

# MPLS LDP

# Agenda

- LDP Overview
- LDP Protocol Details
- LDP Configuration and Monitoring

# Label Distribution Protocol

## Overview

- MPLS nodes need to exchange label information with each other
  - Ingress PE node (Push operation)  
Needs to know what label to use for a given FEC to send packet to neighbor
  - Core P node (Swap operation)  
Needs to know what label to use for swap operation for incoming labeled packets
  - Egress PE node (Pop operation)  
Needs to tell upstream neighbor what label to use for specific FEC type LDP used for exchange of label (mapping) information
- Label Distribution Protocol (LDP)
  - Defined in RFC 3035 and RFC3036; updated by RFC5036
  - LDP is a superset of the Cisco-specific Tag Distribution Protocol
- Note that, in addition LDP, also other protocols are being used for label information exchange
  - Will be discussed later

# Label Distribution Protocol

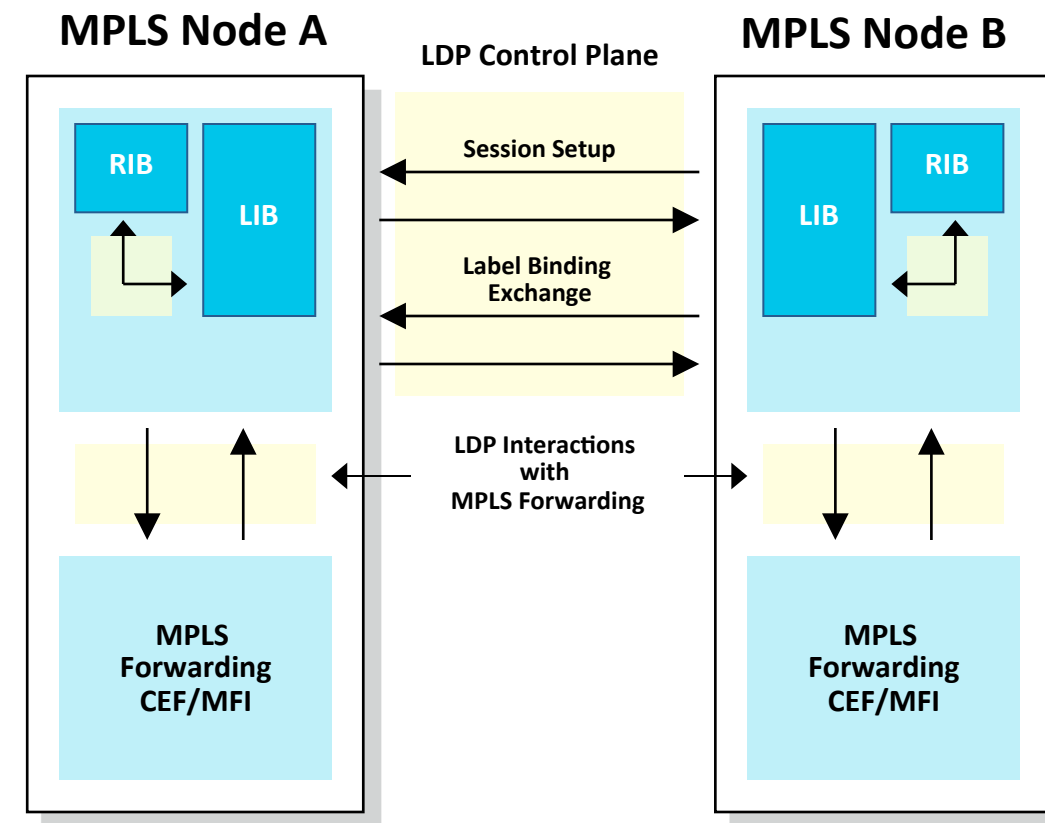
## Some More Details

- Assigns, distributes, and installs (in forwarding) labels for prefixes advertised by unicast routing protocols
  - OSPF, IS-IS, EIGRP, etc.
- Also used for Pseudowire/PW (VC) signaling
  - Used for L2VPN control plane signaling
- Uses UDP (port 646) for session discovery and TCP (port 646) for exchange of LDP messages
- LDP operations
  - LDP Peer Discovery
  - LDP Session Establishment
  - MPLS Label Allocation, Distribution, and Updating MPLS forwarding
- Information repositories used by LDP
  - LIB: Label Information Database (read/write)
  - RIB: Routing Information Database/routing table (read-only)

# Label Distribution Protocol

## Operations Details

- LDP startup
  - Local labels assigned to RIB prefixes and stored in LIB
  - Peer discovery and session setup
  - Exchange of MPLS label bindings
- Programming of MPLS forwarding
  - Based on LIB info
  - CEF/MFI updates



# Agenda LDP Protocol Details

- 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 (Updated by RFC 5036)

# TDP/LDP Transport

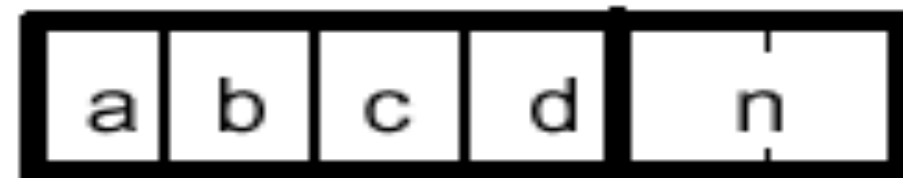
- Uses TCP for reliable transport
- Well-known TCP port
  - *LDP (port 646)*
  - ~~*TDP (port 711)*~~
- LSR with higher LDP router-id opens a connection to port 646 of other LSR
- Design Choice:
  - *One TDP/LDP session per TCP connection*



# LDP Identifier

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

**a.b.c.d:n**

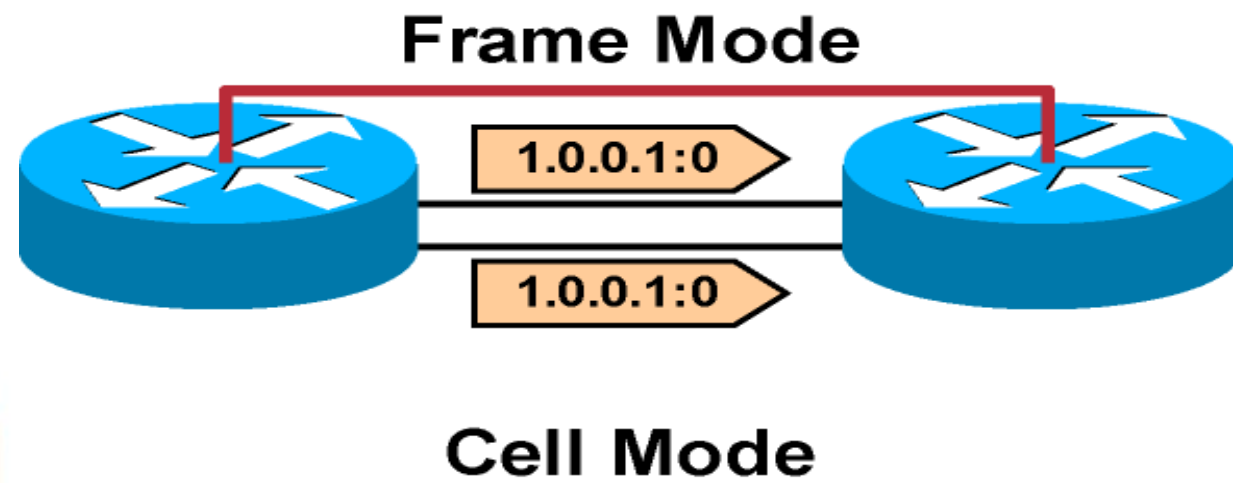


Router ID      Tag Space ID

# 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

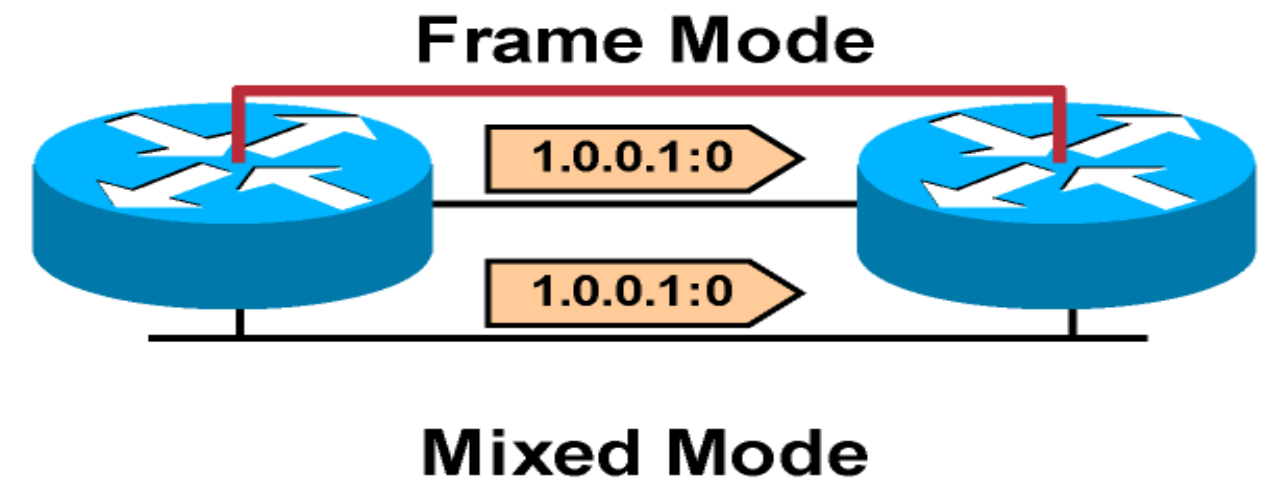
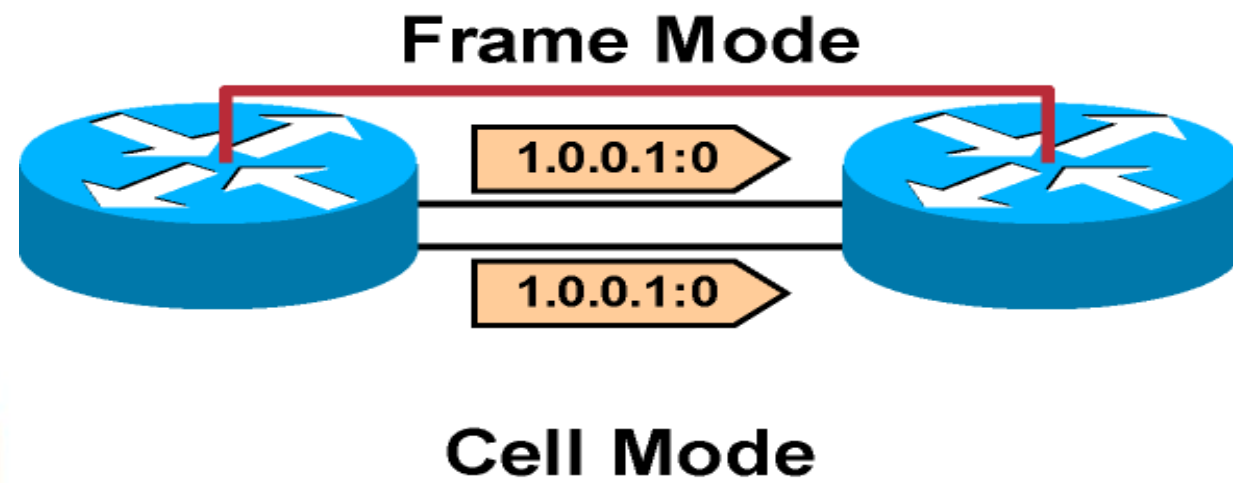


**Frame Mode**

**Mixed Mode**

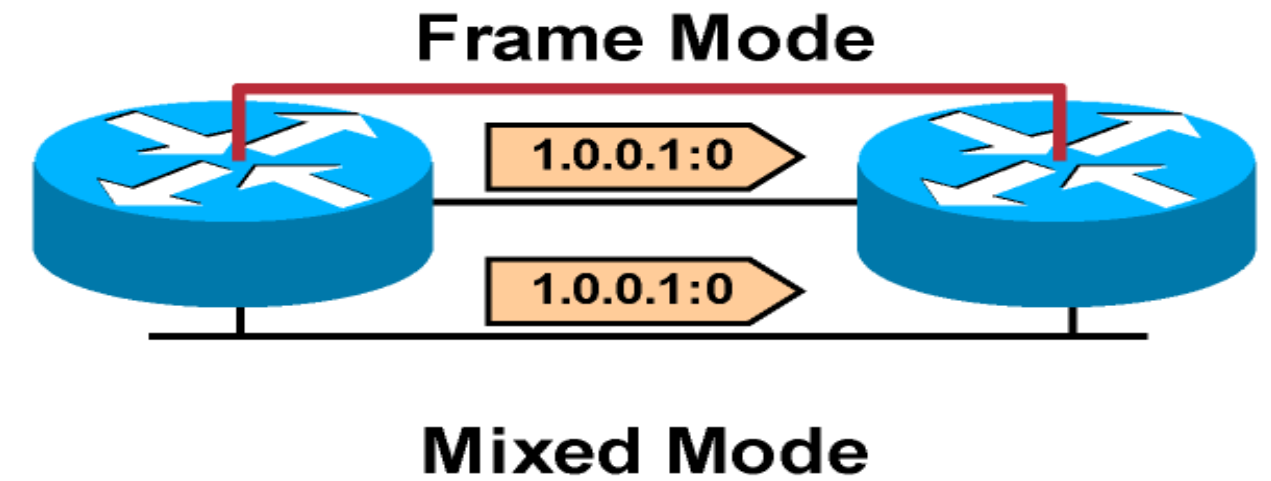
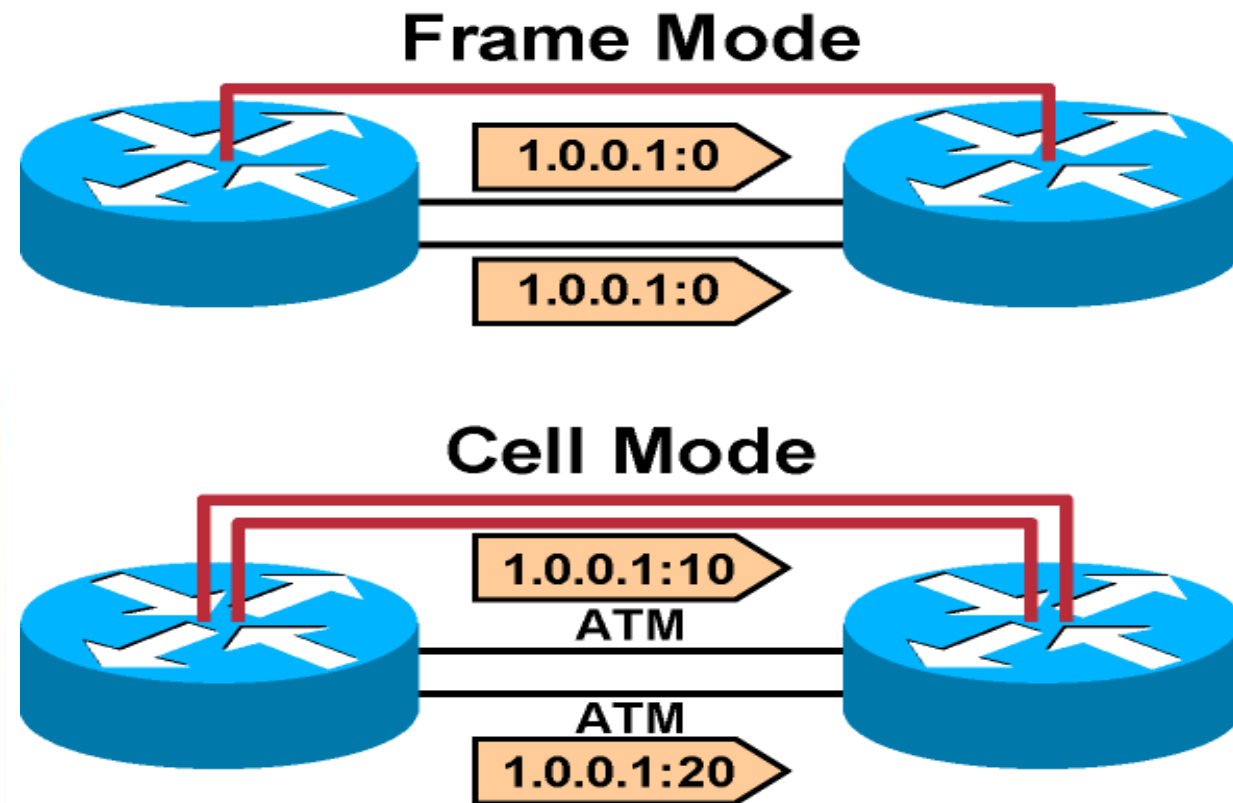
020G\_318

# Label Space and number of LDP sessions (Cont.)

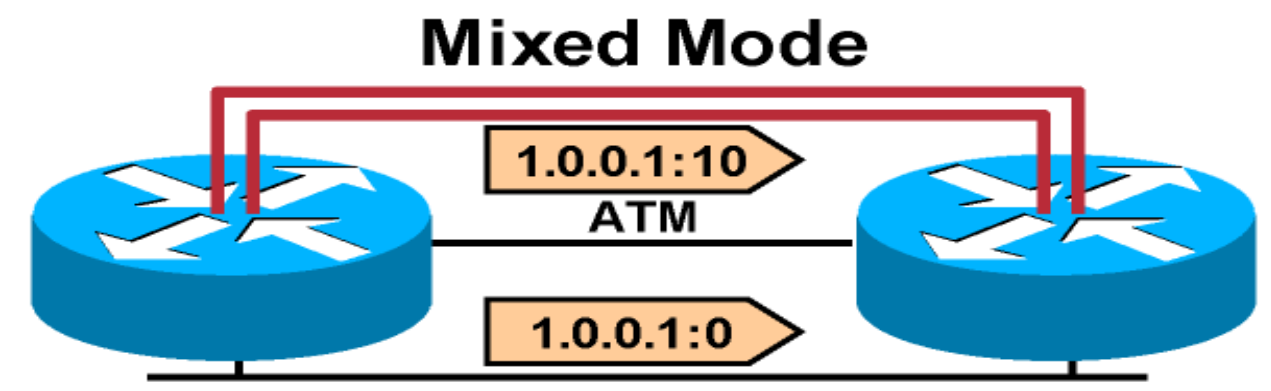
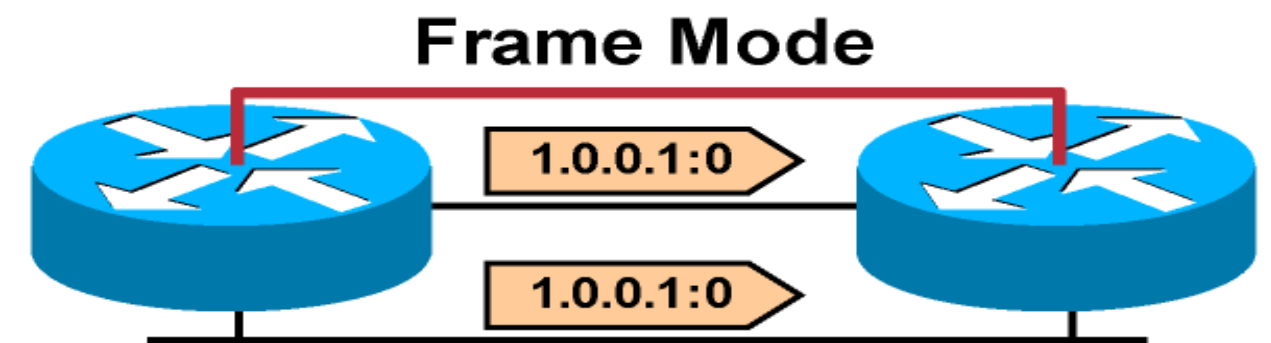
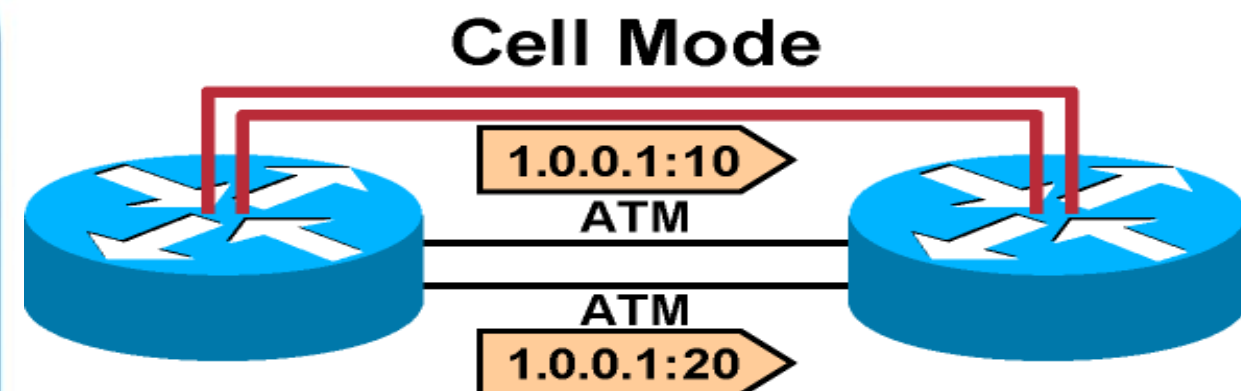
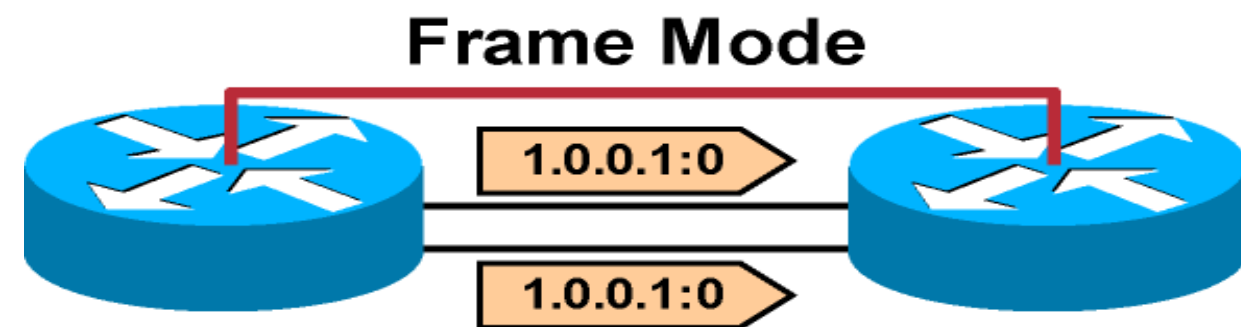


020G\_317

# Label Space and number of LDP sessions (Cont.)



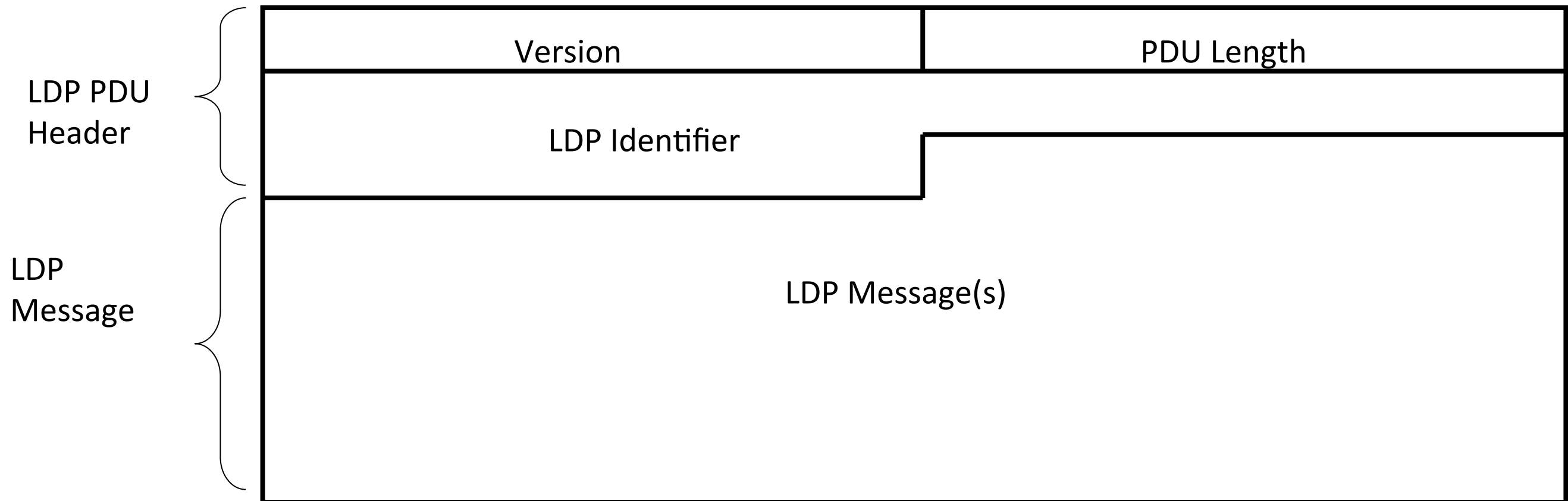
# Label Space and number of LDP sessions (Cont.)



- 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



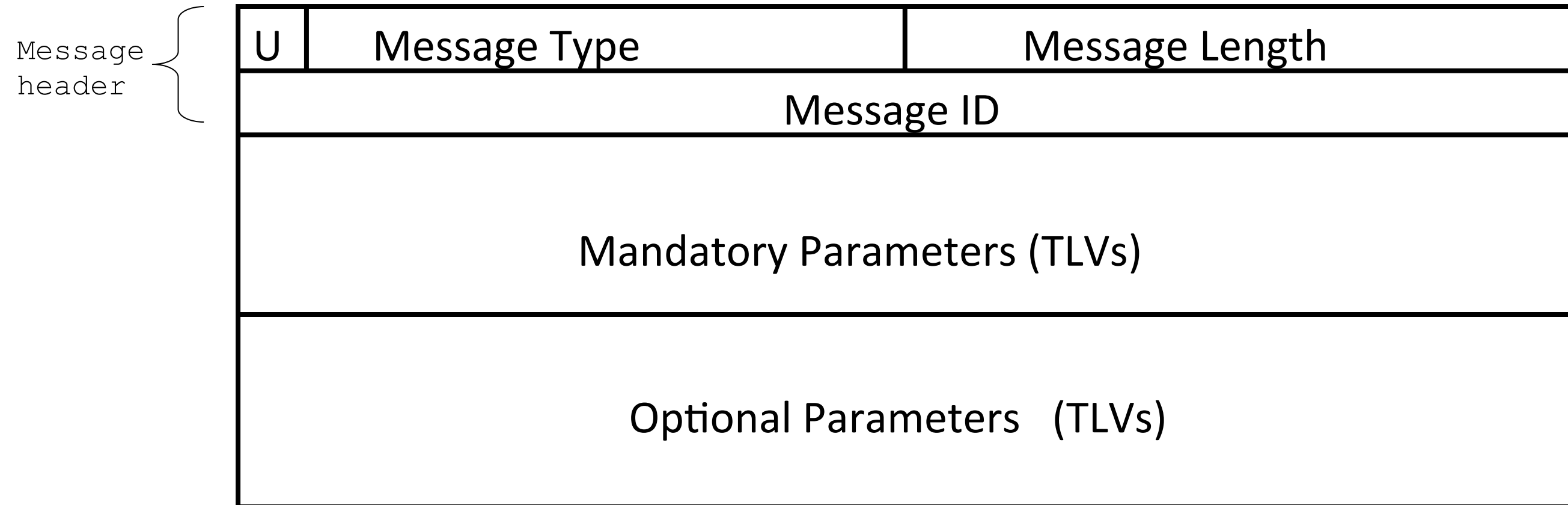
**Version** => LDP version. Current LDP version is 1

**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

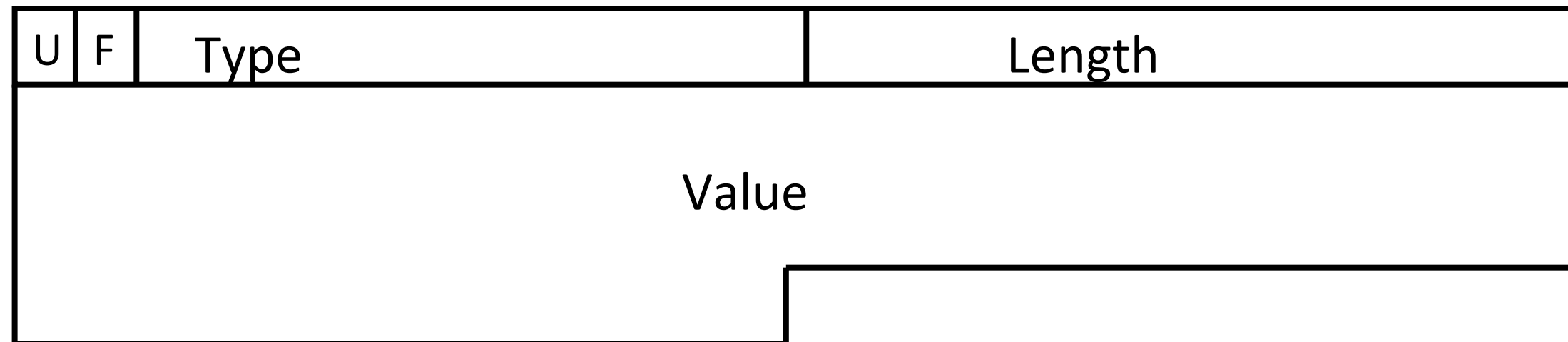
# LDP Message



**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

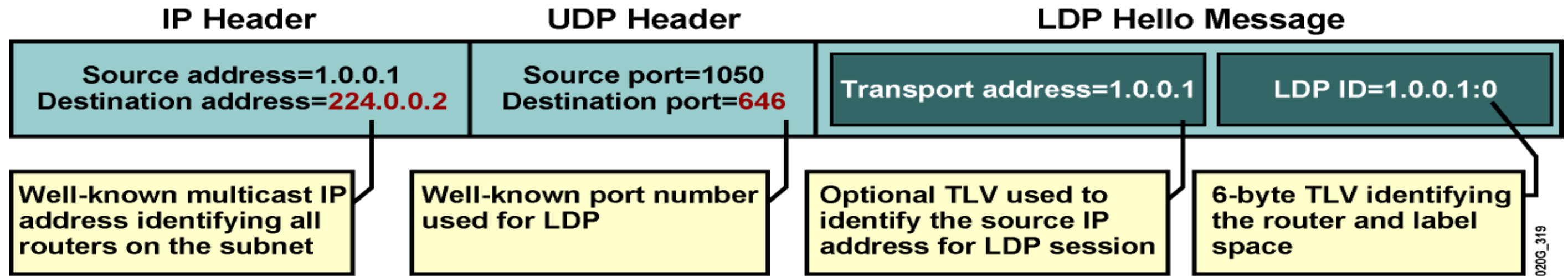
- **NOTIFICATION**

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).

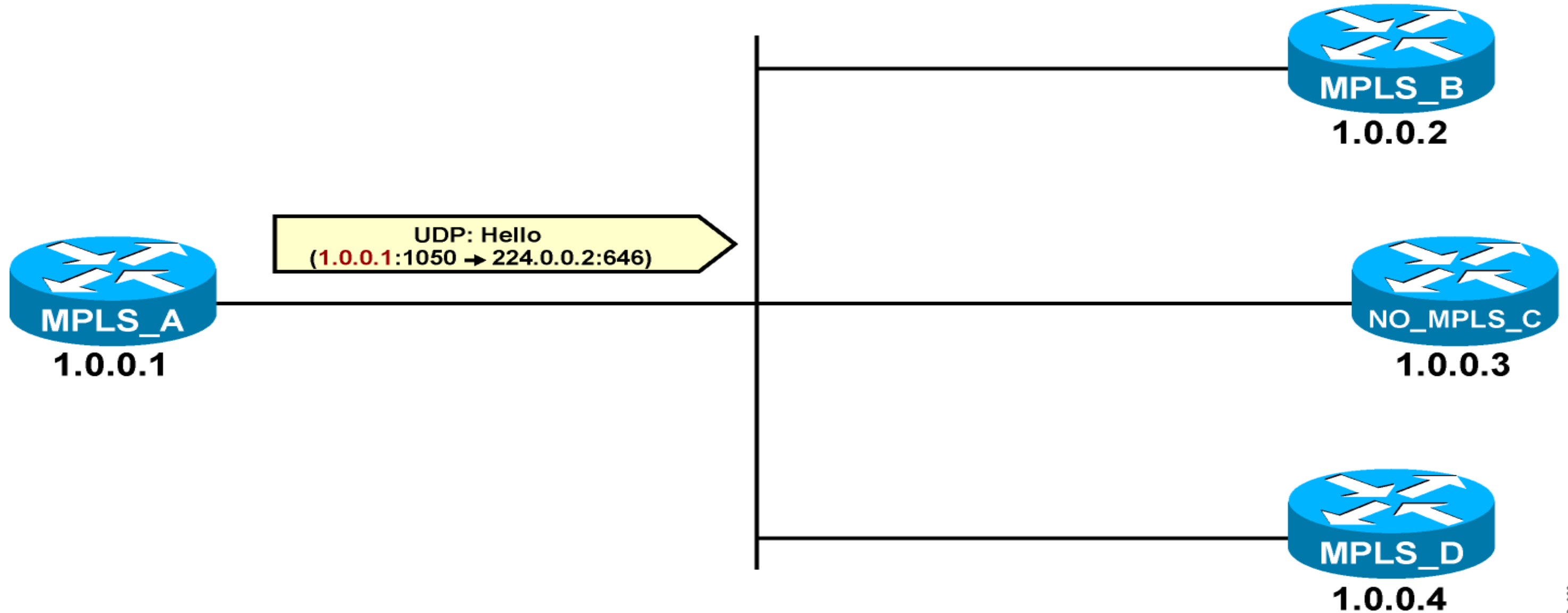
# LDP Hello Message



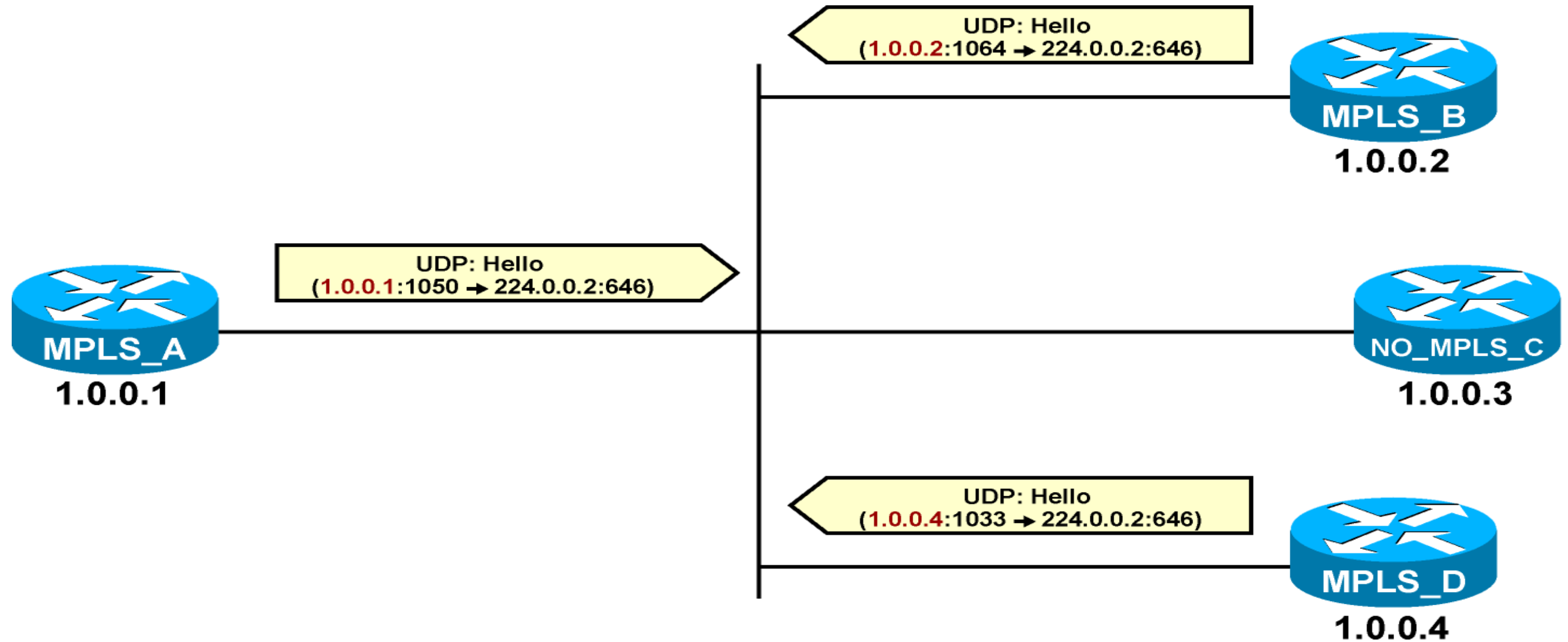
- Hello messages are targeted at all routers reachable through an interface.
- LDP uses well-known (UDP and TCP) port number 646.
- The source address used for an LDP session can be set by adding the transport address TLV to the hello message.
- A 6-byte LDP identifier (TLV) identifies the router (first four bytes) and label space (last two bytes).



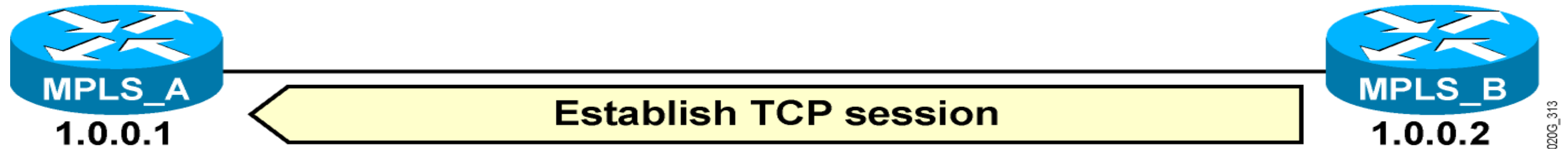
# LDP Neighbor Discovery



# LDP Neighbor Discovery

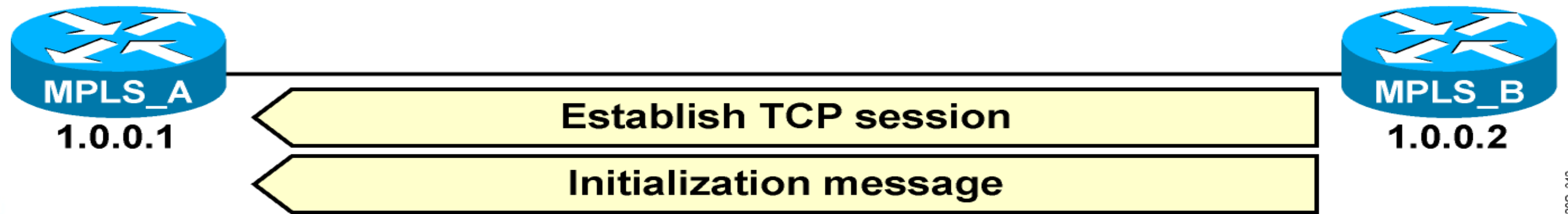


# LDP Session: Transport Connection



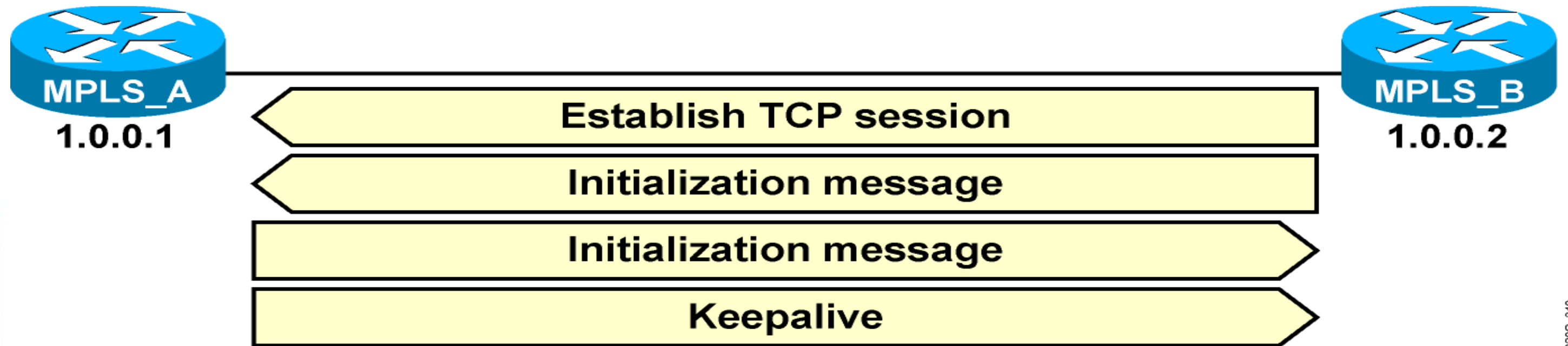
- Once LDP peers receive hellos, they establish a TCP connection
- Peer with higher LDP router-id is active LSR and the peer with lower LDP router-id is the passive LSR
- Active LSR tries to open a TCP connection to the well-known LDP port number 646 of the passive LSR, while the passive LSR waits for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

# LDP Session: Session Initialization



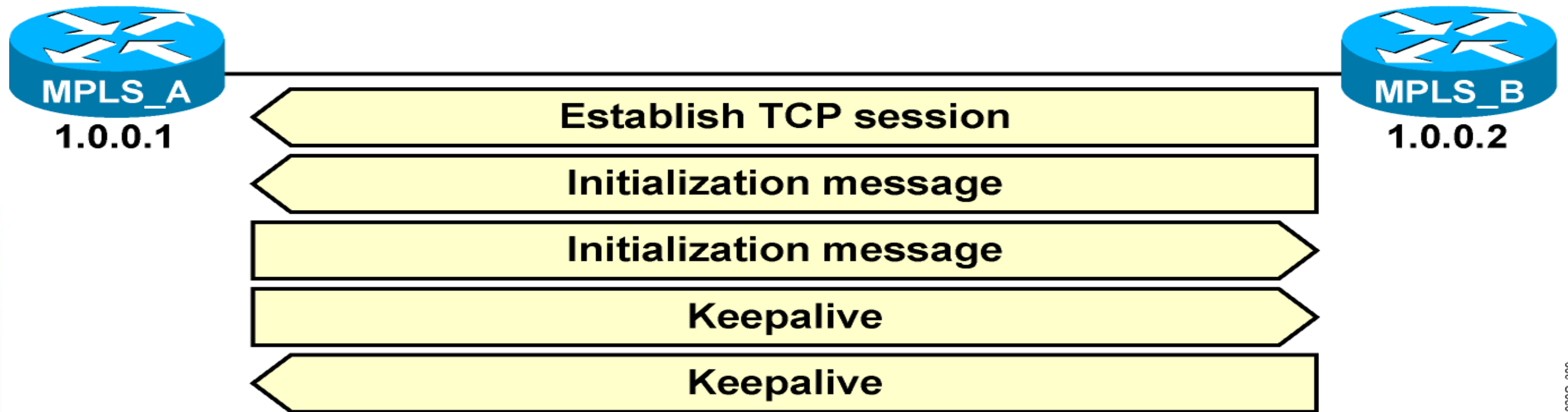
- Active LDP peer (1.0.0.2) sends Initialization message to passive LDP peer
- Initialization message contains important parameters:
  - Session keepalive time (default=180 sec)
  - Label distribution method: Downstream unsolicited
  - Max PDU length
  - Receiver's LDP Identifier
  - Whether Loop Detection is enabled
  - Some optional parameters

# LDP Session: Session Initialization (cont.)



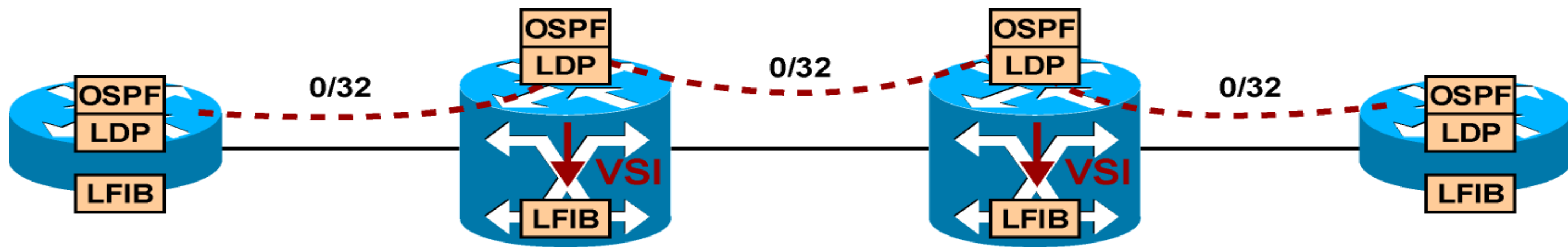
- Passive LDP peer sends Initialization message and/or keepalive message to active LDP peer if Initialization message parameters are acceptable
- Passive LDP peer could also send Error Notification & close the LDP connection if something was unacceptable

# 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.

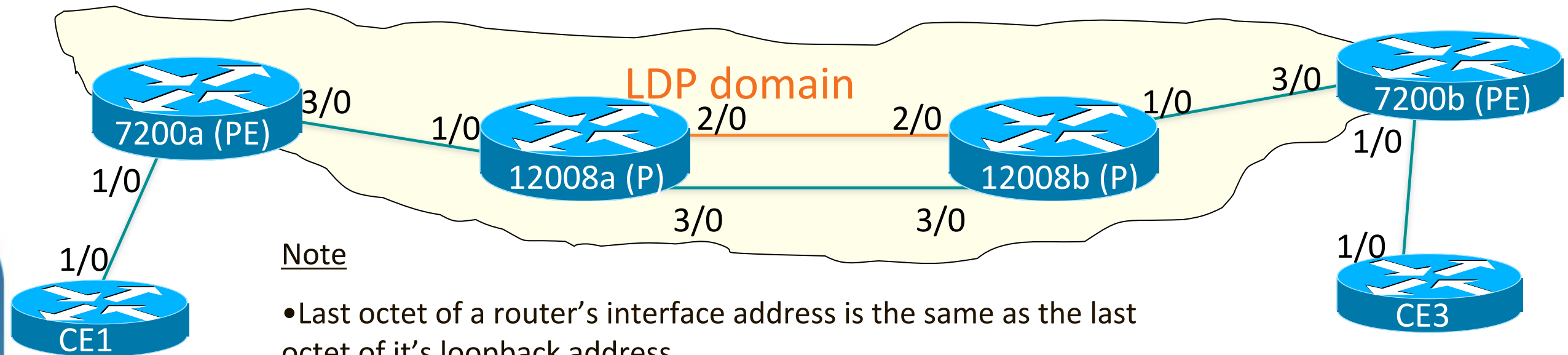
# Agenda LDP Configuration and Monitoring

- Configuration
- Verifying Your Configuration
- Monitoring LDP

# Configuring MPLS

- **Mandatory**
  - Enable CEF switching if this is not the platform default
  - Configure TDP or LDP on every label-enabled interface
  - TDP not supported on IOS XR
- **Optional**
  - Configure MTU size for labeled packets
  - Configure IP TTL propagation
  - Configure conditional label advertising

# Network Topology



Note

- Last octet of a router's interface address is the same as the last octet of its loopback address
- All pt-pt links are /24

	7200a	CE1	12008a	12008b	7200b	CE3
7200a	4.4.4.4 (lo0)	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) 10.0.5.0 (3/0) 10.0.4.0 (2/0)	10.0.4.0 (2/0) 10.0.5.0 (3/0)		
12008b				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 (IOS)

## 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 LDP (IOS XR)

```
mpls ldp
  router-id 1.1.1.1
  explicit-null (optional)
  log
  neighbor
  !
  interface GigabitEthernet0/4/1/3
  !
  !
mpls ip-ttl-propagate disable (optional)
mpls oam
!
```

# Configuring Conditional Label Distribution (IOS)

```
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.

# Configuring Conditional Label Distribution (IOS XR)

```
label advertise [ disable | for prefix-acl [ to peer-acl ] |  
interface type interface-path-id ]
```

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

- Enable conditional label advertisement

```
no tag-switching advertise-tags
!
! Configure conditional advertisements
!
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
```

# Conditional Label Distribution Router Configuration (IOS XR)

- Enable conditional label advertisement

```
ipv4 access-list pfx_acl_1
 10 permit ipv4 host 31.0.0.0 any
!
ipv4 access-list peer_acl_1
 10 permit ipv4 host 9.9.9.9 any
!
mpls ldp
  label
    advertise
    disable
    for pfx_acl_1 to peer_acl_1
  !
!
```

# Agenda LDP Configuration and Monitoring

- Configuration
- Verifying Your Configuration
- Monitoring LDP

# Verifying your configuration (IOS)

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

# Verifying your configuration (IOS XR)

```
mpls ldp
  router-id 1.1.1.1
  log
  neighbor
  !
interface GigabitEthernet0/4/1/3
  !
  !
mpls oam
  !
```

# Agenda LDP Configuration and Monitoring

- 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
Serial2/0
PE2#
```

Interface	IP	Tunnel	Operational
Serial2/0	Yes (ldp)	No	Yes

MPLS Enabled

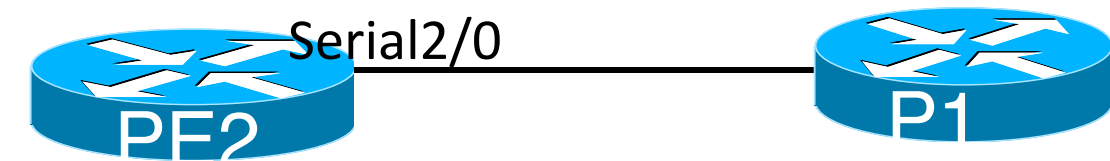
```
PE2#sh mpls interface ser2/0 detail
Interface Serial2/0:
```

```
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
```

LDP Enabled

MPLS MTU

```
MTU = 1508
PE2#
```



```
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/rcv  
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 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. Notice “Advertised to:” field.
- “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
```

# 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 tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes switched	Outgoing interface	Next Hop
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
```

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

BUILT FOR  
THE HUMAN  
NETWORK

