10.0.17.12
```

LDP binding and advertisemsnt debugs

- Be Careful on the production routers
- “debug mpls ldp advertisements”

  Useful to see label bindings that are advertised
- “debug mpls ldp binding”

  Useful to see label bindings that are received
- “debug mpls ldp message sent|received”

  Useful for the protocol understanding purposes

LDP Debugging

Troubleshooting LDP

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

Control Plane – Troubleshooting Tips

- Check for same label protocol to be configured on both sides of the interface

  “Sh mpls ldp discovery | inc ldp|tdp”

- Check whether correct local LSR_ID is used on both LSRs (sh mpls ldp disc)

  “sh mpls ldp discovery” – 2nd line in output

- Don’t assume that the neighbor discovery means everything is good.

Control Plane – Troubleshooting Tips

- Check IP reachability to remote LSR_ID on both LSRs

  “ping <lsr_id>”

- Check for ACL or ICMP unreachable blockages
- Untagged outgoing label for /32 routes i.e. PEs’ loopbacks is almost always alarming.
- Check the label binding for a prefix on both LSRs

  “sh mpls ldp bind <prefix> <mask>”

Control Plane – Troubleshooting Tips

- Make sure the LDP filtering (if configured) is correctly setup via ACL
  - “sh mpls ip bind advertisement-acl | inc Prefix”
- Good practice is to configure the Loopback0 as the router-ID for LDP

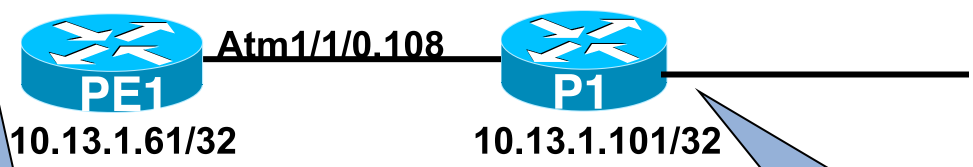
  “mpls ldp router-id loopback0 force”

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Control Plane – Protocol mismatch

Prob#1 – session establishment (Protocol mismatch)




```
PE1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.61:0
    Discovery Sources:
    Interfaces:
     ATM1/1/0.108 (tdp): xmit
PE1#
```

Why no recv?

TDP

```
P1#sh mpls ldp discovery
Local LDP Identifier:
    10.13.1.101:0
LDP Discovery Sources:
    Interfaces:
       ATM2/0.108(ldp): xmit
P1#
```

Why no recv?

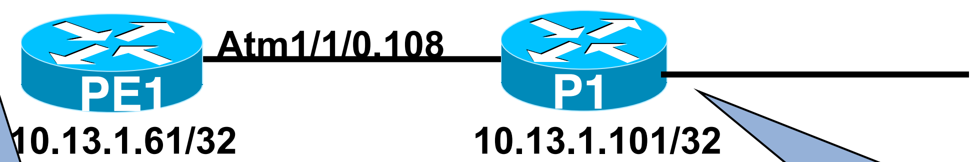
LDP

TIP – Check for the protocol mismatch and fix it.

```
PE1(config)#int atm1/1/0.108
PE1(config-if)#mpls label protocol ldp
```

MPLS Control Plane – No route

Prob#2 – session establishment (No route to peer)




```
PE1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.61:0
    Discovery Sources:
    Interfaces:
        ATM1/1/0.108 (ldp): xmit/recv
            LDP Id: 10.13.1.101:0
PE1#
PE1#sh mpls ldp neigh 10.13.1.101

PE1#
```

Looks Good ☺

But No relationship ☹

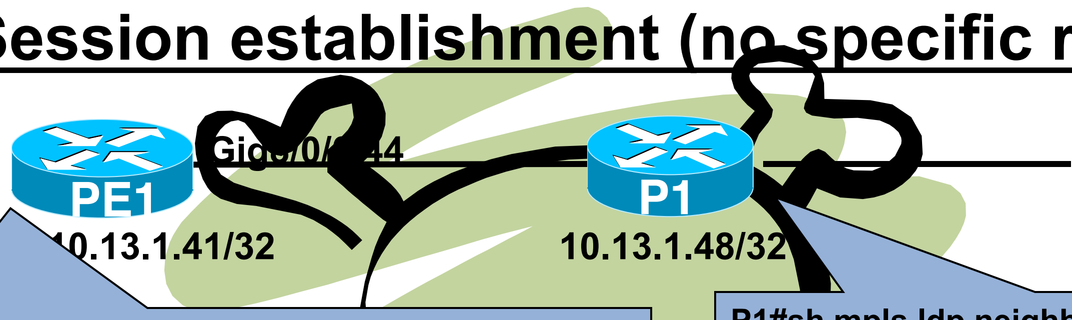
```
P1#sh mpls ldp discovery
Local LDP Identifier:
    10.13.1.101:0
LDP Discovery Sources:
    Interfaces:
        ATM2/0.108: xmit/recv
            LDP Id: 10.13.1.61:0; no route
P1#
P1#sh ip route 10.13.1.61
% Network not in table
P1#
```

This is the problem

TIP – Check for IP reachability to LDP_ID. Fix it by letting PE1 advertise 10.13.1.61/32 via IGP to P1.

MPLS Control Plane – No Specific route

Prob#3 - Session establishment (no specific route)



oops

Ok.

Ok.

```
PE1#sh mpls ldp neighbor 10.13.1.48

PE1#
PE1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.41:0
        Gi8/0/0.44 (ldp): xmit/recv
            LDP Id: 10.13.1.48:0
PE1#
PE1#sh ip route 10.13.1.48
Routing entry for 10.13.1.48/32
  Known via "isis", distance 115, metric 10, type level-1
  Redistributing via isis
  Last update from 10.13.4.9 on Gig8/0/0.44, 20:22:14 ago
  Routing Descriptor Blocks:
  * 10.13.4.9, from 10.13.1.48, via Gigt8/0/0.44
      Route metric is 10, traffic share count is 1
PE1#
```

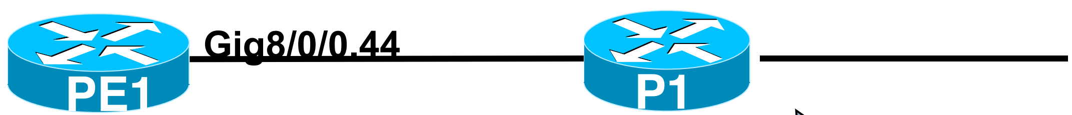
```
P1#sh mpls ldp neighbor 10.13.1.41

P1#
P1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.48:0
        Gi3/0/0.44 (ldp): xmit/recv
            LDP Id: 10.13.1.41:0
P1#
P1#sh ip route 10.13.1.41
Routing entry for 10.13.0.0/22
  Known via "bgp 30000", distance 200, metric 0
  Tag 1, type internal
  Last update from 10.13.1.251 20:10:38 ago
  Routing Descriptor Blocks:
  * 10.13.1.251, from 10.13.1.40, 20:10:38 ago
      Route metric is 0, traffic share count is 1
      AS Hops 5
P1#
```

Ouch

P1 doesn't have a specific route to PE1.

MPLS Control Plane – No Specific route (contd..)

Prob#3 - Session establishment (Contd)



10.13.1.41/32

10.13.1.48/32

PE1#ping 10.13.1.48
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.13.1.48, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
PE1#

P1#ping 10.13.1.41
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.13.1.41, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
P1#

Eeeekks !! It is an IP problem.

TIP – Check for IP connectivity first. Unless Layer3 is up, Layer4 (TCP session for LDP) won't come up.

MPLS Control Plane – Untagged outbound label

Prob#4 - "Untagged" problem

```
PE1#sh tag for 11.10.128.138
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Untagged    11.10.128.138/32  0          PO4/1/0    point2point
PE1#

PE1#sh mpls ldp  bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 14
        local binding:  tag: 16
PE1#
```

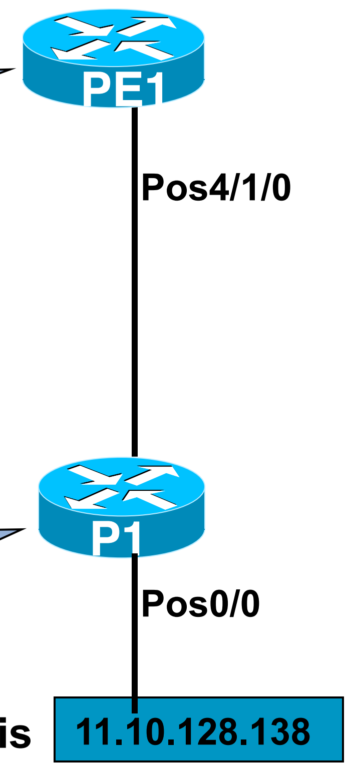
Untagged ?

No remote binding. Huh

```
P1#sh mpls ldp bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 4849(no route)
        local binding:  tag: 630
        remote binding: tsr: 10.13.1.54:0,   tag: 16
        remote binding: tsr: 11.10.65.12:0, tag: 48

P1#sh ip route 11.10.128.138
Routing entry for 11.10.0.0/16
  Known via "isis", distance 115, metric 44, type level-2
  Redistributing via isis
  Last update from 11.10.65.13 on POS0/0, 1d00h ago
  Routing Descriptor Blocks:
  * 11.10.65.13, from 11.10.128.31, via POS0/0
      Route metric is 44, traffic share count is 1
P1#
```

No route

But there is a RIB entry. Let's check FIB entry -



MPLS Control Plane – Untagged outbound label (contd..)

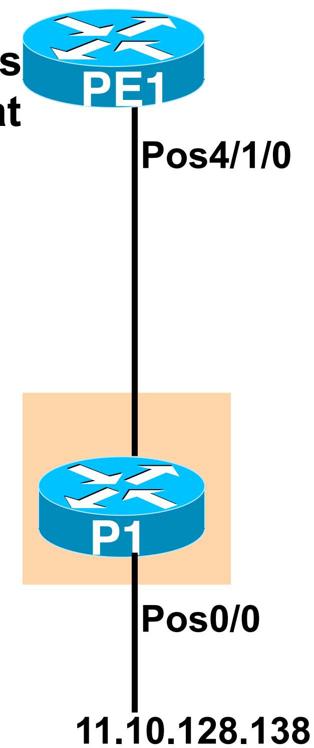
Prob#4 - "Untagged" problem (contd)

```
P1#sh ip cef 11.10.128.138
11.10.0.0/16, version 142, cached adjacency to POS0/0
0 packets, 0 bytes
  tag information set
    local tag: 307
    fast tag rewrite with PO0/0, point2point, tags imposed {48}
  via 11.10.65.13, POS0/0, 0 dependencies
    next hop 11.10.65.13, POS0/0
    unresolved        <<<<<<<<<<<<<<<<<<<<<<<<<<<
    valid cached adjacency
    tag rewrite with PO0/0, point2point, tags imposed {48}
P1#
```

FIB's local label is different from that of LIB

Unresolved ?

```
P1#clear ip route 11.10.128.138
P1#sh mpls ldp bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 4849
        local binding:  tag: 307
        remote binding: tsr: 10.13.1.54:0, tag: 16
        remote binding: tsr: 11.10.65.20:0,tag:48
P1#
```



TIP – If local label for a prefix is not same in FIB and LIB, then issue "clear ip route <prefix>" to fix.

MPLS Control Plane – No LFIB entry

Prob#5 – LFIB entry disappears

- No LFIB entry
- This might occur if the RIB owner for an IPv4 routes changes from IGP to BGP
- LDP doesn't allocate labels for the BGP owned IPv4 routes
- Notice the absence of local binding in LIB for that route

MPLS Control Plane – No LFIB entry (contd..)

```
7206-PE-SOUTH-1#sh mpls ldp bind 4.4.0.0 24
  tib entry: 4.4.0.0/24, rev 152
        remote binding: tsr: 10.13.1.69:0, tag: 213
        remote binding: tsr: 10.13.1.68:0, tag: 212
7206-PE-SOUTH-1#
```

No Local Binding

```
7206-PE-SOUTH-1#sh ip route 4.4.0.0
Routing entry for 4.4.0.0/24
  Known via "bgp 30000", distance 200, metric 0
  Tag 1, type internal
  Redistributing via isis, ospf 1
  Last update from 10.13.1.251 5d17h ago
  Routing Descriptor Blocks:
  * 10.13.1.251, from 10.13.1.40, 5d17h ago
      Route metric is 0, traffic share count is 1
      AS Hops 5
      Route tag 1

7206-PE-SOUTH-1#
```

Because it is a BGP learned prefix

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

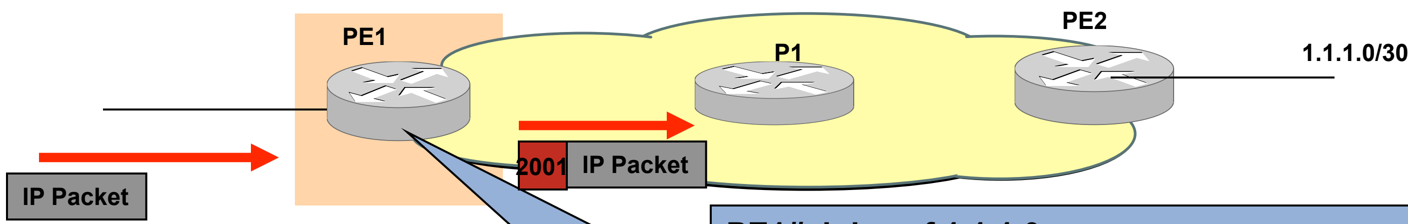
MPLS Forwarding Plane

- Three cases in the MPLS forwarding -
  1) Label Imposition - IP to MPLS conversion
  2) Label swapping - MPLS to MPLS
  3) Label disposition - MPLS to IP conversion

- So, depending upon the case, we need to check-
  1) FIB - For IP packets that get forwarded as MPLS
  2) LFIB - For MPLS packets that get fwded as MPLS
  3) LFIB - For MPLS packets that get fwded as IP

MPLS Forwarding Plane

Case 1: IP packets get forwarded as MPLS

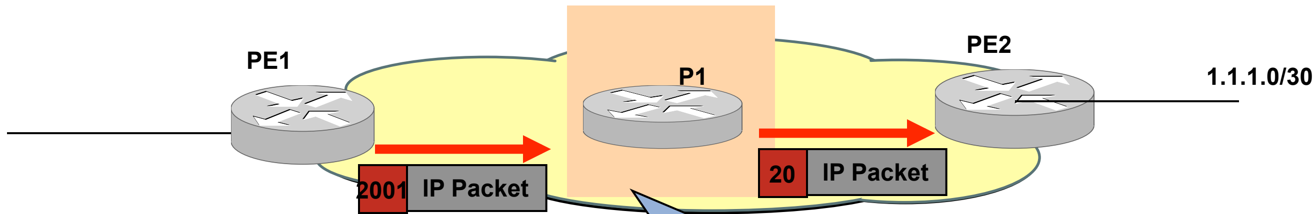



```
PE1#sh ip cef 1.1.1.0
1.1.1.0/30, version 25, epoch 0, cached adjacency 10.13.1.5
0 packets, 0 bytes
  tag information set
    local tag: 20
    fast tag rewrite with Et0/0, 10.13.1.5, tags imposed: {2001}
  via 10.13.1.5, Ethernet0/0, 0 dependencies
    next hop 10.13.1.5, Ethernet0/0
    valid cached adjacency
    tag rewrite with Et0/0, 10.13.1.5, tags imposed: {2001}
PE1#
```

- PE1 does a FIB lookup for the incoming IP packet
- It imposes the label (if there is one)
- For troubleshooting, look at the FIB (not LFIB)

MPLS Forwarding Plane

Case 2: MPLS packets get forwarded as MPLS



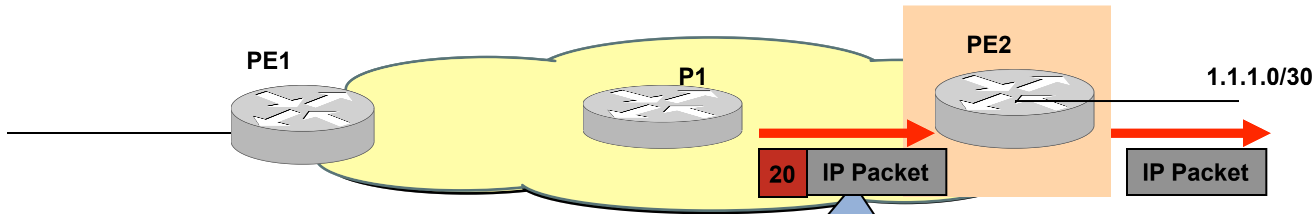
- P1 does the LFIB lookup for incoming MPLS packets
- P1 could swap (or dispose) the label
- For troubleshooting, look at the LFIB (not FIB)

```
P1#sh mpls for 1.1.1.0
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
2001   20      1.1.1.1.0/30     0         Se2/0      point2point
P1#
```

```
P1#sh mpls for 10.13.1.62
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
2001   Pop tag     10.13.1.62/32    0          Se2/0      point2point
P1#
```

MPLS Forwarding Plane

<u>Case 3: MPLS packets get forwarded as IP</u>




```
PE2#sh mpls for 1.1.1.0
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Untagged    1.1.1.1.0/30     0          Se2/0       point2point
PE2#
```

- Typically happen at the edge.
- Could also happen at the PHP router
- For troubleshooting, look at the LFIB (not FIB)

Agenda

- Control Plane
    - Troubleshooting Tips
    - Case Studies
- Forwarding Plane
    - Types of forwarding cases
    - Load sharing
    - MTU issues
    - Troubleshooting Tips
    - Case Studies

MPLS Forwarding Plane - Loadsharing

- Loadsharing (due to multiple paths to a prefix) in MPLS is no different from that of IP
- Hashing-algorithm is still the typical ‘FIB based’ i.e per-dest loadsharing by default **
- So the below “show command” is still relevant

  “Sh ip cef exact-route <source> <dest>” etc.
- But the dest must be known in the FIB table, otherwise the command won’t work.

  Won’t work on P routers for the VPN prefixes.

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
  - Load sharing
  - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Fwd Plane - Fragmentation

- After the Layer2 header is added to the IP packet, the resulting packet size shouldn't exceed the max packet size (IP MTU size) applicable . Otherwise, packet will be fragmented.
- MTU size needs to be tuned to avoid fragmentation in MPLS network
- MTU could be increased only for MPLS packets => MPLS MTU

Fragmentation
MTU Setting in MPLS

- Most of the interfaces (depending upon the hardware) support transmitting packets bigger than the “interface MTU” size
- “mpls mtu <bytes>” can be applied to an interface to change the MPLS MTU size on the interface
- MPLS MTU size is checked by the router

    while converting an IP packet into a labeled packet or transmitting a labeled packet

Fragmentation
MTU Setting in MPLS

- Remember that -
- ‘mpls mtu <bytes>” command has no effect on “interface or IP MTU” size.
- By default, MPLS MTU = interface MTU
- MPLS MTU setting doesn’t affect MTU handling for IP-to-IP packet switching

MTU Setting in MPLS
Configuring the MPLS MTU

```
RSP-PE-WEST-4(config)#int fa1/1/0
RSP-PE-WEST-4(config-if)#mpls mtu 1508
RSP-PE-WEST-4(config-if)#^Z
RSP-PE-WEST-4#
```

MTU Setting in MPLS
Before setting the MPLS MTU

- Interface MTU is 1500 bytes (no change):

```
RSP-PE-WEST-4#sh int fa1/1/0
  FastEthernet1/1/0 is up, line protocol is up
  Hardware is cyBus FastEthernet Interface, address is 0004.4e75.4828
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  ………………….
RSP-PE-WEST-4#
```

- MPLS MTU is 1508 bytes (changed):

```
RSP-PE-WEST-4#sh mpls interface fa1/1/0 detail
Interface FastEthernet1/1/0:
        IP tagging enabled
        TSP Tunnel tagging not enabled
       Tagging operational
        ……………………..
        MTU = 1508
RSP-PE-WEST-4#
```

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Fwd Plane – Troubleshooting Tips

- If PXF based platform, then check the PXF[1]
- On distributed platforms, check the FIB/LFIB entries on the LC
- On distributed platforms that have HW-based forwarding, check the FIB/LFIB on specific HW i.e. PSA (E2), Alpha(E3) on GSR etc.

  Sh ip psa-cef, sh tag psa-tag, sh ip alpha-cef etc

[1] **Not all PXF based platform support MPLS; they punt the MPLS packets to the CEF path.**

MPLS Fwd Plane – Troubleshooting Tips

- Label imposition is always done using FIB
- Label swapping and disposition is always done using LFIB
- Increase the MPLS MTU to accommodate the largest packet payload size
- Make sure that baby giant/jumbo is enabled on the Ethernet switches

MPLS Fwd Plane – Troubleshooting Tips

- Check that MPLS enabled interface has “TAG” adjacency via
  - “sh adjacency <interface>”
- Check that the LFIB’s outgoing label is same as the incoming label in neighbor’s LFIB
- Check the LSP via traceroute that shows labels used by each router in the path **

  “traceroute <prefix>”

MPLS Forwarding Plane – TAG adj

- Make sure that the interface has the “tag” adjacency along with “IP” adj, otherwise MPLS packets will not get switched on that interface

```
PE1#sh adjacency e0/0 de
Protocol Interface                 Address
TAG      Ethernet0/0               10.13.1.5(6)
                                   0 packets, 0 bytes
                                   AABBCC006500AABBCC0001008847
                                   mpls adj   never
                                   Epoch: 0
IP       Ethernet0/0               10.13.1.5(35)
                                   0 packets, 0 bytes
                                   AABBCC006500AABBCC0001000800
                                   ARP        03:46:13
                                   Epoch: 0
PE1#
```

L2 header for MPLS

L2 header for IP

MPLS Fwd Plane – Show commands

- "sh mpls forwarding"

  Shows all LFIB entries (vpn, non-vpn, TE etc.)

- "sh mpls forwarding <prefix>"

  LFIB lookup based on a prefix

- "sh mpls forwarding label <label>"

  LFIB lookup based on an incoming label

- "sh mpls forwarding <prefix> detail"

  Shows detailed info such as L2 encap etc

MPLS Fwd Plane – Debugs

- <u>Be Careful on the production routers</u>

- "Debug mpls lfib cef"

  Useful for seeing FIB and LFIB interaction when a label is missing for a prefix

- "debug mpls lfib struct"

  Shows changes in the LFIB structures when label is allocated/ deallocated

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
  - Load sharing
  - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Forwarding Plane - No entry in LFIB

Prob#1 - No entries in LFIB

```
P1#sh mpls forwarding-table 10.13.1.61
Tag switching is not operational.
CEF or tag switching has not been enabled.
Local   Outgoing     Prefix             Bytes tag  Outgoing    Next Hop
tag     tag or VC    or Tunnel Id       switched   interface
P1#
```

```
P1#sh mpls ip binding
 10.13.1.61/32
        out label:    imp-null  lsr: 10.13.1.61:0
        out label:    21        lsr: 10.13.1.62:0
  10.13.1.62/32
        out label:    imp-null  lsr: 10.13.1.62:0
        out label:    17        lsr: 10.13.1.61:0
  10.13.1.101/32
        out label:    19        lsr: 10.13.1.62:0
        out label:    18        lsr: 10.13.1.61:0
  10.13.2.4/30
        out label:    imp-null  lsr: 10.13.1.62:0
        out label:    19        lsr: 10.13.1.61:0
P1#
```

```
P1#sh ip cef
%CEF not running
Prefix              Next Hop               Interface
P1#
```

TIP – Enable CEF. It is must for MPLS.

MPLS Forwarding Plane- Out label is Untagged

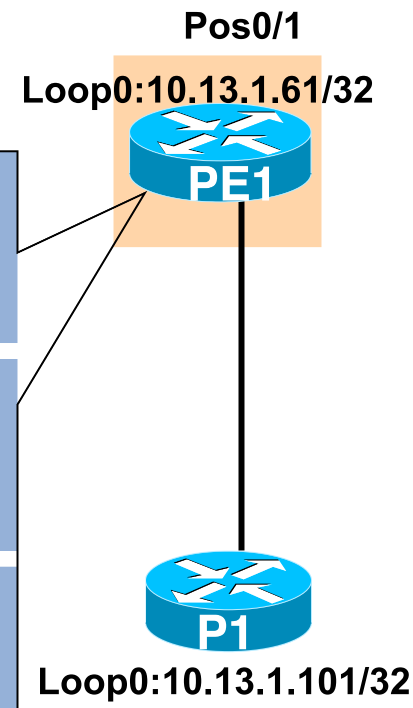
Prob#2 - “Untagged” problem

- LDP session is UP;LIB has correct binding;but LFIB has “Untagged” ☹

```
PE1#sh mpls for 10.13.1.101
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Untagged    10.13.1.101/32    0          PO0/1       point2point
PE1#
```

```
PE1#sh mpls ip bind 10.13.1.101 32
  10.13.1.101/32
        in label:     20
        out label:    imp-null  lsr: 10.13.1.101:0
PE1#
```

```
PE1#sh adjacency pos0/1
Protocol Interface                 Address
TAG      POS0/1                    point2point(7) (incomplete)  <<=====Oops
IP       POS0/1                    point2point(39)
PE1#
```



TAG ADJ for pos0/1 is incomplete. No good.

MPLS Forwarding Plane- Out label is Untagged (contd..)

(contd)

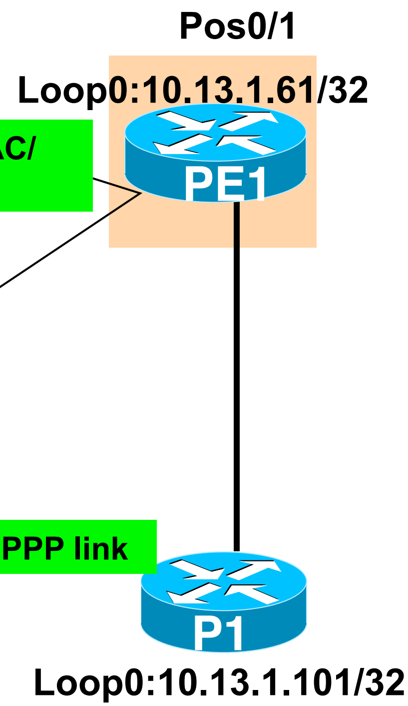
- **Adj is incomplete;check the interface.**

```
PE1#sh mpls for 10.13.1.101 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
12318  Untagged    10.13.1.101/32    0          PO0/1      point2point
            MAC/Encaps=0/0, MRU=4474, Tag Stack{}
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE1#
```

<<====== Another hint- why MAC/ Encap is 0/0?

```
PE1#sh int pos0/1
POS0/1 is up, line protocol is up
  Hardware is Packet over SONET
  Description: OC48 to Redback
  Internet address is 10.1.17.1/24
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive not set
  Scramble disabled
  LCP Open
  Listen: TAGCP, CDPCP
  Open: IPCP
  Last input 00:00:01, output 00:00:03, output hang never
....
PE1#
```

<<========== TAGCP should also be in the Open state on PPP link



MPLS Forwarding Plane- Out label is Untagged (contd..)

(contd)

```
PE1#deb mpls adj
PE1#deb mpls lfib enc
PE1#
01:43:19: LFIB: finish res:inc tag=12318,outg=Imp_null,next_hop=0.0.0.0,POS0/1
01:43:19: LFIB: get ip adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=7
01:43:19: LFIB: get tag adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=90 INCOMPLETE
01:43:19: TAG ADJ: check 0.0.0.0, POS0/1 (537CF240/537CEE80)
01:43:19: LFIB: get ip adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=7
01:43:19: LFIB: get tag adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=90
01:43:19: LFIB: encaps:zero encaps,enc=0,mac=0,tag_adj POS0/1,itag=12318
```

TIP – If the interface doesn't have "TAG" adj, then the label will not get installed in LFIB. Fix PPP in this case.

MPLS Forwarding Plane- Recursive rewrite

Prob#3 - "Recursive rewrite" problem

- If you ever see "Recursive rewrite via…" in the "sh ip cef .." output, then it might indicate a problem.

```
2611-CE-30#sh ip cef 10.13.1.74
10.13.1.74/32, version 43, epoch 0, cached adjacency 5.5.5.14
0 packets, 0 bytes
  tag information set
    local tag: BGP route head
    fast tag rewrite with
      Recursive rewrite via 217.60.217.2/32, tags imposed {23}
  via 217.60.217.2, 0 dependencies, recursive
    next hop 5.5.5.14, Ethernet0/0.2 via 217.60.217.2/32
    valid cached adjacency
    tag rewrite with
      Recursive rewrite via 217.60.217.2/32, tags imposed {23}
2611-CE-30#
```

Problem with the 217.60.217.2. Check its label binding in FIB/ LIB.

MPLS Forwarding Plane- Recursive rewrite (contd..)

<u>(contd)</u>

- “Recursive rewrite” usually means that

  (a) Either the label to the next-hop is not available

  (b) Or there is an internal problem with the CEF recursion resolution process

- (a) usually turns out to be a LDP problem, and should be fixed by investigating into LDP
- (b) could be fixed by “clear ip route <prefix>” or “clear ip bgp *”

MPLS Forwarding Plane- Recursive rewrite (contd..)

(contd)

- In order to troubleshoot (a), check the label availability for the next-hop (in LIB). If it is missing, then fix LDP.

```
2611-CE-30#sh mpls for 217.60.217.2
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
17     Untagged    217.60.217.2/32   0          Et0/0.2    5.5.5.14
2611-CE-30#
```

Untagged outgoing label

```
2611-CE-30#sh mpls ldp bind 217.60.217.2 32
  tib entry: 217.60.217.2/32, rev 14
        local binding:  tag: 17
2611-CE-30#
```

No remote label binding in LIB

```
2611-CE-30#sh mpls ldp dis
 Local LDP Identifier:
    217.60.217.3:0
    Discovery Sources:
    Interfaces:
        Ethernet0/0.2 (ldp): xmit
2611-CE-30#co
```

Because there is no LDP neighbor.

MPLS Forwarding Plane- Recursive rewrite (contd..)

(contd)

- LDP session needs to be established first.
- It is an LDP (control plane) problem.
- Troubleshoot for the LDP (as shown in the control plane section)

Conclusion

- Break down troubleshooting into systematic steps
- Look at things from a control plane and a forwarding plane perspective
- Do not panic

