

Implementing MPLS Label Distribution Protocol

The Multiprotocol Label Switching (MPLS) is a standards-based solution driven by the Internet Engineering Task Force (IETF) that was devised to convert the Internet and IP backbones from best-effort networks into business-class transport mediums.

MPLS, with its label switching capabilities, eliminates the need for an IP route look-up and creates a virtual circuit (VC) switching function, allowing enterprises the same performance on their IP-based network services as with those delivered over traditional networks such as Frame Relay or ATM.

Label Distribution Protocol (LDP) performs label distribution in MPLS environments. LDP provides the following capabilities:

- LDP performs hop-by-hop or dynamic path setup; it does not provide end-to-end switching services.
- LDP assigns labels to routes using the underlying Interior Gateway Protocols (IGP) routing protocols.
- LDP provides constraint-based routing using LDP extensions for traffic engineering.

Finally, LDP is deployed in the core of the network and is one of the key protocols used in MPLS-based Layer 2 and Layer 3 virtual private networks (VPNs).

Feature History for Implementing MPLS LDP

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	No modification.
Release 4.0.1	Support was added for these features: • IP LDP Fast Reroute Loop Free Alternate • Downstream on Demand
Release 4.2.1	Support was added for LDP Implicit Null for IGP Routes.
Release 5.1	Support was added for MPLS over IRB.
Release 5.1.1	The feature MPLS LDP Carrier Supporting Carrier for Multiple VRFs was introduced.
Release 5.3.0	IPv6 Support in MPLS LDP was introduced.

Release	Modification
Release 6.0.1	Dual-Stack Capability TLV feature was introduced.
Release 7.0.1	The UDP Decapsulation of MPLS-Over-UDP Traffic feature was introduced.
Release 7.1.1	Multiple MPLS-TE tunnel end points can be enabled on an LER using the TLV 132 function in IS-IS.

- Prerequisites for Implementing Cisco MPLS LDP, on page 2
- Information About Implementing Cisco MPLS LDP, on page 2
- How to Implement MPLS LDP, on page 33
- Configuration Examples for Implementing MPLS LDP, on page 98
- Entropy Label Support for Transit Routers, on page 122
- Additional References, on page 124

Prerequisites for Implementing Cisco MPLS LDP

These prerequisites are required to implement MPLS LDP:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must be running Cisco IOS XR software.
- You must install a composite mini-image and the MPLS package.
- You must activate IGP.
- We recommend to use a lower session holdtime bandwidth such as neighbors so that a session down occurs before an adjacency-down on a neighbor. Therefore, the following default values for the hello times are listed:
  - Holdtime is 15 seconds.
  - Interval is 5 seconds.

  For example, the LDP session holdtime can be configured as 30 seconds by using the **holdtime** command.

Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand these concepts:

Overview of Label Distribution Protocol

LDP performs label distribution in MPLS environments. LDP uses hop-by-hop or dynamic path setup, but does not provide end-to-end switching services. Labels are assigned to routes that are chosen by the underlying IGP routing protocols. The Label Switched Paths (LSPs) that result from the routes, forward labeled traffic across the MPLS backbone to adjacent nodes.

Label Switched Paths

LSPs are created in the network through MPLS. They can be created statically, by RSVP traffic engineering (TE), or by LDP. LSPs created by LDP perform hop-by-hop path setup instead of an end-to-end path.

LDP Control Plane

The control plane enables label switched routers (LSRs) to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information.

Related Topics

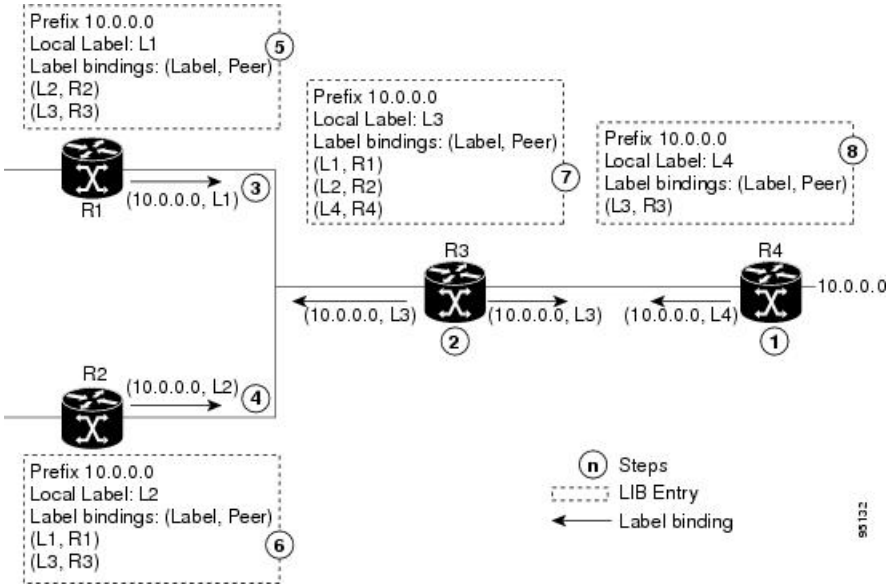
Configuring LDP Discovery Parameters, on page 33
Configuring LDP Discovery Over a Link, on page 36
Configuring LDP Link: Example, on page 98
Configuring LDP Discovery for Active Targeted Hellos, on page 38
Configuring LDP Discovery for Passive Targeted Hellos, on page 40
Configuring LDP Discovery for Targeted Hellos: Example, on page 98

Exchanging Label Bindings

LDP creates LSPs to perform the hop-by-hop path setup so that MPLS packets can be transferred between the nodes on the MPLS network.

Figure 1: Setting Up Label Switched Paths

This figure illustrates the process of label binding exchange for setting up LSPs.



For a given network (10.0.0.0), hop-by-hop LSPs are set up between each of the adjacent routers (or, nodes) and each node allocates a local label and passes it to its neighbor as a binding:

1. R4 allocates local label L4 for prefix 10.0.0.0 and advertises it to its neighbors (R3).
2. R3 allocates local label L3 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R2, R4).
3. R1 allocates local label L1 for prefix 10.0.0.0 and advertises it to its neighbors (R2, R3).

4. R2 allocates local label L2 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R3).
5. R1's label information base (LIB) keeps local and remote labels bindings from its neighbors.
6. R2's LIB keeps local and remote labels bindings from its neighbors.
7. R3's LIB keeps local and remote labels bindings from its neighbors.
8. R4's LIB keeps local and remote labels bindings from its neighbors.

Related Topics

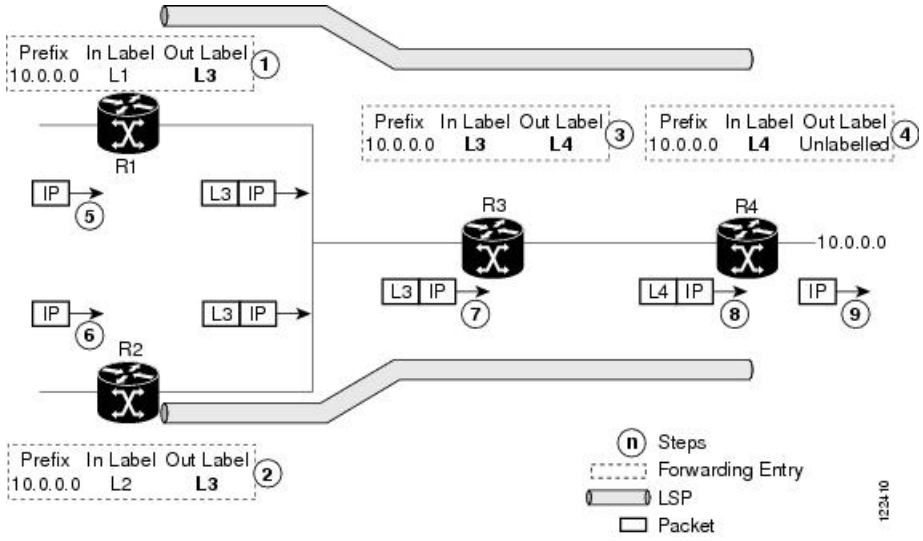
Setting Up LDP Neighbors, on page 43
Configuring LDP Neighbors: Example, on page 100

LDP Forwarding

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in the following figure.

Figure 2: Forwarding Setup

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in this figure.



1. Because R3 is next hop for 10.0.0.0 as notified by the FIB, R1 selects label binding from R3 and installs forwarding entry (label L1, label L3).
2. Because R3 is next hop for 10.0.0.0 (as notified by FIB), R2 selects label binding from R3 and installs forwarding entry (label L2, label L3).
3. Because R4 is next hop for 10.0.0.0 (as notified by FIB), R3 selects label binding from R4 and installs forwarding entry (label L3, label L4).
4. Because next hop for 10.0.0.0 (as notified by FIB) is beyond R4, R4 uses NO-LABEL as the outbound and installs the forwarding entry (label L4); the outbound packet is forwarded IP-only.
5. Incoming IP traffic on ingress LSR R1 gets label-imposed and is forwarded as an MPLS packet with label L3.

6. Incoming IP traffic on ingress LSR R2 gets label-imposed and is forwarded as an MPLS packet with label L3.
7. R3 receives an MPLS packet with label L3, looks up in the MPLS label forwarding table and switches this packet as an MPLS packet with label L4.
8. R4 receives an MPLS packet with label L4, looks up in the MPLS label forwarding table and finds that it should be Unlabeled, pops the top label, and passes it to the IP forwarding plane.
9. IP forwarding takes over and forwards the packet onward.



Note For local labels, only up to 12000 rewrites are supported. If the rewrites exceed this limit, MPLS LSD or MPLS LDP or both the processes may crash.

Related Topics

Setting Up LDP Forwarding, on page 46
Configuring LDP Forwarding: Example, on page 100

LDP Graceful Restart

LDP (Label Distribution Protocol) graceful restart provides a control plane mechanism to ensure high availability and allows detection and recovery from failure conditions while preserving Nonstop Forwarding (NSF) services. Graceful restart is a way to recover from signaling and control plane failures without impacting forwarding.

Without LDP graceful restart, when an established session fails, the corresponding forwarding states are cleaned immediately from the restarting and peer nodes. In this case LDP forwarding restarts from the beginning, causing a potential loss of data and connectivity.

The LDP graceful restart capability is negotiated between two peers during session initialization time, in FT SESSION TLV. In this typed length value (TLV), each peer advertises the following information to its peers:

Reconnect time

Advertises the maximum time that other peer will wait for this LSR to reconnect after control channel failure.

Recovery time

Advertises the maximum time that the other peer has on its side to reinstate or refresh its states with this LSR. This time is used only during session reestablishment after earlier session failure.

FT flag

Specifies whether a restart could restore the preserved (local) node state for this flag.

Once the graceful restart session parameters are conveyed and the session is up and running, graceful restart procedures are activated.

When configuring the LDP graceful restart process in a network with multiple links, targeted LDP hello adjacencies with the same neighbor, or both, make sure that graceful restart is activated on the session before any hello adjacency times out in case of neighbor control plane failures. One way of achieving this is by configuring a lower session hold time between neighbors such that session timeout occurs before hello adjacency timeout. It is recommended to set LDP session hold time using the following formula:

```
Session Holdtime <= (Hello holdtime - Hello interval) * 3
```

This means that for default values of 15 seconds and 5 seconds for link Hello holdtime and interval respectively, session hold time should be set to 30 seconds at most.

For more information about LDP commands, see *MPLS Label Distribution Protocol Commands* module of the *MPLS Command Reference for Cisco ASR 9000 Series Routers*.

Related Topics

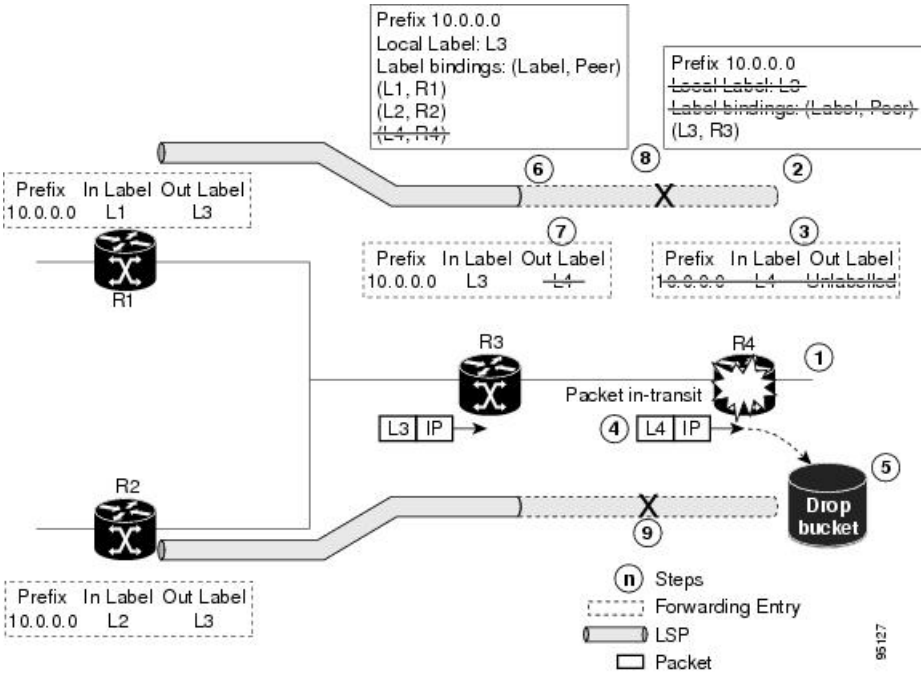
Phases in Graceful Restart, on page 7
Recovery with Graceful-Restart, on page 7
Setting Up LDP NSF Using Graceful Restart, on page 50
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 100

Control Plane Failure

When a control plane failure occurs, connectivity can be affected. The forwarding states installed by the router control planes are lost, and the in-transit packets could be dropped, thus breaking NSF.

Figure 3: Control Plane Failure

This figure illustrates a control plane failure and shows the process and results of a control plane failure leading to loss of connectivity.



1. The R4 LSR control plane restarts.
2. LIB is lost when the control plane restarts.
3. The forwarding states installed by the R4 LDP control plane are immediately deleted.
4. Any in-transit packets flowing from R3 to R4 (still labeled with L4) arrive at R4.

5. The MPLS forwarding plane at R4 performs a lookup on local label L4 which fails. Because of this failure, the packet is dropped and NSF is not met.
6. The R3 LDP peer detects the failure of the control plane channel and deletes its label bindings from R4.
7. The R3 control plane stops using outgoing labels from R4 and deletes the corresponding forwarding state (rewrites), which in turn causes forwarding disruption.
8. The established LSPs connected to R4 are terminated at R3, resulting in broken end-to-end LSPs from R1 to R4.
9. The established LSPs connected to R4 are terminated at R3, resulting in broken LSPs end-to-end from R2 to R4.

Phases in Graceful Restart

The graceful restart mechanism is divided into different phases:

Control communication failure detection

Control communication failure is detected when the system detects either:

- Missed LDP hello discovery messages
- Missed LDP keepalive protocol messages
- Detection of Transmission Control Protocol (TCP) disconnection a with a peer

Forwarding state maintenance during failure

Persistent forwarding states at each LSR are achieved through persistent storage (checkpoint) by the LDP control plane. While the control plane is in the process of recovering, the forwarding plane keeps the forwarding states, but marks them as stale. Similarly, the peer control plane also keeps (and marks as stale) the installed forwarding rewrites associated with the node that is restarting. The combination of local node forwarding and remote node forwarding plane states ensures NSF and no disruption in the traffic.

Control state recovery

Recovery occurs when the session is reestablished and label bindings are exchanged again. This process allows the peer nodes to synchronize and to refresh stale forwarding states.

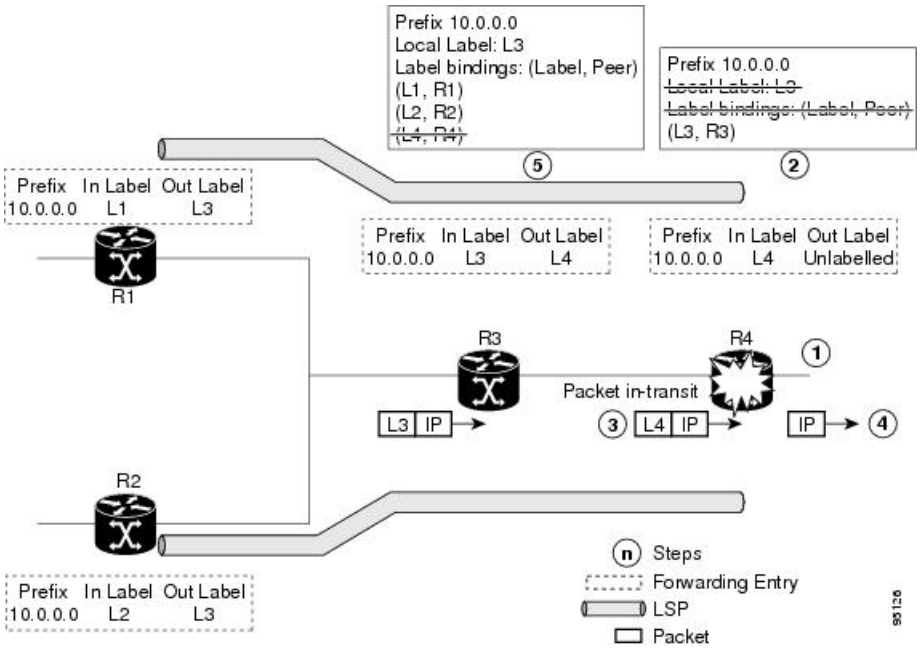
Related Topics

LDP Graceful Restart, on page 5
Recovery with Graceful-Restart, on page 7
Setting Up LDP NSF Using Graceful Restart, on page 50
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 100

Recovery with Graceful-Restart

Figure 4: Recovering with Graceful Restart

This figure illustrates the process of failure recovery using graceful restart.



1. The router R4 LSR control plane restarts.
2. With the control plane restart, LIB is gone but forwarding states installed by R4's LDP control plane are not immediately deleted but are marked as stale.
3. Any in-transit packets from R3 to R4 (still labeled with L4) arrive at R4.
4. The MPLS forwarding plane at R4 performs a successful lookup for the local label L4 as forwarding is still intact. The packet is forwarded accordingly.
5. The router R3 LDP peer detects the failure of the control plane and channel and deletes the label bindings from R4. The peer, however, does not delete the corresponding forwarding states but marks them as stale.
6. At this point there are no forwarding disruptions.
7. The peer also starts the neighbor reconnect timer using the reconnect time value.
8. The established LSPs going toward the router R4 are still intact, and there are no broken LSPs.

When the LDP control plane recovers, the restarting LSR starts its forwarding state hold timer and restores its forwarding state from the checkpointed data. This action reinstates the forwarding state and entries and marks them as old.

The restarting LSR reconnects to its peer, indicated in the FT Session TLV, that it either was or was not able to restore its state successfully. If it was able to restore the state, the bindings are resynchronized.

The peer LSR stops the neighbor reconnect timer (started by the restarting LSR), when the restarting peer connects and starts the neighbor recovery timer. The peer LSR checks the FT Session TLV if the restarting peer was able to restore its state successfully. It reinstates the corresponding forwarding state entries and receives binding from the restarting peer. When the recovery timer expires, any forwarding state that is still marked as stale is deleted.

If the restarting LSR fails to recover (restart), the restarting LSR forwarding state and entries will eventually timeout and is deleted, while neighbor-related forwarding states or entries are removed by the Peer LSR on expiration of the reconnect or recovery timers.

Related Topics

LDP Graceful Restart, on page 5
Phases in Graceful Restart, on page 7
Setting Up LDP NSF Using Graceful Restart, on page 50
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 100

Label Advertisement Control (Outbound Filtering)

By default, LDP advertises labels for all the prefixes to all its neighbors. When this is not desirable (for scalability and security reasons), you can configure LDP to perform outbound filtering for local label advertisement for one or more prefixes to one more peers. This feature is known as *LDP outbound label filtering*, or *local label advertisement control*.

Related Topics

Configuring Label Advertisement Control (Outbound Filtering), on page 42
Configuring Label Advertisement (Outbound Filtering): Example, on page 99

Label Acceptance Control (Inbound Filtering)

By default, LDP accepts labels (as remote bindings) for all prefixes from all peers. LDP operates in liberal label retention mode, which instructs LDP to keep remote bindings from all peers for a given prefix. For security reasons, or to conserve memory, you can override this behavior by configuring label binding acceptance for set of prefixes from a given peer.

The ability to filter remote bindings for a defined set of prefixes is also referred to as *LDP inbound label filtering*.

Note Inbound filtering can also be implemented using an outbound filtering policy; however, you may not be able to implement this system if an LDP peer resides under a different administration domain. When both inbound and outbound filtering options are available, we recommend that you use outbound label filtering.

Related Topics

Configuring Label Acceptance Control (Inbound Filtering), on page 53
Configuring Label Acceptance (Inbound Filtering): Example, on page 101

Local Label Allocation Control

By default, LDP allocates local labels for all prefixes that are not Border Gateway Protocol (BGP) prefixes[1]. This is acceptable when LDP is used for applications other than Layer 3 virtual private networks (L3VPN) core transport. When LDP is used to set up transport LSPs for L3VPN traffic in the core, it is not efficient or even necessary to allocate and advertise local labels for, potentially, thousands of IGP prefixes. In such a case, LDP is typically required to allocate and advertise local label for loopback /32 addresses for PE routers. This

[1] For L3VPN Inter-AS option C, LDP may also be required to assign local labels for some BGP prefixes.

is accomplished using LDP local label allocation control, where an access list can be used to limit allocation of local labels to a set of prefixes. Limiting local label allocation provides several benefits, including reduced memory usage requirements, fewer local forwarding updates, and fewer network and peer updates.

Tip You can configure label allocation using an IP access list to specify a set of prefixes that local labels can allocate and advertise.

Related Topics

Configuring Local Label Allocation Control, on page 54
Configuring Local Label Allocation Control: Example, on page 101

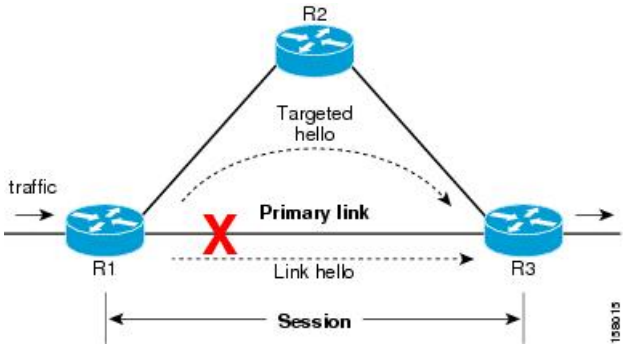
Session Protection

When a link comes up, IP converges earlier and much faster than MPLS LDP and may result in MPLS traffic loss until MPLS convergence. If a link flaps, the LDP session will also flap due to loss of link discovery. LDP session protection minimizes traffic loss, provides faster convergence, and protects existing LDP (link) sessions by means of "parallel" source of targeted discovery hello. An LDP session is kept alive and neighbor label bindings are maintained when links are down. Upon reestablishment of primary link adjacencies, MPLS convergence is expedited as LDP need not relearn the neighbor label bindings.

LDP session protection lets you configure LDP to automatically protect sessions with all or a given set of peers (as specified by peer-acl). When configured, LDP initiates backup targeted hellos automatically for neighbors for which primary link adjacencies already exist. These backup targeted hellos maintain LDP sessions when primary link adjacencies go down.

The Session Protection figure illustrates LDP session protection between neighbors R1 and R3. The primary link adjacency between R1 and R3 is directly connected link and the backup; targeted adjacency is maintained between R1 and R3. If the direct link fails, LDP link adjacency is destroyed, but the session is kept up and running using targeted hello adjacency (through R2). When the direct link comes back up, there is no change in the LDP session state and LDP can converge quickly and begin forwarding MPLS traffic.

Figure 5: Session Protection



Note When LDP session protection is activated (upon link failure), protection is maintained for an unlimited period time.

Related Topics

Configuring Session Protection, on page 55
Configuring LDP Session Protection: Example, on page 101

IGP Synchronization

Lack of synchronization between LDP and IGP can cause MPLS traffic loss. Upon link up, for example, IGP can advertise and use a link before LDP convergence has occurred; or, a link may continue to be used in IGP after an LDP session goes down.

LDP IGP synchronization synchronizes LDP and IGP so that IGP advertises links with regular metrics only when MPLS LDP is converged on that link. LDP considers a link converged when at least one LDP session is up and running on the link for which LDP has sent its applicable label bindings and received at least one label binding from the peer. LDP communicates this information to IGP upon link up or session down events and IGP acts accordingly, depending on sync state.

In the event of an LDP graceful restart session disconnect, a session is treated as converged as long as the graceful restart neighbor is timed out. Additionally, upon local LDP restart, a checkpointed recovered LDP graceful restart session is used and treated as converged and is given an opportunity to connect and resynchronize.

Under certain circumstances, it might be required to delay declaration of resynchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.

From the 7.1.1 release, you can configure multiple MPLS-TE tunnel end points on an LER using the TLV 132 function in IS-IS. You can configure a maximum of 63 IPv4 addresses or 15 IPv6 addresses on an LER.

Note The configuration for LDP IGP synchronization resides in respective IGPs (OSPF and IS-IS) and there is no LDP-specific configuration for enabling of this feature. However, there is a specific LDP configuration for IGP sync delay timer.

Related Topics

Configuring LDP IGP Synchronization: OSPF, on page 56
Configuring LDP IGP Synchronization—OSPF: Example, on page 102
Configuring LDP IGP Synchronization: ISIS, on page 60
Configuring LDP IGP Synchronization—ISIS: Example, on page 102

IGP Auto-configuration

To enable LDP on a large number of interfaces, IGP auto-configuration lets you automatically configure LDP on all interfaces associated with a specified IGP interface; for example, when LDP is used for transport in the core network. However, there needs to be one IGP set up to enable LDP auto-configuration.

Typically, LDP assigns and advertises labels for IGP routes and must often be enabled on all active interfaces by an IGP. Without IGP auto-configuration, you must define the set of interfaces under LDP, a procedure that is time-intensive and error-prone.

Note LDP auto-configuration is supported for IPv4 unicast family in the default VRF. The IGP is responsible for verifying and applying the configuration.

You can also disable auto-configuration on a per-interface basis. This permits LDP to enable all IGP interfaces except those that are explicitly disabled and prevents LDP from enabling an interface when LDP auto-configuration is configured under IGP.

Related Topics

Enabling LDP Auto-Configuration for a Specified OSPF Instance, on page 61
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance, on page 62
Disabling LDP Auto-Configuration, on page 63
Configuring LDP Auto-Configuration: Example, on page 102

LDP Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) failover, invisible to routing peers with minimal to no disruption of convergence performance. By default, NSR is globally enabled on all LDP sessions except AToM.

A disruption in service may include any of these events:

- Route processor (RP) or distributed route processor (DRP) failover
- LDP process restart
- In-service system upgrade (ISSU)
- Minimum disruption restart (MDR)

Note Unlike graceful restart functionality, LDP NSR does not require protocol extensions and does not force software upgrades on other routers in the network, nor does LDP NSR require peer routers to support NSR.

Process failures of active TCP or LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action. For more information about how to configure switchover as a recovery action for NSR, see *Configuring Transports* module in *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*.

Related Topics

Configuring LDP Nonstop Routing, on page 64

IP LDP Fast Reroute Loop Free Alternate

The IP Fast Reroute is a mechanism that enables a router to rapidly switch traffic, after an adjacent link failure, node failure, or both, towards a pre-programmed loop-free alternative (LFA) path. This LFA path is used to switch traffic until the router installs a new primary next hop again, as computed for the changed network topology.

The goal of LFA FRR is to reduce failure reaction time to 50 milliseconds by using a pre-computed alternate next hop, in the event that the currently selected primary next hop fails, so that the alternate can be rapidly used when the failure is detected.

This feature targets to address the fast convergence ability by detecting, computing, updating or enabling prefix independent pre-computed alternate loop-free paths at the time of failure.

IGP pre-computes a backup path per IGP prefix. IGP selects one and only one backup path per primary path. RIB installs the best path and download path protection information to FIB by providing correct annotation for protected and protecting paths. FIB pre-installs the backup path in dataplane. Upon the link or node failure, the routing protocol detects the failure, all the backup paths of the impacted prefixes are enabled in a prefix-independent manner.

Prerequisites

The Label Distribution Protocol (LDP) can use the loop-free alternates as long as these prerequisites are met:

The Label Switching Router (LSR) running LDP must distribute its labels for the Forwarding Equivalence Classes (FECs) it can provide to all its neighbors, regardless of whether they are upstream, or not.

There are two approaches in computing LFAs:

- **Link-based (per-link)**--In link-based LFAs, all prefixes reachable through the primary (protected) link share the same backup information. This means that the whole set of prefixes, sharing the same primary, also share the repair or fast reroute (FRR) ability. The per-link approach protects only the next hop address. The per-link approach is suboptimal and not the best for capacity planning. This is because all traffic is redirected to the next hop instead of being spread over multiple paths, which may lead to potential congestion on link to the next hop. The per-link approach does not provide support for node protection.
- **Prefix-based (per-prefix)**--Prefix-based LFAs allow computing backup information per prefix. It protects the destination address. The per-prefix approach is the preferred approach due to its greater applicability, and the greater protection and better bandwidth utilization that it offers.



Note The repair or backup information computed for a given prefix using prefix-based LFA may be different from the computed by link-based LFA.

The per-prefix LFA approach is preferred for LDP IP Fast Reroute LFA for these reasons:

- Better node failure resistance
- Better capacity planning and coverage

Features Not Supported

These interfaces and features are not supported for the IP LDP Fast Reroute Loop Free Alternate feature:

- BVI interface (IRB) is not supported either as primary or backup path.
- GRE tunnel is not supported either as primary or backup path.
- Cisco ASR 9000 Series SPA Interface Processor-700 POS line card on Cisco ASR 9000 Series Router is not supported as primary link. It can be used as LFA backup only on main interface.

- In a multi-topology scenerio, the route in topology T can only use LFA within topology T. Hence, the availability of a backup path depends on the topology.

For more information about configuring the IP Fast Reroute Loop-free alternate , see Implementing IS-IS on Cisco IOS XR Software module of the *Routing Configuration Guide for Cisco ASR 9000 Series Routers*.

Related Topics

Configure IP LDP Fast Reroute Loop Free Alternate: Examples, on page 103
Verify IP LDP Fast Reroute Loop Free Alternate: Example, on page 105

Downstream on Demand

This Downstream on demand feature adds support for downstream-on-demand mode, where the label is not advertised to a peer, unless the peer explicitly requests it. At the same time, since the peer does not automatically advertise labels, the label request is sent whenever the next-hop points out to a peer that no remote label has been assigned.

To enable downstream-on-demand mode, this configuration must be applied at mpls ldp configuration mode:

```
mpls ldp downstream-on-demand with ACL
```

The ACL contains a list of peer IDs that are configured for downstream-on-demand mode. When the ACL is changed or configured, the list of established neighbors is traversed. If a session's downstream-on-demand configuration has changed, the session is reset in order that the new down-stream-on-demand mode can be configured. The reason for resetting the session is to ensure that the labels are properly advertised between the peers. When a new session is established, the ACL is verified to determine whether the session should negotiate for downstream-on-demand mode. If the ACL does not exist or is empty, downstream-on-demand mode is not configured for any neighbor.

For it to be enabled, the Downstream on demand feature has to be configured on both peers of the session. If only one peer in the session has downstream-on-demand feature configured, then the session does not use downstream-on-demand mode.

If, after, a label request is sent, and no remote label is received from the peer, the router will periodically resend the label request. After the peer advertises a label after receiving the label request, it will automatically readvertise the label if any label attribute changes subsequently.

Related Topics

Configuring LDP Downstream on Demand mode, on page 67

Explicit-Null and Implicit-Null Labels

Cisco MPLS LDP uses null label, implicit or explicit, as local label for routes or prefixes that terminate on the given LSR. These routes include all local, connected, and attached networks. By default, the null label is **implicit-null** that allows LDP control plane to implement penultimate hop popping (PHOP) mechanism. When this is not desirable, you can configure **explicit-null** that allows LDP control plane to implement ultimate hop popping (UHOP) mechanism. You can configure this explicit-null feature on the ultimate hop LSR. This configuration knob includes an access-list to specify the IP prefixes for which PHOP is desired.

This new enhancement allows you to configure implicit-null local label for **non-egress (ultimate hop LSR)** prefixes by using the **implicit-null-override** command. This enforces implicit-null local label for a specific prefix even if the prefix requires a non-null label to be allocated by default. For example, by default, an LSR

allocates and advertises a non-null label for an IGP route. If you wish to terminate LSP for this route on penultimate hop of the LSR, you can enforce implicit-null label allocation and advertisement for this prefix using **implicit-null-override** feature.



Note If a given prefix is permitted in both explicit-null and implicit-null-override feature, then implicit-null-override supercedes and an implicit-null label is allocated and advertised for the prefix.

In order to enable implicit-null-override mode, this configuration must be applied at MPLS LDP label configuration mode:

```
mpls ldp
   label
     implicit-null-override for <prefix><ACL>

  !
```

This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.

MPLS over IRB

The Integrated Routing and Bridging (IRB) feature in Cisco IOS XR Software enables routing of a given protocol between routed interfaces and bridge groups within a single router. IRB support for MPLS introduces these capabilities:

- Bridge-Group Virtual Interface (BVI) support under MPLS LDP
- Targeted LDP session to BVI neighbor
- MPLS OAM for BVI interfaces
- Netflow for BVI interfaces while MPLS is enabled
- L2VPN using targeted MPLS LDP to BVI destination
- L3VPN
- 6PE/6VPE

MPLS over IRB is supported completely on ASR 9000 Enhanced Ethernet Line Card and Cisco ASR 9001. MPLS over IRB is not supported on ASR 9000 Ethernet Line Card.

For more information on MPLS over IRB, see the *Implementing MPLS Label Distribution Protocol* chapter in the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*. For more information on MPLS over IRB commands, see the *MPLS Label Distribution Protocol Commands* chapter in the *MPLS Command Reference for Cisco ASR 9000 Series Routers*.

MPLS LDP Carrier Supporting Carrier for Multiple VRFs

The carrier supporting carrier (CSC) support for MPLS LDP feature enables MPLS label distribution protocol (LDP) to provide CSC support for Layer 3 Virtual Private Networks (L3VPN). To support LDP as label distribution protocol between PE-CE devices in an MPLS CSC L3VPN, LDP is required to operate in multiple Virtual Private Network routing and forwarding (VRF) contexts.

MPLS Carrier Supporting Carrier L3VPN: Introduction

The carrier supporting carrier feature enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the *backbone carrier*. The service provider that uses the segment of the backbone network is called the *customer carrier*.

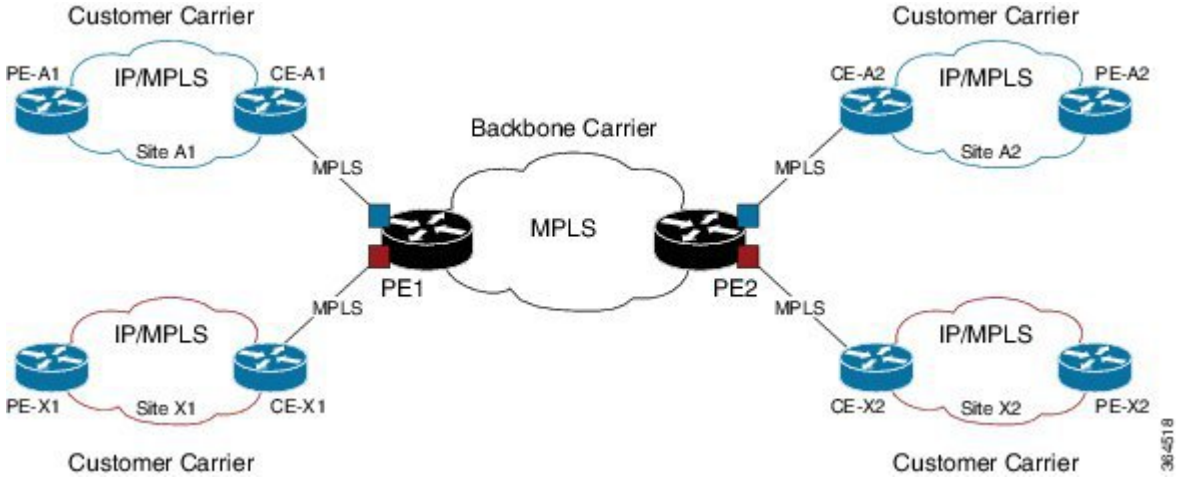
A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

In either case, MPLS is run in the backbone network and between the backbone and customer carrier (the PE-CE link).

Figure 6: MPLS Carrier Supporting Carrier L3VPN

This figure illustrates an MPLS CSC L3VPN.



The figure shows two customers, A and X, connecting their remote sites through the backbone carrier. The PE device of the backbone network connects with both customers through MPLS but under different VRFs according to interface-VRF mapping. The MPLS label distribution protocol for PE-CE connectivity can be either BGP or LDP, and requires them to run in a customer VRF context on the PE device.

Benefits of MPLS LDP CSC

The MPLS LDP CSC provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.
- The MPLS LDP CSC feature is scalable. CSC can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The CSC feature

enables tens of thousands of VPNs to be configured over the same network, and it allows a service provider to offer both VPN and internet services.

- The MPLS LDP CSC feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS LDP CSC feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.
- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.
- Customer carriers can use any link layer technology (SONET, Digital Subscriber Line, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS LDP CSC feature is link layer independent. The CE routers and PE routers use IP or MPLS to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Multiple VRF Support

To support multiple VRFs, IOS XR LDP configuration model is extended to allow VRF submode and per-VRF configuration and feature or interface enabling.

IOS XR LDP process is not distributed nor it is multi-instance, hence the single LDP process services all the configured VRFs. In large scale VRF deployment, it is recommended to enable VRF under LDP with appropriate policies and label filtering.

RSI

To obtain VRF and routing tables' related information, LDP interacts with the router space infrastructure (RSI) server. For every LDP enabled non-default VRF, LDP registers with RSI to get notifications upon VRF default (IPv4/IPv6) tables getting created or deleted, and populate the LDP VRF database accordingly.

VRF Table ID Database

A new database is added in the LDP process to keep track of all VRFs enabled under LDP. This database holds both active as well as forward-reference VRF records. In addition to serving as an LDP context, each active record of this database also holds VRF's default (IPv4/IPv6 unicast) table IDs.

VRF-Interface Mapping

To enable LDP on an interface for a given address family under a VRF context , it is required to list interface and its address family explicitly under a LDP VRF submode. LDP does not enforce or check correctness of the interface and VRF mapping at the time of configuration, and hence configuration may be accepted by LDP. The interface with incorrect VRF mapping is not made operational by LDP and remains down from the LDP point of view.

This means that an interface remains LDP operationally down for which either:

- LDP has not received any address update, or
- LDP has received update with different table-id (VRF) than configured under LDP.

Also, a user must not configure the same LDP interface under more than one VRF.

Context Isolation

Each active VRF under LDP points to a separate context under which LDP runs. This means that various variables, database, tables, FSM are kept separate in their respective VRF contexts and do not interfere or interact with each other. This allows the LDP to provide per-VRF isolation and support CSC with customers with overlapping addresses or routing information.

Default Context

The default (global) context is enabled at the time of the LDP process startup and remains enabled always. It is not possible to disable IPv4 for the default context. Also, it is required to explicitly enable IPv4 for non-default context. Therefore you can effectively disable IPv4 for non-default context by not configuring it. This means that, it is possible to enable or disable the non-default context under LDP, whereas the same is not possible for a default context.

Restrictions and Recommendations

The following restrictions and recommendations apply to the MPLS LDP CSC feature:

- Only IPv4 address family is supported for a default or a non-default VRF.
- No T-LDP support in a VRF context.
- An address family under VRF and VRF interface must be configured for non-default VRFs.
- Following scenarios are not supported :
  - Different VRFs between a given PE-CE device pair (VRFs configured on different links and interfaces)
  - LDP/BGP CSC co-existence on a given VRF between a given PE-CE device pair:
    - Single link
    - Parallel links: LDP CSC on one link and BGP CSC on the other
- LDP router-id must be configured per-VRF. If not configured for non-default VRF, LDP computes router-id from available loopback interfaces under the VRF.
- It is recommended to configure a routable discovery transport address under a VRF IPv4 address-family submode for deterministic transport endpoint and connection.
- When LDP CSC is configured and in use:
  - BGP label allocation policy for VRF prefixes must be per-prefix
  - Selective VRF Download (SVD) feature must be disabled

IPv6 Support in MPLS LDP

Internet Protocol version 6 (IPv6) support in MPLS LDP (Label Distribution Protocol) feature makes the LDP control plane to run on IPv6 in order to setup LSPs for IPv6 prefixes. This support enables most of the LDP functions supported on IPv4 to be extended to IPv6. In this context, support for native MPLS LDP over IPv6 is provided in order to seamlessly continue providing existing services while enabling new ones.

LDP associates a forwarding equivalence class (FEC) with each label switched path (LSP) it creates. The FEC associated with an LSP specifies which packets are mapped to that LSP. LDP establishes sessions with peers and exchanges FEC label bindings with them to enable creation of LSPs to carry MPLS traffic destined to IP prefixes.

LDP base specification, RFC 5036 defines procedures and messages for exchanging bindings for IPv4 and IPv6 addresses and routing prefixes. LDP IPv6 specification (draft-ietf-mpls-ldp-ipv6) updates LDP base specifications for IPv6 support, and further clarifies and focuses on the procedures for supporting LDP IPv6 control plane and binding advertisement.

The procedures of address bindings, label bindings, and forwarding setup are same for IPv4 and IPv6 address families in LDP. The only difference is that, a different address format is used according to the IP address family. While a single-stack IP address family (IPv4-only or IPv6-only) enabled interfaces between a set of routers is the most typical deployment, scenarios for LSR interconnections using both IPv4 and IPv6 interfaces are also supported.

IPv6 support in MPLS LDP implements draft-ietf-mpls-ldp-ipv6 version12 issued by the Internet Engineering Task Force (IETF).

LDP IPv6 Functionality

LDP functionality can be broadly divided into two categories:

- Control Plane

  Control plane includes functions such as: neighbor discovery (hello adjacencies), transport connection/endpoint (TCP connection), session and peering, and bindings exchange.

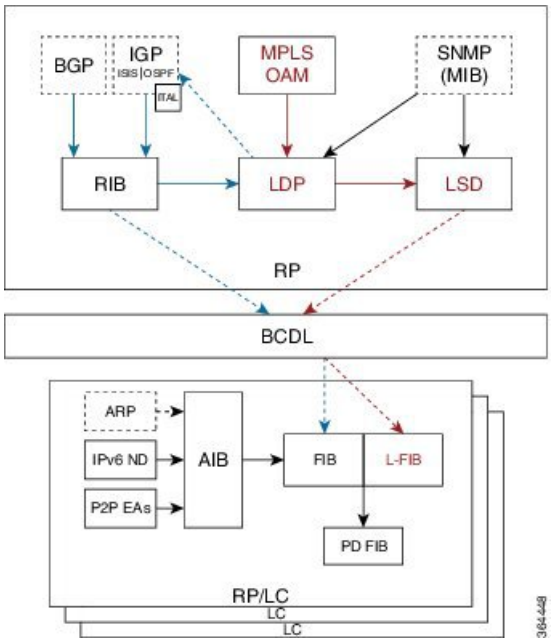
- LSP Setup

  LSP setup includes functions such as: acquire FEC information through RIB, assign and advertise local label bindings for FEC, advertise local (interface) IP address bindings and setup forwarding rewrites.

For the control plane, the underlying address family can be either IPv4-only, IPv6-only or both. Whereas for the LSP setup, an LSP is setup for IPv4 or IPv6 FEC prefix.

Figure 7: LDP IPv6 Architecture

This figure illustrates the main components that collaborate to achieve the required functionality for the LDP IPv6 feature.

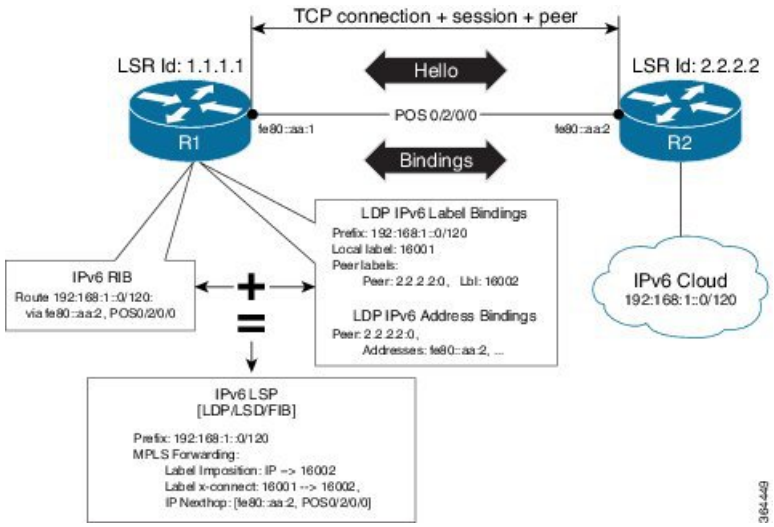


The functions of LDP in the MPLS LDP IPv6 setup are as follows:

- Receive routing updates from routing information base (RIB) for global IPv6 prefixes
- Assign local labels for IPv6 prefixes
- Receive IPv6 address or state notifications for local IPv6 enabled interfaces from IP Address Repository Manager (IP-ARM/IM) and LAS for IPv6 link-local unicast addresses
- Advertise/Accept IPv6 label bindings and address bindings to/from peers
- Setup MPLS forwarding to create IPv6 LSPs
- Provide IPv6 LSP information to MPLS OAM as and when requested
- Service MIB requests for IPv6 control plane queries and generate MIB traps
- Provide LDPv6 convergence status for a link to IGP for LDP-IGP Sync feature for IPv6
- Support IPv6 address family for all existing LDP features that intersect with prefixes and/or addresses

Figure 8: LDP IPv6 Control Plane and LSP Setup

This figure illustrates the high level functionality of LDP in terms of control plane and LSP setup in an IPv6 environment.



Topological Scenarios

A typical deployment scenario consists of single-stack IP address-family (IPv4-only or IPv6-only) enabled interfaces between a set of routers.

Three topology scenarios in which the LSRs are connected through one or more dual-stack LDP enabled interfaces, or one or more single-stack LDP enabled interfaces are defined as follows:

Note **R2** is the main router.

1. One dual-stack interface/same neighbor:

```
R1_ _ _ _ _ _ _ _R2
     IPv4+IPv6
```

2. Two single-stack interfaces/same neighbor:

```
     1. (IPv4)
R1_ _ _ _ _ _ _ _ _R2
  _ _ _ _ _ _ _ _ _
     2. (IPv6)
```

3. Two single-stack interfaces/different neighbors with different address families:

```
     1. (IPv4)          2. (IPv6)
R1_ _ _ _ _ _ _ _ R2_ _ _ _ _ _ _ _ R3
```

Case Study

A description of the control plane and LSP setup scenarios for the previously shown three configurations are as follows:

Case 1:

Neighbor Discovery: Both IPv4 and IPv6 Hellos sent on the interface to R1.

Transport Connection: IPv4 endpoints or IPv6 endpoints (as per user preference).

Label binding exchange: Both IPv4 and IPv6 prefixes.

Address binding exchange: Both IPv4 and IPv6 addresses.

LSPs: Both IPv4 and IPv6 over the same nexthop interface to R1.

Case 2:

Neighbor Discovery: IPv4 Hellos on interface-1 to R1 and IPv6 Hellos on interface-2 to R1.

Transport Connection: IPv4 endpoints or IPv6 endpoints (as per user preference).

Label binding exchange: Both IPv4 and IPv6 prefixes.

Address binding exchange: Both IPv4 and IPv6 addresses.

LSPs: IPv4 over nexthop interface-1 to R1 and IPv6 over nexthop interface-2 to R1.

Case 3:

Neighbor Discovery: IPv4 Hellos on interface-1 to R1 and IPv6 Hellos on interface-2 to R3.

Transport Connection: IPv4 endpoints with R1 and IPv6 endpoints with R3.

Label binding exchange: Both IPv4 and IPv6 prefixes to R1 and R3.



Note Even if all the three LSRs are dual-stack, traffic from R1 to R3 will not be completely labeled.

- If there is IPv6 traffic, it is unlabeled from R1 to R2. Labels are imposed only at R2 (although in this specific case implicit null imposition) to R3.
- If there is IPv4 traffic, it is labeled from R1 to R2. But the traffic will go unlabeled between R2 and R3 given that no IPv4 adjacency exists between R2 and R3.

Address binding exchange: Both IPv4 and IPv6 addresses to R1 and R3.

LSPs: IPv4 over nexthop interface-1 to R1 and IPv6 over nexthop interface-2 to R3.

Restrictions

IPv6 support in MPLS LDP has the following restrictions and constraints:

- IPv6 address family is supported only under default VRF
- Implicit enabling of IPv6 address family is not allowed. It needs explicit enabling.
- It is recommended to configure a routable IPv6 **discovery transport address** when only LDP IPv6 is configured without explicitly specifying a router-id

Features Supported in LDP IPv6

The following features are supported in LDP IPv6:

- Single-stack (native IPv6) and dual-stack (IPv4+IPv6) topologies
- New operating modes in LDP:
  - Native LDP IPv6
  - LDP IPv6 over IPv4 and LDP IPv4 over IPv6 connection endpoints

    LDP Hellos carry optional transport address type length value (TLV) to notify a peer about TCP or transport connection endpoint. An LSR can include either IPv4 or IPv6 transport address TLV in an IPv4 or IPv6 Hello message. There is no difference in the TLV format of transport address for IPv4 and IPv6.

    Only one transport connection is established between two discovered peers, whether there be single address family Hello adjacencies or multi-address family (both IPv4 and IPv6) Hello adjacencies.

    In a dual-stack setup, when LDP has the option to establish transport connection either using IPv4 endpoints or IPv6 endpoints, IPv6 connection is preferred over IPv4 connection. If LDP is locally enabled for both IPv4 and IPv6 address families, every new session is treated as potential dual-stack connection. Under such circumstances, IPv6 preference is kept in place for maximum fifteen seconds for the session to establish, after which the LDP tries to establish a connection with the peer using IPv4. A user can override this default behavior by specifying the preference for a set of dual-stack peers to use IPv4 transport for the connection. Furthermore, a user may also specify maximum wait time to wait to establish the preferred transport connection. If the preferred transport establishment times out, LDP tries to establish connection with other non-preferred transport address families. This applies to both the cases when an LSR acts as active side or passive side for the TCP connection.

    To override default IPv6 transport preference for dual-stack cases, use the **mpls ldp neighbor dual-stack transport-connection prefer ipv4 for-peers** command. To specify the maximum time the preferred address family connection must wait to establish a connection before resorting to a non-preferred address family, use the **mpls ldp neighbor dual-stack transport-connection max-wait** command.

    Once a transport connection is established, it is not torn down depending on preferences. If the address family related to established transport connection is disabled under LDP, the corresponding transport connection is reset to reestablish the connection.

    For a single-stack setup, there is no contention; the transport connection uses the given address family.

- LDP Control Plane is IPv6 aware
- LDP IPv6 LSP forwarding setup

  LDP interacts with LSD in order to setup IPv6 LSP forwarding. The steps involved in this interaction are:

  - Label allocation for an IPv6 prefix is learnt from RIB.
  - Setup imposition and label switching forwarding path for given IPv6 prefix by creating IPv6 forwarding rewrites.
  - Like LDP IPv4, rewrite delete and label free operations are performed when a route disappears or is disallowed under LDP due to label policy.

- There is no new requirement related to MPLS enabling or disabling. LDP also MPLS-enables in LSD (if not already) any LDP enabled interface, which is in the *UP* state for IP4 and/or IPv6 and has IPv4 and/or IPv6 addresses assigned.
- In case of dual-stack LDP, a single Resource-Complete is sent by LDP to LSD once RIB-Converged notification is received for both IPv4 and IPv6 redistribute tables.

- Distribution of IPv4 and IPv6 bindings over a single LDP session established over IPv4 or IPv6
- LDP Downstream on Demand
- LDP session protection

  LDP session protection is a feature to protect an IPv6 LDP session. In case of dual-stack hello adjacencies with a peer, there is only a single targeted hello adjacency to protect the session. Session protection forms targeted adjacency of address family same as the transport connection. For IPv6, the target of the session protection is the remote transport connection endpoint. For IPv4, the target of the session protection is remote LSR ID.

- LDP IGPv6 sync on IPv6 interface

  This feature lets IGP support LDP IGP Sync feature for IPv6 address family. This means that Intermediate System-to-Intermediate System (IS-IS) allows IGP under an interface's IPv6 address family, whereas OSPFv3 implements it just like existing support in OSPF for IPv4. When the IGP Sync feature is enabled, LDP convergence status on an interface is considered by the IGP under the context of a given address family. This behavior applies to IGP Sync for both non-TE as well as TE tunnel interfaces.

- LDP Typed Wildcard for IPv6 prefix FEC

  This feature adds support for Typed Wildcard for IPv6 Prefix FEC. The support includes:

  - Being able to send or receive IPv6 Prefix Typed Wildcard FEC element in label messages.
  - Respond to Typed Wildcard Label Requests received from peer by replaying its label database for IPv6 prefixes.
  - Make use of Typed Wildcard Label Requests towards peers to request replay of peer label database for IPv6 prefixes. For example, on local inbound policy changes.

- Label allocation, advertisement and accept policies for IPv6 prefixes
- Local label assignment and advertisement for IPv6 default-route (::/0)
- Session MD5 authentication for IPv6 transport
- IPv6 Explicit-Null label

  IPv6 explicit null label feature support includes:

  - Advertisement and receipt of IPv6 explicit-null label to and from peers.
  - IPv6 explicit-null outgoing label in forwarding setup.
  - Explicit-null advertisement policy for a set of IPv6 prefixes and/or set of peers.
  - Explicit-null configuration change. Change in explicit-null configuration is handled by first transferring a wildcard withdraw with null label to peer(s), followed by advertising the appropriate null (implicit or explicit) label to the peer(s) again. This works without any issue as long as a single IP address family is enabled. In case of a dual-stack LSR peer, a change of configuration related to

explicit-null advertisement for a given address family may cause unnecessary mix-up in the other address family.

- LDP IPv6 LFA FRR

  Local LFA FRR for IPv6 is supported. However, it is required that the primary and backup paths are of the same address family type, that is, an IPv4 primary path must not have an IPv6 backup path.

- NSF for LDP IPv6 traffic

  Non-stop forwarding (NSF) support is either provided through LDP NSR or graceful restart mechanisms.

- IGP/LDP NSR for IPv6
- IGP/LDP Graceful Restart for IPv6
- LDP ICCP IPv6 neighbor node

  LDP Inter-Chassis Communication Protocol (ICCP) is supported with IPv6 neighbor node. ICCP is used as a mechanism for multi-chassis LACP.

- SSO/ISSU for LDP IPv6
- MPLS OAM: New FECs

  LSPV supports two new FECs.

  - LDP IPv6 Prefix FEC Encoding/Decoding

    Label Switched Path Verification (LSPV) encodes/decodes the LDP IPv6 Prefix FEC. Prefix is in the network byte order and the trailing bits are to be set to zero when prefix length is shorter than 128 bits.

  - Generic IPv6 Prefix FEC Encoding/Decoding

    LSPV encodes/decodes the generic IPv6 Prefix FEC. Prefix is in the network byte order and the trailing bits are to be set to zero when prefix length is shorter than 128 bits.

    Generic IPv6 FEC is used in addition to the LDP IPv6 FEC. This serves the following primary purposes:

    - Allows user to perform LSP ping and traceroute to verify data plane without involving control plane of the FEC in echo request and response.
    - If support for a new FEC is preferred in the future, the generic FEC can be used until corresponding control plane is explicitly supported by LSPV.

- IPv6 LSR MIB

  MPLS OAM LDP MIBS is extended to support IPv6. All LSR MIB objects that reference an InSegment prefix and OutSegment next hop address are modified to support IPv6.

- LSP ping support for LDP IPv6
- LSP trace-route support for LDP IPv6
- LSP tree-trace support for LDP IPv6

The following features are not supported in LDP IPv6:

- LDPv6 over TEv4 (traffic engineering)
- L2VPN/PW (over IPv6 LSPs)
- L3VPN (over IPv6 LSPs)
- LDP auto-config for IPv6 IGP/Interfaces
- LDP ICCP with IPv6 neighbor node
- Multicast extension to LDP (mLDP) for IPv6 FEC with label binding through IPv4 and IPv6 transport
- Native IPv4 and IPv6 L3VPN over LDP IPv6 core
- L2VPN signaling with LDP when the nexthop address is IPv6
- IPv6 LDP CSC

Implicit IPv4 Disable

The LDP configuration model was changed with the introduction of explicit address family enabling under LDP (VRF) global and LDP (VRF) interfaces. However, in order to support backward compatibility, the old configuration model was still supported for default VRF. There was, however, no option to disable the implicitly enabled IPv4 address family under default VRF's global or interface level.

A new configuration **mpls ldp default-vrf implicit-ipv4 disable** is now available to the user to disable the implicitly enabled IPv4 address family for the default VRF. The new configuration provides a step towards migration to new configuration model for the default VRF that mandates enabling address family explicitly. This means that if the new option is configured, the user has to explicitly enable IPv4 address family for default VRF global and interface levels. It is recommended to migrate to this explicitly enabled IPv4 configuration model.

For detailed configuration steps, see Disabling Implicit IPv4, on page 94

IPv6 Label Bindings

LDP stores label bindings associated with FEC prefix in its Label Information Base (LIB) [TIB in Cisco LDP]. An entry in LIB corresponds to a prefix and holds the following bindings:

- Local binding: Local label assigned for this prefix (which is learnt through local RIB).
- Remote bindings: Array of peer labels (prefix-label bindings received in label mapping message from peer(s)).

An entry in LIB can exist due to local binding presence, or due to remote binding(s) presence, or due to both local and remote bindings presence. The forwarding setup, however, mandates that local binding be present for a prefix.

Extensions have been implemented to support IPv6 prefixes for LIB in LDP. For per-address family convergence or preference reasons, separate or new LIB is implemented to keep and maintain IPv6 prefixes. In case of dual-stack LDP, LIBv4 is preferred over LIBv6 wherever possible. For example, during background *housekeeping* function, LIBv4 is processed before LIBv6.

IPv6 Address Bindings

LDP needs to maintain IPv6 address database for local and peer interface addresses. The IPv4 address module for local/peer addresses is extended to keep IPv4/IPv6 addresses in their respective databases, much like LIB

database. In case of a dual-stack LDP, IPv4 local address database function is preferred over IPv6 local address database function where ever possible.

Default Transport Address

LDP computes default local transport address for IPv6 from its IPv6 interface or address database by picking the lowest operational loopback interface with global unicast IPv6 address. This means that any change in this loopback state or address, flaps or changes the default transport address for IPv6 and may cause session flaps using such an address as transport endpoint. For example, if a session is currently active on Loopback2 as during it's inception it was the lowest loopback with an IPv6 address, and a lower loopback, Loopback0, is configured with an IPv6 address, the session does not flap. However, if it does flap, the next time the session is attempted, Loopback0 is used.

The session flaps when configuring discovery transport address explicitly.

Use the **discovery transport-address** command under the LDP address family submode to specify the global transport address for IPv4 or IPv6.

It is recommended to configure global transport-address for IPv6 address family to avoid a potentially unstable default transport address.

LDP Control Plane: Bindings Advertisement

LDP base specification allows exchange of IPv4/IPv6 bindings (address/label) on an established session. When both IPv4 and IPv6 address families are enabled under LDP, LDP distributes address/label bindings for both address families to its established peer according to local policies. Following are a few significant points pertaining to bindings support for IPv6:

- LDP allocates/advertises local label bindings for link-local IPv6 address prefixes. If received, such FEC bindings are ignored.
- LDP sends only the Prefix FEC of the single address family type in a FEC TLV and not include both. If such a FEC binding is received, the entire message is ignored.
- LDP sends only the addresses belonging to same address family in a single address list TLV (in address or address withdraw message).

If an address family is not enabled on receiving LSR, LDP discards any bindings received from peer(s) for the address family. This means that when address family is enabled, LDP needs to reset existing sessions with the peers in order to re-learn the discarded bindings. The implementation is optimized to reset only those sessions which were previously known to be dual-stack and had sent bindings for both address families.

LSP Mapping

LDP uses IPv6 adjacency information instead of IP address to map an IPv6 link-local nexthop to an LDP peer.

In addition to other usual checks before using a label from nexthop LDP peer, LDP uses the nexthop label for a prefix of a given address family, if there are one or more LDP hello adjacencies of the same address family type established with the peer.

Label Policies

LDP allows a user to configure label policies for allocation, acceptance, receipt, and advertisement of labels for the given prefixes.

Following are the significant points pertaining to the IPv6 support for label policies:

- Label policies and their configurations are allowed under address family IPv6.
- Any policy that specifies prefix or a set of prefixes through an ACL, supports both IPv4 and IPv6 variants for address(s) or ACLs.
- Any policy that specifies peer address or set of peer addresses through an ACL, supports both IPv4 and IPv6 variant for peer address(s) or ACL.
- Any policy that specifies the peer's LSR ID in a peer ACL continues to take IPv4 ACL based policy irrespective of the feature configuration.

IS-IS

Intermediate System-to-Intermediate System (IS-IS) is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to IPv4.

Previously, IS-IS supported registration of only LDP IPv4 sync status change. This has now been enhanced to support registration of notifications of LDP IPv6 sync status change. IS-IS determines the link-metrics to be advertised based on the LDP-IGP sync status on the IPv4 and IPv6 address families.

IS-IS supports non-stop forwarding (NSF) by preserving the LDPv6-IGP sync status across high availability (HA) events of IS-IS process restarts and failover.

IS-IS also supports LDPv6-IGP sync for LFA-FRR by checking the sync status of the backup interface (if it is configured with LDP IPv6 sync).

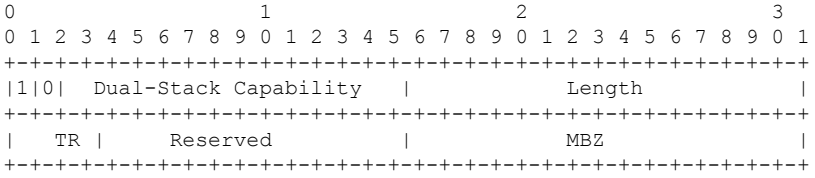
Dual-Stack Capability TLV

Clear rules are specified in RFC 5036 to determine transport connection roles in setting up a TCP connection for single-stack LDP. But RFC 5036 is not clear about dual-stack LDP, in which an LSR may assume different roles for different address families, causing issues in establishing LDP sessions.

To ensure a deterministic transport connection role for the dual-stack LDP, the dual-stack LSR conveys its transport connection preference in every LDP Hello message. This preference is encoded in a new TLV (Type Length Value) called the Dual-Stack Capability TLV. Dual-stack LSR always checks for the presence of the dual-stack capability TLV in the received LDP Hello messages and takes appropriate action for establishing or maintaining sessions.

RFC 7552 specifies more details about updates to LDP for IPv6.

Dual-Stack Capability TLV Format



Dual-Stack Capability TLV Fields

Field	Description
U and F bits	1 and 0 (as specified by RFC 5036)
Dual-Stack Capability	TLV code point (0x0701)
TR: Transport Connection Preference	TR: Transport Connection Preference: • 0100: LDPoIPv4 connection • 0110: LDPoIPv6 connection (default)
Reserved	This field is reserved. It must be set to zero on transmission and ignored on receipt
MBZ	Must be zero

Compliance Check

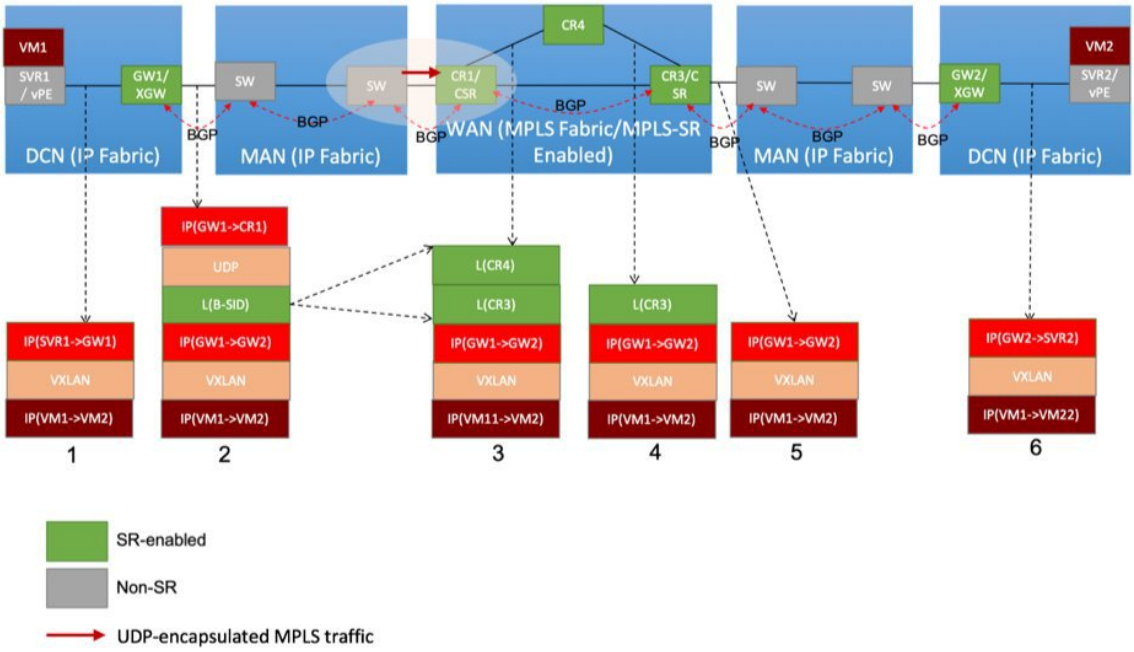
The compliance check prevents sessions being formed with prior RFC 7552 implementation of LDP IPv6.

If the dual-stack capability TLV is not present in the received Hellos and the compliance check is configured, the local and remote preferences must match to establish a session. If the preferences do not match, the LDP Hellos are dropped and the session is not established. Compliance check has therefore been disabled by default.

Use the command **neighbor dual-stack tlv-compliance** in MPLS LDP configuration to enable the compliance check.

UDP Decapsulation of MPLS-Over-UDP Traffic

You can encapsulate MPLS traffic in a UDP header (as per RFC 7510). UDP-encapsulated MPLS traffic allows better load balancing of MPLS traffic over ECMP (and LAGs) by acting as an entropy field. MPLS traffic can pass through two adjacent LSRs in an LSP, even when separated by an IP network. A metropolitan-area network (MAN) or LAN deploys this configuration, and not a WAN.



The image depicts the MAN border router **sw** sending UDP-encapsulated MPLS traffic to the WAN edge Cisco ASR 9000 Series router **CR1/CSR**. The destination IP address field contains the (peering) loopback IP address of the WAN edge router. The destination UDP port field is 6635, allocated for UDP tunnels that transport MPLS traffic. The WAN edge router removes the UDP header. Based on the MPLS label, it forwards the MPLS traffic toward the destination.

To enable the UDP decapsulation function on the ASR 9000 Series router, configure the **hw-module l3 feature mpls-over-udp-decap enable** command in global configuration mode. If you don't enable this function, the ASR 9000 Series router drops the UDP-encapsulated MPLS traffic it receives. Configuration:

```
Router# configure
Router(config)# hw-module l3 feature mpls-over-udp-decap enable
Router(config)# commit
```

Route Convergence with NHID Option for LDP in MPLS Networks

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Route Convergence with Next-Hop ID Option for LDP in MPLS Networks	Release 24.3.1	This release improves route convergence during routing changes and reduces downtime due to asynchronous Forwarding Information Base (FIB) updates by introducing Next-Hop ID (NHID) support for LDP via UDP sockets in MPLS networks. By synchronizing FIB updates across line cards and RPs, NHID ensures swift packet forwarding decisions without the need to reprogram all entries.

NHID for LDP Sessions in MPLS Networks

When there are more routes and links in a network, it can take longer for the Forwarding Information Base (FIB) to update and program new paths into the hardware. Asynchronous route updates on line cards and the Route Processor (RP) can further impact control plane connections resulting in traffic loss. This happens because the software FIB on the RP updates faster than the hardware, leading to mismatches in forwarding paths. If the time difference between these updates is too long, it can disrupt control plane connections, especially for LDP discovery hello messages used to maintain LDP sessions.

NHID is a feature that

- uses a predefined UDP socket option with NHID capability for LDP-related UDP traffic,
- improves network convergence during asynchronous FIB route updates by uniquely identifying the next hop in a routing path to expedite traffic forwarding decisions,
- minimizes traffic loss during FIB route convergence events,
- integrates with LDP to enhance the network's ability to handle increased routing scales and the demands of numerous parallel links, and
- benefits any unicast UDP control packets, such as discovery hello messages, sent to keep LDP sessions alive.

Usage Guidelines for NHID Option for UDP Sockets in MPLS

- This feature is enabled by default and you can neither disable it or configure it.
- The feature is applicable for both IPv4 and IPv6.

Verify NHID Options for UDP Sockets in MPLS

To verify the NHID option for UDP sockets is supported for LDP control packets, you can check the LDP operational parameters and identify specific UDP sockets with port 646, and then verify if the NHID option is enabled for those sockets.

Procedure

Step 1 Verify the NHID option for UDP sockets is supported for LDP control packets.

Example:

```
Router#show mpls ldp parameters
LDP Parameters:
  Role: Active
  Protocol Version: 1
  Router ID: 37.0.0.1
  Null Label:
    IPv4: Implicit
  Session:
    Hold time: 180 sec
    Keepalive interval: 60 sec
    Backoff: Initial:15 sec, Maximum:120 sec
    Global MD5 password: Disabled
  Discovery:
    Link Hellos:     Holdtime:15 sec, Interval:5 sec
    Targeted Hellos: Holdtime:45 sec, Interval:15 sec
                     Accepting:
                       IPv4: all;
    Quick-start: Enabled (by default)
    Transport address:
      IPv4: 37.0.0.1
  Graceful Restart:
    Enabled
    Reconnect Timeout:120 sec, Forwarding State Holdtime:180 sec
  NSR: Enabled, Not Sync-ed
  Timeouts:
    Housekeeping periodic timer: 10 sec
    Local binding: 300 sec
    Forwarding state in LSD: 360 sec
  Delay in AF Binding Withdrawl from peer: 180 sec
  Max:
    5000 interfaces (4000 attached, 1000 TE tunnel), 2000 peers
  OOR state
    Memory: Normal
  IGP sync delay:
    Interface up: 60 sec
  MPLS NHID Support: Yes
```

Step 2 Identify the UDP sockets with port 646 for LDP control packets from a list of all the UDP sockets.

Example:

```
Router#show udp brief
   PCB     VRF-ID     Recv-Q Send-Q  Local Address          Foreign Address
0x7fe448003420 0x00000000      0      0  :::0                   :::0
0x7fe448001c50 0x60000000      0      0  0.0.0.0:68             0.0.0.0:0
0x7fe448000f90 0x00000000      0      0  0.0.0.0:68             0.0.0.0:0
0x7fe448002c70 0x60000000      0      0  0.0.0.0:646            0.0.0.0:0
```

```
0x7fe448002400 0x00000000      0      0  0.0.0.0:646            0.0.0.0:0
```

Step 3 Verify the NHID option is enabled for a specific UDP socket with port 646.

Example:

```
Router#show udp detail pcb 0x7fe448002c70
Fri Oct  6 15:46:40.359 UTC

================================================
PCB is 0x7fe448002c70 SO 0x7fe448001f50, Family: 2, VRF: 0x60000000
Local host: 0.0.0.0:646 (Local App PID: 2698)
Foreign host: 0.0.0.0:0
Current send queue size: 0
Current receive queue size: 0
PCB flags: RECVDSTADDR, RECVIF
PCB Transport flags: MPLS_NHID_LBL
Traffic Class: 192
```

How to Implement MPLS LDP

A typical MPLS LDP deployment requires coordination among several global neighbor routers. Various configuration tasks are required to implement MPLS LDP :

Configuring LDP Discovery Parameters

Perform this task to configure LDP discovery parameters (which may be crucial for LDP operations).

Note The LDP discovery mechanism is used to discover or locate neighbor nodes.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **discovery** { **hello** | **targeted-hello** } **holdtime** *seconds*
5. **discovery** { **hello** | **targeted-hello** } **interval** *seconds*
6. **commit**
7. (Optional) **show mpls ldp** [**vrf** *vrf-name*] **parameters**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface IP address. By default, LDP uses the global router ID (configured by the global router ID process).

Step 4	**discovery** { **hello** | **targeted-hello** } **holdtime** *seconds*
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# discovery hello holdtime 30`
	`RP/0/RSP0/CPU0:router(config-ldp)# discovery targeted-hello holdtime 180`
		Specifies the time that a discovered neighbor is kept without receipt of any subsequent hello messages. The default value for the *seconds* argument is 15 seconds for link hello and 90 seconds for targeted hello messages.

Step 5	**discovery** { **hello** | **targeted-hello** } **interval** *seconds*
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# discovery hello interval 15`
	`RP/0/RSP0/CPU0:router(config-ldp)# discovery targeted-hello interval 20`
		Selects the period of time between the transmission of consecutive hello messages. The default value for the *seconds* argument is 5 seconds for link hello messages and 10 seconds for targeted hello messages.

Step 6	**commit**	

Step 7	(Optional) **show mpls ldp** [**vrf** *vrf-name*] **parameters**
	Example:
	`RP/0/RSP0/CPU0:router # show mpls ldp parameters`
	`RP/0/RSP0/CPU0:router # show mpls ldp vrf red parameters`
		Displays all the current MPLS LDP parameters.
		Displays the LDP parameters for the specified VRF.

Related Topics

LDP Control Plane, on page 3

Configure Label Distribution Protocol Targeted Neighbor

LDP session between LSRs that are not directly connected is known as targeted LDP session. For LDP neighbors which are not directly connected, you must manually configure the LDP neighborship on both the routers.

Configuration Example

This example shows how to configure LDP for non-directly connected routers.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# mpls ldp
RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.0.2.1
RP/0/RSP0/CPU0:router(config-ldp)# neighbor 198.51.100.1:0 password encrypted 13061E010803
RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4
RP/0/RSP0/CPU0:router(config-ldp-af)# discovery targeted-hello accept
RP/0/RSP0/CPU0:router(config-ldp-af)# neighbor 198.51.100.1 targeted
RP/0/RSP0/CPU0:router(config-ldp-af)# commit
```

Running Configuration

This section shows the LDP targeted neighbor running configuration.

```
mpls ldp
router-id 192.0.2.1
neighbor 198.51.100.1:0 password encrypted 13061E010803
address-family ipv4
  discovery targeted-hello accept
  neighbor 198.51.100.1 targeted
!
```

Verification

Verify LDP targeted neighbor configuration.

```
RP/0/RSP0/CPU0:router#show mpls ldp discovery
Wed Nov 28 04:30:31.862 UTC

Local LDP Identifier: 192.0.2.1:0
Discovery Sources:
  Targeted Hellos:  <<< targeted hellos based session
    192.0.2.1 -> 198.51.100.1(active/passive), xmit/recv   <<< both transmit and receive
of targeted hellos between the neighbors
      LDP Id: 198.51.100.1:0
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: Nov 28 04:19:55.340 (00:10:36 ago)

RP/0/RSP0/CPU0:router#show mpls ldp neigbhor
Wed Nov 28 04:30:38.272 UTC

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:0:13183 - 192.0.2.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
```

```
  State: Oper; Msgs sent/rcvd: 20/20; Downstream-Unsolicited
  Up time: 00:10:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active/passive) <<< targeted LDP based
session
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (4)
     198.51.100.1        10.0.0.1       172.16.0.1       192.168.0.1
    IPv6: (0)
```

Configuring LDP Discovery Over a Link

Perform this task to configure LDP discovery over a link.

Note There is no need to enable LDP globally.

Before you begin

A stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **interface** *type interface-path-id*
5. **commit**
6. (Optional) **show mpls ldp discovery**
7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:**	Enters MPLS LDP configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# mpls ldp`	
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface name or IP address. By default, LDP uses the global router ID (configured by the global router ID process).

Step 4	**interface** *type interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# interface tunnel-te 12001`
	`RP/0/RSP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.

Step 5	**commit**	
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery`	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:**	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Related Topics

LDP Control Plane, on page 3
Configuring LDP Link: Example, on page 98

Configuring LDP Discovery for Active Targeted Hellos

Perform this task to configure LDP discovery for active targeted hellos.

Note The active side for targeted hellos initiates the unicast hello toward a specific destination.

Before you begin

These prerequisites are required to configure LDP discovery for active targeted hellos:

- Stable router ID is required at either end of the targeted session. If you do not assign a router ID to the routers, the system will default to the global router ID. Please note that default router IDs are subject to change and may cause an unstable discovery.
- One or more MPLS Traffic Engineering tunnels are established between non-directly connected LSRs.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **interface** *type interface-path-id*
5. **commit**
6. (Optional) **show mpls ldp discovery**
7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. In Cisco IOS XR software, the router ID is specified as an interface name or IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol.
Step 5	**commit**	
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery`	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:**	Displays the brief status of the LDP discovery process for all VRFs.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery brief`	
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Related Topics

LDP Control Plane, on page 3
Configuring LDP Discovery for Targeted Hellos: Example, on page 98

Configuring LDP Discovery for Passive Targeted Hellos

Perform this task to configure LDP discovery for passive targeted hellos.

A passive side for targeted hello is the destination router (tunnel tail), which passively waits for an incoming hello message. Because targeted hellos are unicast, the passive side waits for an incoming hello message to respond with hello toward its discovered neighbor.

Before you begin

Stable router ID is required at either end of the link to ensure that the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **discovery targeted-hello accept**
5. **commit**
6. (Optional) **show mpls ldp discovery**
7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).
Step 4	**discovery targeted-hello accept** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# discovery targeted-hello accept`	Directs the system to accept targeted hello messages from any source and activates passive mode on the LSR for targeted hello acceptance. • This command is executed on the receiver node (with respect to a given MPLS TE tunnel). • You can control the targeted-hello acceptance using the **discovery targeted-hello accept** command.
Step 5	**commit**	
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery`	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.

	Command or Action	Purpose
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Related Topics

LDP Control Plane, on page 3
Configuring LDP Discovery for Targeted Hellos: Example, on page 98

Configuring Label Advertisement Control (Outbound Filtering)

Perform this task to configure label advertisement (outbound filtering).

By default, a label switched router (LSR) advertises all incoming label prefixes to each neighboring router. You can control the exchange of label binding information using the **mpls ldp label advertise** command. Using the optional keywords, you can advertise selective prefixes to all neighbors, advertise selective prefixes to defined neighbors, or disable label advertisement to all peers for all prefixes.

Note Prefixes and peers advertised selectively are defined in the access list.

Before you begin

Before configuring label advertisement, enable LDP and configure an access list.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
4. **label local advertise** [**to** *ldp-id* **for** *prefix-acl* | **interface** *type interface-path-id*]
5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	[**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
		(Optional) Specifies a non-default VRF.
		Enables the LDP IPv4 or IPv6 address family.

Step 4	**label local advertise** [**to** *ldp-id* **for** *prefix-acl* | **interface** *type interface-path-id*]
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp-af)# label local advertise to 10.0.0.1:0 for pfx_acl1`
	`RP/0/RSP0/CPU0:router(config-ldp-af)# label local advertise interface POS 0/1/0/0`
		Configures outbound label advertisement control by specifying one of the following options:
		interface
		Specifies an interface for label advertisement.
		to ***ldp-id***
		for ***prefix-acl***
		Specifies neighbors to advertise and receive label advertisements.

Step 5	**commit**	

Related Topics

Label Advertisement Control (Outbound Filtering), on page 9
Configuring Label Advertisement (Outbound Filtering): Example, on page 99

Setting Up LDP Neighbors

Perform this task to set up LDP neighbors.

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6** }
5. **discovery transport-address** [*ip-address* | **interface**]
6. **exit**
7. **holdtime** *seconds*
8. [**vrf** *vrf-name*] **neighbor** *ldp-id* **password** [**encrypted**] *password*
9. **backoff** *initial maximum*
10. **commit**
11. (Optional) **show mpls ldp neighbor**
12. (Optional) **show mpls ldp vrf** *vrf-name* **neighbor**
13. (Optional) **show mpls ldp vrf all neighbor brief**
14. (Optional) **clear mpls ldp neighbor**
15. (Optional) **clear mpls ldp vrf all neighbor**
16. (Optional) **clear mpls ldp vrf** *vrf-name* **neighbor**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.

Step 4	**address-family** {**ipv4**	**ipv6** }
		Example:
		`RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv4`
		or
		`RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv6`
			Enables the LDP IPv4 or IPv6 address family.

	Command or Action	Purpose

Step 5	**discovery transport-address** [*ip-address* | **interface**]
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp-if-af)# discovery transport-address 192.168.1.42`
	`RP/0/RSP0/CPU0:router(config-ldp-if-af)# discovery transport-address 5:6::78`
	or
	`RP/0/RSP0/CPU0:router(config-ldp-if-af)# discovery transport-address interface`
		Provides an alternative transport address for a TCP connection.
		• Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.
		• Transport address configuration is applied for a given LDP-enabled interface.
		• If the interface version of the command is used, the configured IP address of the interface is passed to its neighbors as the transport address.

Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 7	**holdtime** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# holdtime 30`	Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer. • Outgoing keepalive interval is adjusted accordingly (to make three keepalives in a given holdtime) with a change in session holdtime value. • Session holdtime is also exchanged when the session is established. • In this example holdtime is set to 30 seconds, which causes the peer session to timeout in 30 seconds, as well as transmitting outgoing keepalive messages toward the peer every 10 seconds.
Step 8	[**vrf** *vrf-name*] **neighbor** *ldp-id* **password** [**encrypted**] *password* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# neighbor 192.168.2.44:0 password secretpasswd`	(Optional) Specifies a non-default VRF. Configures password authentication (using the TCP MD5 option) for a given neighbor.
Step 9	**backoff** *initial maximum* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# backoff 10 20`	Configures the parameters for the LDP backoff mechanism. The LDP backoff mechanism prevents two incompatibly configured LSRs from engaging in an unthrottled sequence of session setup failures. If a session setup attempt fails due to such incompatibility, each LSR delays its next attempt (backs off), increasing the delay exponentially with each successive failure until the maximum backoff delay is reached.
Step 10	**commit**	

	Command or Action	Purpose
Step 11	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp neighbor`	Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.
Step 12	(Optional) **show mpls ldp vrf** *vrf-name* **neighbor** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red neighbor`	Displays the status of the LDP session with its neighbors for the specified VRF. This command can be run with the brief option.
Step 13	(Optional) **show mpls ldp vrf all neighbor brief** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all neighbor brief`	Displays the brief LDP session neighbor information for all VRFs.
Step 14	(Optional) **clear mpls ldp neighbor** **Example:** `RP/0/RSP0/CPU0:router# clear mpls ldp neighbor`	Resets an LDP session.
Step 15	(Optional) **clear mpls ldp vrf all neighbor** **Example:** `RP/0/RSP0/CPU0:router# clear mpls ldp vrf all neighbor`	Resets LDP session for all VRFs.
Step 16	(Optional) **clear mpls ldp vrf** *vrf-name* **neighbor** **Example:** `RP/0/RSP0/CPU0:router# clear mpls ldp vrf red neighbor`	Resets LDP session for the specified VRF.

Related Topics

Configuring LDP Neighbors: Example, on page 100

Setting Up LDP Forwarding

Perform this task to set up LDP forwarding.

By default, the LDP control plane implements the penultimate hop popping (PHOP) mechanism. The PHOP mechanism requires that label switched routers use the implicit-null label as a local label for the given Forwarding Equivalence Class (FEC) for which LSR is the penultimate hop. Although PHOP has certain advantages, it may be required to extend LSP up to the ultimate hop under certain circumstances (for example, to propagate MPL QoS). This is done using a special local label (explicit-null) advertised to the peers after which the peers use this label when forwarding traffic toward the ultimate hop (egress LSR).

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **address-family** {**ipv4** | **ipv6** }
4. **label local advertise explicit-null**
5. **commit**
6. (Optional) **show mpls ldp forwarding**
7. (Optional) **show mpls ldp vrf all forwarding**
8. (Optional) **show mpls ldp vrf all forwarding summary**
9. (Optional) **show mpls ldp vrf** *vrf-name* **ipv4 forwarding**
10. (Optional) **show mpls ldp forwarding summary all**
11. (Optional) **clear mpls ldp vrf** *vrf-name* **ipv4 forwarding**
12. (Optional) **clear mpls ldp** [**ipv4** | **ipv6**]**forwarding**
13. (Optional) **show mpls ldp afi-all forwarding**
14. (Optional) **show mpls ldp ipv6 forwarding**
15. (Optional) **show mpls forwarding**
16. (Optional) **ping** *ip-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	[**vrf** *vrf-name*] **address-family** {**ipv4**	**ipv6** }
		Example:
		`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`
		or
		`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
			(Optional) Specifies a non-default VRF.
			Enables the LDP IPv4 or IPv6 address family.

	Command or Action	Purpose
Step 4	**label local advertise explicit-null** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# label local advertise explicit-null`	Causes a router to advertise an explicit null label in situations where it normally advertises an implicit null label (for example, to enable an ultimate-hop disposition instead of PHOP).
Step 5	**commit**	
Step 6	(Optional) **show mpls ldp forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp forwarding`	Displays the MPLS LDP view of installed forwarding states (rewrites). **Note** For local labels, only up to 12000 rewrites are supported. If the rewrites exceed this limit, MPLS LSD or MPLS LDP or both the processes may crash.
Step 7	(Optional) **show mpls ldp vrf all forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all forwarding`	Displays the forwarding setup information of all LDP configured VRFs.
Step 8	(Optional) **show mpls ldp vrf all forwarding summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all forwarding summary`	Displays the forwarding setup summary of all LDP configured VRFs.
Step 9	(Optional) **show mpls ldp vrf** *vrf-name* **ipv4 forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red ipv4 forwarding`	Displays the forwarding setup information for the specified VRF for IPv4.
Step 10	(Optional) **show mpls ldp forwarding summary all** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp forwarding summary all`	Displays the aggregate summary across LDP processes and all VRFs.
Step 11	(Optional) **clear mpls ldp vrf** *vrf-name* **ipv4 forwarding** **Example:** `RP/0/RSP0/CPU0:router# clear mpls ldp vrf red ipv4 forwarding`	Resets the MPLS forwarding rewrites for the specified VRF for IPv4.
Step 12	(Optional) **clear mpls ldp** [**ipv4** | **ipv6**]**forwarding** **Example:**	Resets the MPLS forwarding rewrites for either IPv4 or IPv6 addresses.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router# clear mpls ldp ipv4 forwarding`
	or
	`RP/0/RSP0/CPU0:router# clear mpls ldp ipv6 forwarding`	

Step 13	(Optional) **show mpls ldp afi-all forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp afi-all forwarding`	Displays the forwarding setup information of all address families.
Step 14	(Optional) **show mpls ldp ipv6 forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp ipv6 forwarding`	Displays the MPLS LDP view of installed forwarding states (rewrites) for IPv6.
Step 15	(Optional) **show mpls forwarding** **Example:** `RP/0/RSP0/CPU0:router# show mpls forwarding`	Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).
Step 16	(Optional) **ping** *ip-address* **Example:** `RP/0/RSP0/CPU0:router# ping 192.168.2.55`	Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the **show mpls forwarding** command).

Related Topics

LDP Forwarding, on page 4
Configuring LDP Forwarding: Example, on page 100

Configuring Global Transport Address

Perform this task to configure global transport address for the IPv4 address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **address-family ipv4**
4. **discovery transport-address** *ip-address*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`	Enables LDP IPv4 address family.
Step 4	**discovery transport-address** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# discovery transport-address 192.168.1.42`	Provides an alternative transport address for a TCP connection. • Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Setting Up LDP NSF Using Graceful Restart

Perform this task to set up NSF using LDP graceful restart.

LDP graceful restart is a way to enable NSF for LDP. The correct way to set up NSF using LDP graceful restart is to bring up LDP neighbors (link or targeted) with additional configuration related to graceful restart.

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **exit**
5. **graceful-restart**
6. **graceful-restart forwarding-state-holdtime** *seconds*
7. **graceful-restart reconnect-timeout** *seconds*
8. **commit**
9. (Optional) **show mpls ldp** [**vrf** *vrf-name*] **parameters**
10. (Optional) **show mpls ldp neighbor**
11. (Optional) **show mpls ldp graceful-restart**
12. (Optional) **show mpls ldp vrf all graceful-restart**
13. (Optional) **show mpls ldp vrf** *vrf-name* **graceful-restart**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`
	`RP/0/RSP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol.

Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.

	Command or Action	Purpose
Step 5	**graceful-restart** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart`	Enables the LDP graceful restart feature.
Step 6	**graceful-restart forwarding-state-holdtime** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart forwarding-state-holdtime 180`	Specifies the length of time that forwarding can keep LDP-installed forwarding states and rewrites, and specifies wh en the LDP control plane restarts. • After restart of the control plane, when the forwarding state holdtime expires, any previously installed LDP forwarding state or rewrite that is not yet refreshed is deleted from the forwarding. • Recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer.
Step 7	**graceful-restart reconnect-timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169`	Specifies the length of time a neighbor waits before restarting the node to reconnect before declaring an earlier graceful restart session as down. This command is used to start a timer on the peer (upon a neighbor restart). This timer is referred to as *Neighbor Liveness* timer.
Step 8	**commit**	

Step 9	(Optional) **show mpls ldp** [**vrf** *vrf-name*] **parameters**
	Example:
	`RP/0/RSP0/CPU0:router # show mpls ldp parameters`
	`RP/0/RSP0/CPU0:router # show mpls ldp vrf red parameters`
		Displays all the current MPLS LDP parameters.
		Displays the LDP parameters for the specified VRF.

Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp neighbor`	Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.
Step 11	(Optional) **show mpls ldp graceful-restart** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp graceful-restart`	Displays the status of the LDP graceful restart feature. The output of this command not only shows states of different graceful restart timers, but also a list of graceful restart neighbors, their state, and reconnect count.

	Command or Action	Purpose
Step 12	(Optional) **show mpls ldp vrf all graceful-restart** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all graceful-restart`	Displays the status of the LDP graceful restart for all VRFs.
Step 13	(Optional) **show mpls ldp vrf** *vrf-name* **graceful-restart** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red graceful-restart`	Displays the status of the LDP graceful restart for the specified VRF.

Related Topics

LDP Graceful Restart, on page 5
Phases in Graceful Restart, on page 7
Recovery with Graceful-Restart, on page 7
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 100

Configuring Label Acceptance Control (Inbound Filtering)

Perform this task to configure LDP inbound label filtering.

Note By default, there is no inbound label filtering performed by LDP and thus an LSR accepts (and retains) all remote label bindings from all peers.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **label accept for** *prefix-acl* **from** *ip-address*
4. [**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
5. **label remote accept from** *ldp-id* **for** *prefix-acl*
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:**	Enters the MPLS LDP configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# mpls ldp`	

Step 3	**label accept for** *prefix-acl* **from** *ip-address*
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1`
	`RP/0/RSP0/CPU0:router(config-ldp)# label accept for pfx_acl_2 from 192.168.2.2`
		Configures inbound label acceptance for prefixes specified by prefix-acl from neighbor (as specified by its IP address).

Step 4	[**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
		(Optional) Specifies a non-default VRF.
		Enables the LDP IPv4 or IPv6 address family.

Step 5	**label remote accept from** *ldp-id* **for** *prefix-acl* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# label remote accept from 192.168.1.1:0 for pfx_acl_1`	Configures inbound label acceptance control for prefixes specified by prefix-acl from neighbor (as specified by its LDP ID).
Step 6	**commit**	

Related Topics

Label Acceptance Control (Inbound Filtering), on page 9
Configuring Label Acceptance (Inbound Filtering): Example, on page 101

Configuring Local Label Allocation Control

Perform this task to configure label allocation control.

Note By default, local label allocation control is disabled and all non-BGP prefixes are assigned local labels.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
4. **label local allocate for** *prefix-acl*

5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	[**vrf** *vrf-name*] **address-family** { **ipv4** | **ipv6**}
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
		(Optional) Specifies a non-default VRF.
		Enables the LDP IPv4 or IPv6 address family.

Step 4	**label local allocate for** *prefix-acl* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# label local allocate for pfx_acl_1`	Configures label allocation control for prefixes as specified by prefix-acl.
Step 5	**commit**	

Related Topics

Local Label Allocation Control, on page 9
Configuring Local Label Allocation Control: Example, on page 101

Configuring Session Protection

Perform this task to configure LDP session protection.

By default, there is no protection is done for link sessions by means of targeted hellos.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **session protection** [**for** *peer-acl*] [**duration** *seconds*]
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**session protection** [**for** *peer-acl*] [**duration** *seconds*] **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# session protection for peer_acl_1 duration 60`	Configures LDP session protection for peers specified by peer-acl with a maximum duration, in seconds.
Step 4	**commit**	

Related Topics

Session Protection, on page 10
Configuring LDP Session Protection: Example, on page 101

Configuring LDP IGP Synchronization: OSPF

Perform this task to configure LDP IGP Synchronization under OSPF.

Note By default, there is no synchronization between LDP and IGPs.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. (Optional) **vrf** *vrf-name*
4. Use one of the following commands:
   - **mpls ldp sync**
   - **area** *area-id* **mpls ldp sync**
   - **area** *area-id* **interface** *name* **mpls ldp sync**
5. (Optional) Use one of the following commands:
   - **mpls ldp sync**
   - **area** *area-id* **mpls ldp sync**
   - **area** *area-id* **interface** *name* **mpls ldp sync**

6. **commit**
7. (Optional) **show mpls ldp vrf** *vrf-name* **ipv4 igp sync**
8. (Optional) **show mpls ldp vrf all ipv4 igp sync**
9. (Optional) **show mpls ldp** { **ipv4** | **ipv6** }**igp sync**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router ospf** *process-name* **Example:** `RP/0/RSP0/CPU0:router(config)# router ospf 100`	Identifies the OSPF routing process and enters OSPF configuration mode.
Step 3	(Optional) **vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# vrf red`	Specifies the non-default VRF.
Step 4	Use one of the following commands: • **mpls ldp sync** • **area** *area-id* **mpls ldp sync** • **area** *area-id* **interface** *name* **mpls ldp sync** **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# mpls ldp sync`	Enables LDP IGP synchronization on an interface.

Step 5	(Optional) Use one of the following commands:
	• **mpls ldp sync**
	• **area** *area-id* **mpls ldp sync**
	• **area** *area-id* **interface** *name* **mpls ldp sync**
	Example:
	`RP/0/RSP0/CPU0:router(config-ospf-vrf)# mpls ldp sync`
	`RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 1 mpls ldp sync`
	`RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 1 interface POS 0/2/0/0 mpls ldp sync`
		Enables LDP IGP synchronization on an interface for the specified VRF.

	Command or Action	Purpose
Step 6	**commit**	
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **ipv4 igp sync** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red ipv4 igp sync`	Displays the LDP IGP synchronization information for the specified VRF for address family IPv4.
Step 8	(Optional) **show mpls ldp vrf all ipv4 igp sync** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 igp sync`	Displays the LDP IGP synchronization information for all VRFs for address family IPv4.

Step 9	(Optional) **show mpls ldp** { **ipv4** | **ipv6** }**igp sync**
	Example:
	`RP/0/RSP0/CPU0:router# show mpls ldp ipv4 igp sync`
	`RP/0/RSP0/CPU0:router# show mpls ldp ipv6 igp sync`
		Displays the LDP IGP synchronization information for IPv4 or IPv6 address families.

Related Topics

IGP Synchronization, on page 11
Configuring LDP IGP Synchronization—OSPF: Example, on page 102

Disabling LDP IGP Synchronization: OSPF

Perform this task to disable LDP IGP Synchronization under OSPF.

You can disable LDP IGP synchronization on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. (Optional) **vrf** *vrf-name*
4. Use one of the following commands:
   - **area** *area-id* **mpls ldp sync disable**
   - **area** *area-id* **interface** *name* **mpls ldp sync disable**
5. (Optional) Use one of the following commands:
   - **area** *area-id* **mpls ldp sync disable**
   - **area** *area-id* **interface** *name* **mpls ldp sync disable**

6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router ospf** *process-name* **Example:** `RP/0/RSP0/CPU0:router(config)# router ospf 109`	Identifies the OSPF routing process and enters OSPF configuration mode.
Step 3	(Optional) **vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# vrf red`	Specifies the non-default VRF.

Step 4	Use one of the following commands:
	• **area** *area-id* **mpls ldp sync disable**
	• **area** *area-id* **interface** *name* **mpls ldp sync disable**
	Example:
	`RP/0/RSP0/CPU0:router(config-ospf)# area 1 mpls ldp sync disable`
	`RP/0/RSP0/CPU0:router(config-ospf)# area 1 interface POS 0/2/0/0 mpls ldp sync disable`
		Disables LDP IGP synchronization on an interface.

Step 5	(Optional) Use one of the following commands:
	• **area** *area-id* **mpls ldp sync disable**
	• **area** *area-id* **interface** *name* **mpls ldp sync disable**
	Example:
	`RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 1 mpls ldp sync disable`
	`RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 1 interface POS 0/2/0/0 mpls ldp sync disable`
		Disables LDP IGP synchronization on an interface for the specified VRF.

Step 6	**commit**	

Configuring LDP IGP Synchronization: ISIS

Perform this task to configure LDP IGP Synchronization under ISIS.

Note By default, there is no synchronization between LDP and ISIS.

SUMMARY STEPS

1. **configure**
2. **router isis** *instance-id*
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **mpls ldp sync**
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**router isis** *instance-id*
	Example:
	`RP/0/RSP0/CPU0:router(config)# router isis 100`
	`RP/0/RSP0/CPU0:router(config-isis)#`
		Enables the Intermediate System-to-Intermediate System (IS-IS) routing protocol and defines an IS-IS instance.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-isis)# interface POS 0/2/0/0`
	`RP/0/RSP0/CPU0:router(config-isis-if)#`
		Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.

Step 4	**address-family** {**ipv4**	**ipv6**} **unicast**
		Example:
		`RP/0/RSP0/CPU0:router(config-isis-if)#`
		`address-family ipv4 unicast`
		`RP/0/RSP0/CPU0:router(config-isis-if-af)#`
		
		`RP/0/RSP0/CPU0:router(config-isis-if)#`
		`address-family ipv6 unicast`
		`RP/0/RSP0/CPU0:router(config-isis-if-af)#`
			Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) or IP version 6 (IPv6) address prefix.

	Command or Action	Purpose
Step 5	**mpls ldp sync** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if-af)# mpls ldp sync`	Enables LDP IGP synchronization.
Step 6	**commit**	

Related Topics

IGP Synchronization, on page 11
Configuring LDP IGP Synchronization—ISIS: Example, on page 102

Enabling LDP Auto-Configuration for a Specified OSPF Instance

Perform this task to enable IGP auto-configuration globally for a specified OSPF process name.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

Note This feature is supported for IPv4 unicast family in default VRF only.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **mpls ldp auto-config**
4. **area** *area-id*
5. **interface** *type interface-path-id*
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# router ospf 190`
	`RP/0/RSP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

	Command or Action	Purpose
Step 3	**mpls ldp auto-config** **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# mpls ldp auto-config`	Enables LDP auto-configuration.
Step 4	**area** *area-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# area 8`	Configures an OSPF area and identifier. ***area-id*** Either a decimal value or an IP address.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`	Enables LDP auto-configuration on the specified interface. **Note** LDP configurable limit for maximum number of interfaces does not apply to IGP auto-configuration interfaces.
Step 6	**commit**	

Related Topics

IGP Auto-configuration, on page 11
Configuring LDP Auto-Configuration: Example, on page 102
Disabling LDP Auto-Configuration, on page 63

Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance

Perform this task to enable IGP auto-configuration in a defined area with a specified OSPF process name.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

Note This feature is supported for IPv4 unicast family in default VRF only.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **area** *area-id*
4. **mpls ldp auto-config**
5. **interface** *type interface-path-id*
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# router ospf 100`
	`RP/0/RSP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

Step 3	**area** *area-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-ospf)# area 8`
	`RP/0/RSP0/CPU0:router(config-ospf-ar)#`
		Configures an OSPF area and identifier.
		area-id
		Either a decimal value or an IP address.

Step 4	**mpls ldp auto-config** **Example:** `RP/0/RSP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config`	Enables LDP auto-configuration.

Step 5	**interface** *type interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`
	`RP/0/RSP0/CPU0:router(config-ospf-ar-if)`
		Enables LDP auto-configuration on the specified interface. The LDP configurable limit for maximum number of interfaces does not apply to IGP auto-config interfaces.

Step 6	**commit**	

Related Topics

IGP Auto-configuration, on page 11
Configuring LDP Auto-Configuration: Example, on page 102
Disabling LDP Auto-Configuration, on page 63

Disabling LDP Auto-Configuration

Perform this task to disable IGP auto-configuration.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**

3. **interface** *type interface-path-id*
4. **igp auto-config disable**
5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**mpls ldp**
	Example:
	`RP/0/RSP0/CPU0:router(config)# mpls ldp`
	`RP/0/RSP0/CPU0:router(config-ldp)#`
		Enters the MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface pos 0/6/0/0`	Enters interface configuration mode and configures an interface.
Step 4	**igp auto-config disable** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if)# igp auto-config disable`	Disables auto-configuration on the specified interface.
Step 5	**commit**	

Related Topics

IGP Auto-configuration, on page 11
Configuring LDP Auto-Configuration: Example, on page 102

Configuring LDP Nonstop Routing

Perform this task to configure LDP NSR.

Note By default, NSR is globally-enabled on all LDP sessions except AToM.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **nsr**

4. **commit**
5. (Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr statistics**
6. (Optional) **show mpls ldp vrf** *vrf-name* **nsr statistics neighbor**
7. (Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr summary**
8. (Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr pending**
9. (Optional) **show mpls ldp vrf** *vrf-name* **nsr pending neighbor**
10. (Optional) **show mpls ldp vrf all nsr summary**
11. (Optional) **show mpls ldp nsr summary all**
12. (Optional) **clear mpls ldp vrf** *vrf-name* **nsr statistics neighbor**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**nsr** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# nsr`	Enables LDP nonstop routing.
Step 4	**commit**	

Step 5	(Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr statistics**
	Example:
	`RP/0/RSP0/CPU0:router# show mpls ldp nsr statistics`
	`RP/0/RSP0/CPU0:router# show mpls ldp vrf red nsr statistics`
		Displays MPLS LDP NSR statistics.
		Displays LDP NSR statistics for the specified VRF.

Step 6	(Optional) **show mpls ldp vrf** *vrf-name* **nsr statistics neighbor** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red nsr statistics neighbor 10.0.0.1`	Displays LDP NSR statistics for the specified VRF for a given neighbor.
Step 7	(Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr summary** **Example:**	Displays MPLS LDP NSR summarized information. Displays LDP NSR summarized information for the specified VRF.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router# show mpls ldp nsr summary`
	
	`RP/0/RSP0/CPU0:router# show mpls ldp vrf red nsr summary`	

Step 8	(Optional) **show mpls ldp** [**vrf** *vrf-name*] **nsr pending**
	Example:
	`RP/0/RSP0/CPU0:router# show mpls ldp nsr pending`
	
	`RP/0/RSP0/CPU0:router# show mpls ldp vrf red nsr pending`
		Displays MPLS LDP NSR pending information.
		Displays LDP NSR pending information for the specified VRF.

Step 9	(Optional) **show mpls ldp vrf** *vrf-name* **nsr pending neighbor** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf red nsr pending neighbor 172.16.0.1`	Displays LDP NSR pending information for the specified VRF for a given neighbor.
Step 10	(Optional) **show mpls ldp vrf all nsr summary** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp vrf all nsr summary`	Displays all LDP configured VRF (including default VRF) summarized information.
Step 11	(Optional) **show mpls ldp nsr summary all** **Example:** `RP/0/RSP0/CPU0:router# show mpls ldp nsr summary all`	Displays aggregate summary across LDP processes and all VRFs.
Step 12	(Optional) **clear mpls ldp vrf** *vrf-name* **nsr statistics neighbor** **Example:** `RP/0/RSP0/CPU0:router# clear mpls ldp vrf red nsr statistics neighbor`	Resets LDP NSR statistics for the specified VRF for neighbor.

Related Topics

LDP Nonstop Routing, on page 12

Configuring LDP Downstream on Demand mode

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name* **session**] **downstream-on-demand**
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name* **session**] **downstream-on-demand** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# vrf red session downstream-on-demand with ABC`	(Optional) Enters downstream on demand label advertisement mode under the specified non-default VRF. Enters downstream on demand label advertisement mode. The ACL contains the list of peer IDs that are configured for downstream-on-demand mode. When the ACL is changed or configured, the list of established neighbor is traversed.
Step 4	**commit**	

Related Topics

Downstream on Demand, on page 14

Setting Up Implicit-Null-Override Label

Perform this task to configure implicit-null label for non-egress prefixes.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **address-family** {**ipv4** | **ipv6** }
4. **label**
5. **local implicit-null-override for** *access-list*
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	[**vrf** *vrf-name*] **address-family** {**ipv4** | **ipv6** }
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4`
	or
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
		(Optional) Specifies a non-default VRF.
		Enables the LDP IPv4 or IPv6 address family.

Step 4	**label** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# label`	Configures the allocation, advertisement ,and acceptance of labels.
Step 5	**local implicit-null-override for** *access-list* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af-lbl)# local implicit-null-override for 70`	Configures implicit-null local label for non-egress prefixes. **Note** This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.
Step 6	**commit**	

Redistributing MPLS LDP Routes into BGP

Perform this task to redistribute Border Gateway Protocol (BGP) autonomous system into an MPLS LDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **redistribute bgp**
4. **end** or **commit**
5. **show run mpls ldp**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**redistribute bgp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# redistribute bgp advertise-to acl_1`	Allows the redistribution of BGP routes into an MPLS LDP processes. **Note** Autonomous system numbers (ASNs) are globally unique identifiers used to identify autonomous systems (ASs) and enable ASs to exchange exterior routing information between neighboring ASs. A unique ASN is allocated to each AS for use in BGP routing. ASNs are encoded as 2-byte numbers and 4-byte numbers in BGP.
Step 4	**end** or **commit**	• When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command or Action	Purpose
Step 5	**show run mpls ldp** **Example:** `RP/0/RSP0/CPU0:router# show run mpls ldp`	Displays information about the redistributed route information.

Enabling MLDP

Perform this task to enable Multicast Label Distribution Protocol (MLDP) in MPLS LDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**mldp**
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# mldp`
	`RP/0/RSP0/CPU0:router(config-ldp-mldp)#`
		Enables MLDP.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Make-Before-Break

Perform this task to enable the make-before-break (MBB) feature in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **make-before-break** [**delay** *seconds*]
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.

	Command or Action	Purpose
Step 4	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**make-before-break** [**delay** *seconds*] **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp-af)# make-before-break delay 10`	Enables the make-before-break feature. (Optional) Configures the MBB forwarding delay in seconds. Range is 0 to 600.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP MoFRR

Perform this task to enable multicast only fast reroute (MoFRR) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **mofrr**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**mofrr** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp-af)# mofrr`	Enables MoFRR support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Recursive FEC

Perform this task to enable recursive forwarding equivalence class (FEC) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **recursive-fec**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:**	Enables MLDP for IPv4 address family.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	
Step 5	**recursive-fec** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp-af)# recursive-fec`	Enables recursive FEC support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Multipoint to Multipoint LSP

Perform this task to enable static multipoint to multipoint (MP2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static mp2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static mp2mp** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp-af)# static mp2mp 10.10.10.10 1`	Enables static MP2MP LSP support and specifies MP2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Point to Multipoint LSP

Perform this task to enable static point to multipoint (P2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static p2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:**	Enables MLDP for IPv4 address family.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	
Step 5	**static p2mp** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp-af)# static p2mp 10.0.0.1 1`	Enables static P2MP LSP support and specifies P2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling MLDP

Perform this task to disable MLDP on Label Distribution Protocol (LDP) enabled interfaces.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6** }
5. **igp mldp disable**
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.

Step 4	**address-family** {**ipv4** | **ipv6** }
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv4`
	or
	`RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv6`
		Enables the LDP IPv4 or IPv6 address family.

Step 5	**igp mldp disable** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if-af)# igp mldp disable`	Disables MLDP.
Step 6	**commit**	

Controlling State Advertisements In An mLDP-Only Setup

This function explains controlling of state advertisements of non-negotiated Label Distribution Protocol (LDP) applications. This implementation is in conformance with RFC 7473 (Controlling State Advertisements of Non-negotiated LDP Applications).

The main purpose of documenting this function is to use it in a Multipoint LDP (mLDP)-only environment, wherein participating routers don't need to exchange any unicast binding information.

Non-Negotiated LDP Applications

The LDP capabilities framework enables LDP applications' capabilities exchange and negotiation, thereby enabling LSRs to send necessary LDP state. However, for the applications that existed prior to the definition of the framework (called *non-negotiated* LDP applications), there is no capability negotiation done. When an LDP session comes up, an LDP speaker may unnecessarily advertise its local state (without waiting for any capabilities exchange and negotiation). In other words, even when the peer session is established for Multipoint LDP (mLDP), the LSR advertises the state for these early LDP applications.

One example is *IPv4/IPv6 Prefix LSPs Setup* (used to set up Label Switched Paths [LSPs] for IP prefixes). Another example is *L2VPN P2P FEC 128 and FEC 129 PWs Signaling* (an LDP application that signals point-to-point [P2P] Pseudowires [PWs] for Layer 2 Virtual Private Networks [L2VPNs]).

In an mLDP-only setup, you can disable these non-negotiated LDP applications and avoid unnecessary LDP state advertisement. An LDP speaker that only runs mLDP announces to its peer(s) its disinterest (or non-support) in non-negotiated LDP applications. That is, it announces to its peers its disinterest to set up IP Prefix LSPs or to signal L2VPN P2P PW, at the time of session establishment.

Upon receipt of such a capability, the receiving LDP speaker, if supporting the capability, disables the advertisement of the state related to the application towards the sender of the capability. This new capability can also be sent later in a Capability message, either to disable a previously enabled application's state advertisement, or to enable a previously disabled application's state advertisement.

As a result, the flow of LDP state information in an mLDP-only setup is faster. When routers come up after a network event, the network convergence time is fast too.

IP Address Bindings In An mLDP Setup

An LSR typically uses peer IP address(es) to map an IP routing next hop to an LDP peer in order to implement its control plane procedures. mLDP uses a peer's IP address(es) to determine its upstream LSR to reach the root node, and to select the forwarding interface towards its downstream LSR. Hence, in an mLDP-only network, while it is desirable to disable advertisement of label bindings for IP (unicast) prefixes, disabling advertisement of IP address bindings will break mLDP functionality.

Uninteresting State - For the *Prefix-LSP* LDP application, *uninteresting* state refers to any state related to IP Prefix FEC, such as FEC label bindings and LDP Status. IP address bindings are not considered as an *uninteresting* state.

For the P2P-PW application LDP application, *uninteresting* state refers to any state related to P2P PW FEC 128 or FEC 129, such as FEC label bindings, MAC address withdrawal, and LDP PW status.

Control State Advertisement

To control advertisement of *uninteresting* state of non-negotiated LDP applications, the capability parameter TLV *State Advertisement Control Capability* is used. This TLV is only present in the Initialization and Capability messages, and the TLV can hold one or more State Advertisement Control (SAC) Elements.

As an example, consider two LSRs, S (LDP speaker) and P (LDP peer), that support all non-negotiated applications. S is participating (or set to participate) in an mLDP-only setup. Pointers for this scenario:

- By default, the LSRs will advertise state for all LDP applications to their peers, as soon as an LDP session is established.
- The **capabilities sac mldp-only** function is enabled on S.
- P receives an update from S via a Capability message that specifies to disable all four non-negotiated applications states.

- P's outbound policy towards S blocks and disables state for the unneeded applications.
- S only receives mLDP advertisements from specific mLDP-participating peers.

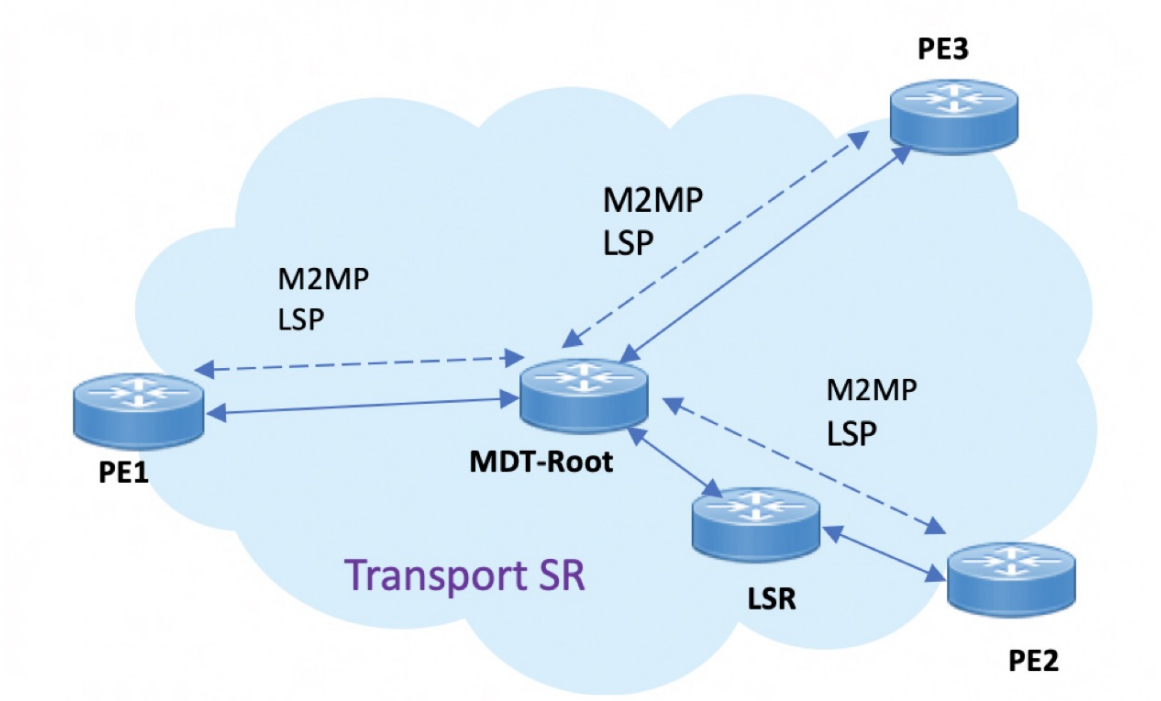
Use Cases For Controlling MLDP State Advertisements

Two use cases are explained, **mLDP-Based MVPN** and **Disable Prefix-LSPs On An L2VPN/PW tLDP Session**.

mLDP-Based MVPN

A sample topology and relevant configurations are noted below.

Figure 9: mLDP-Based MVPN Over Segment Routing



- The topology represents an MVPN profile 1 where an mLDP-based MVPN service is deployed over a Segment Routing core setup
- mLDP is required to signal MP2MP LSPs, whereas SR handles the transport.
- SAC capabilities are used to signal *mLDP-only* capability, which blocks unrequired unicast IPv4, IPv6, FEC128, and FEC129 related label binding advertisements.
- The **mldp-only** option is enabled on PE routers and P routers to remove unwanted advertisements.

Configuration

PE1 Configuration

Configure mLDP SAC capability on PE1.

```
PE1(config)# mpls ldp
PE1(config-ldp)# capabilities sac mldp-only
PE1(config-ldp)# commit
```

PE2 Configuration

Configure mLDP SAC capability on PE2.

```
PE2(config)# mpls ldp
PE2(config-ldp)# capabilities sac mldp-only
PE2(config-ldp)# commit
```

Verification

LDP peers (PE1 and PE2) are configured with **mldp-only** option, disabling all other SAC capabilities.

```
PE1# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !

PE2# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !
```

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 209.165.201.20 capabilities detail

Peer LDP Identifier: 209.165.201.20:0
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
```

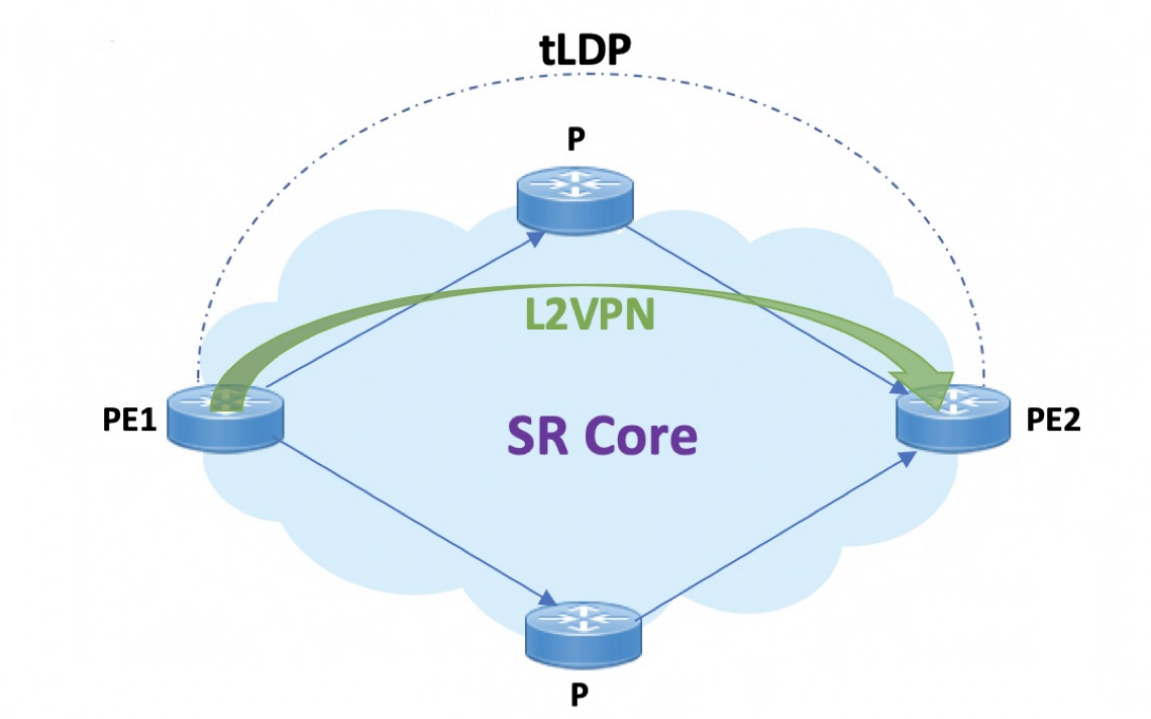
Capabilities Sent shows that **mldp-only** option disables all other advertisements.

Capabilities Received shows that **mldp-only** is enabled on peer PE2 too.

Disable Prefix-LSPs On An L2VPN/PW tLDP Session

A sample topology and relevant configurations are noted below.

Figure 10: L2VPN Xconnect Service Over Segment Routing



- The topology represents an L2VPN Xconnect service over a Segment Routing core setup.
- By default, Xconnect uses tLDP to signal service labels to remote PEs.
- By default, tLDP not only signals the service label, but also known (IPv4 and IPv6) label bindings to the tLDP peer, which is not required.
- The LDP SAC capabilities is an optional configuration enabled under LDP, and users can block IPv4 and IPv6 label bindings by applying configurations on PE1 and PE2.

Configuration

PE1 Configuration

Disable IPv4 prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable
PE1(config)# commit
```

Disable IPv6-prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable ipv6-disable
PE1(config)# commit
```

Note Whenever you disable a non-negotiated LDP application state on a router, you must include previously disabled non-negotiated LDP applications too, in the same command line. If not, the latest configuration overwrites the existing ones. You can see that ipv4-disable is added again, though it was already disabled.

PE2 Configuration

Enable SAC capability awareness on PE2, and make PE2 stop sending IPv4 prefix LSP binding advertisements to PE1:

```
PE2(config)#mpls ldp capabilities sac
PE2(config)#commit
```

Verification

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 198.51.100.1 detail

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:29132 - 192.0.2.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 14/14; Downstream-Unsolicited
  Up time: 00:03:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (3)
      203.0.113.1     209.165.201.1    10.0.0.1   198.51.100.1
      172.16.0.1
    IPv6: (0)
  Peer holdtime: 180 sec; KA interval: 60 sec; Peer state: Estab
  NSR: Disabled
  Clients: AToM
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}  ] (length 1)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
```

Capabilities Sent SAC capability **ipv4-disable** is sent, and local IPv4 label bindings are not generated.

Capabilities Received The peer (PE2) understands SAC capability and won't send its local IPv4 label bindings to local PE.

On PE1, verify SAC capabilities:

```
PE1# show mpls ldp capabilities detail

  Type     Description                                Owner
  ------   ----------------------------------------   ------------
  0x50b    Typed Wildcard FEC                         LDP
     Capability data: None

  0x3eff   Cisco IOS-XR                               LDP
     Capability data:
       Length: 12
       Desc  : [ host=PE1; platform=ASR9000; release=07.01.01 ]
```

```
0x508    MP: Point-to-Multipoint (P2MP)             mLDP
   Capability data: None

0x509    MP: Multipoint-to-Multipoint (MP2MP)       mLDP
   Capability data: None

0x50d    State Advertisement Control                LDP
   Capability data:
     Length: 1
     Desc  : [ {IPv4-disable}  ]

0x703    P2MP PW                                    L2VPN-AToM
   Capability data: None
```

On PE1, verify that local and remote FEC bindings are removed.

```
PE1# show mpls ldp neighbor 198.51.100.1
Wed March 3 13:42:13.359 EDTs
```

LDP IPv6 Configuration

The LDP configuration model is extended to introduce IPv6 as an option under the address family submodes that reside under LDP global and interface configurations. Address family IPv6 is available as a submode under LDP global, LDP VRF global and interface configurations. LDP IPv6 is supported only under default VRF.

Enabling LDP IPv6 Native

Perform this task to enable LDP IPv6 native under LDP.

The user must enable IPv6 address family under LDP submodes.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **address-family ipv6**
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:**	Enters MPLS LDP configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# mpls ldp`	

Step 3	**address-family ipv6**
	Example:
	`RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`
	`RP/0/RSP0/CPU0:router(config-ldp-af)#`
		Enables native LDP IPv6 address family.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling LDP IPv6 Control Plane

Perform this task to enable LDP IPv6 control plane on an LDP interface.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **address-family ipv6**
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface pos 0/6/0/0`	Enters interface configuration mode for the LDP protocol.
Step 4	**address-family ipv6** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv6`	Enables LDP IPv6 control plane.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-if-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-if-af)# commit`
		Note This configuration will be rejected if (mpls-ldp-af) for the given address family is not already enabled.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv6-only LSR

Perform this task to configure IPv6-only LSR.

IPv4 is implicitly enabled under default VRF and any LDP interface under default VRF. In order to operate as an IPv6-only LSR, the user must also explicitly disable IPv4 address family.

SUMMARY STEPS

1. **configure**
2. **interface loopback** *number*
3. **ipv6 address** *prefix*
4. **exit**
5. **interface** *type interface-path-id*
6. **ipv6 address** *prefix*
7. **exit**
8. **router isis** *process-id*
9. **net** *network-entity-title*
10. **interface loopback** *number*
11. **address-family ipv6 unicast**
12. **exit**
13. **exit**
14. **interface** *type interface-path-id*
15. **address-family ipv6 unicast**
16. **exit**
17. **exit**
18. **mpls ldp**
19. **default-vrf implicit-ipv4 disable**
20. **router-id** *lsr id*
21. **address-family ipv6**
22. **exit**
23. **interface** *type interface-path-id*
24. **address-family ipv6**
25. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface loopback** *number* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Loopback 0`	Enters interface configuration mode.
Step 3	**ipv6 address** *prefix* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 6:6:6::6/128`	Configures IPv6 address on interface.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0`	Enters interface configuration mode for the LDP protocol.
Step 6	**ipv6 address** *prefix* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 16:1::6/120`	Configures IPv6 address on interface.
Step 7	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 8	**router isis** *process-id* **Example:** `RP/0/RSP0/CPU0:router(config)# router isis 100`	Enables IS-IS routing for the specified routing process.
Step 9	**net** *network-entity-title* **Example:** `RP/0/RSP0/CPU0:router(config-isis)# net 49.0000.0000.0000.0006.00`	Configures the NET on the router. The NET identifies the router for IS-IS.
Step 10	**interface loopback** *number* **Example:** `RP/0/RSP0/CPU0:router(config-isis)# interface Loopback 0`	Enters interface configuration mode.
Step 11	**address-family ipv6 unicast** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv6 unicast`	Enters the IS-IS interface IPv6 address family configuration submode. Specifies unicast topology.
Step 12	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if-af)# exit`	Exits address family configuration submode and enters interface configuration mode.
Step 13	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if)# exit`	Exits interface configuration mode.
Step 14	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-isis)# interface GigabitEthernet 0/0/0/0`	Enters interface configuration mode for the LDP protocol.
Step 15	**address-family ipv6 unicast** **Example:**	Enters the IS-IS interface IPv6 address family configuration submode. Specifies unicast topology.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-isis-if)# address-family ipv6 unicast`	
Step 16	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if-af)# exit`	Exits address family configuration submode and enters interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-isis-if)# exit`	Exits interface configuration mode.
Step 18	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config-isis)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 19	**default-vrf implicit-ipv4 disable** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# default-vrf implicit-ipv4 disable`	Disables the implicitly enabled IPv4 address family for default VRF.
Step 20	**router-id** *lsr id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 5.5.5.5`	Configures router ID.
Step 21	**address-family ipv6** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`	Enables native LDP IPv6 address family.
Step 22	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# exit`	Exits the current configuration mode.

	Command or Action	Purpose
Step 23	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# interface GigabitEthernet 0/0/0/0`	Enters interface configuration mode for the LDP protocol.
Step 24	**address-family ipv6** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-if)# address-family ipv6`	Enables LDP IPv6 control plane.

Step 25	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-if-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-if-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Example

Configuring Global Transport Address for IPV6

Perform this task to configure global transport address for IPv6 address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **address-family ipv6**

4. **discovery transport-address** *ip-address*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**address-family ipv6** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv6`	Enables native LDP IPv6 address family.
Step 4	**discovery transport-address** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-ldp-af)# discovery transport-address 5:6::78`	Configures the global transport address for the specified IPv6 address.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling Implicit IPv4

Perform this task to disable the implicitly enabled IPv4 address family for default VRF.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **default-vrf implicit-ipv4 disable**
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**default-vrf implicit-ipv4 disable** **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# default-vrf implicit-ipv4 disable`	Disables the implicitly enabled IPv4 address family for default VRF.
Step 4	**end** or **commit** **Example:**	• When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

	Command or Action	Purpose

	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# commit`
		`exiting(yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv4 as Transport Preference

Perform this task to configure IPv4 as the preferred transport (overriding the default setting of IPv6 as preferred transport) to establish connection for a set of dual-stack peers.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id*
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

	Command or Action	Purpose
Step 3	**neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection prefer ipv4 for-peers 5.5.5.5`	Configures IPv4 as the preferred transport connection for the specified peer.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Transport Preference Maximum Wait Time

Perform this task to configure the maximum time (in seconds) the preferred address family connection must wait to establish transport connection before resorting to non-preferred address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection max-wait** *seconds*
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection max-wait** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection max-wait 5`	Configures the maximum wait time.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP/0/RSP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Implementing MPLS LDP

These configuration examples are provided to implement LDP:

Configuring LDP with Graceful Restart: Example

The example shows how to enable LDP with graceful restart on the POS interface 0/2/0/0.

```
mpls ldp
 graceful-restart
 interface pos0/2/0/0
!
```

Configuring LDP Discovery: Example

The example shows how to configure LDP discovery parameters.

```
mpls ldp
 router-id 192.168.70.1
 discovery hello holdtime 15
 discovery hello interval 5
!

show mpls ldp parameters
show mpls ldp discovery
```

Configuring LDP Link: Example

The example shows how to configure LDP link parameters.

```
mpls ldp
 interface pos 0/1/0/0
 !
!

show mpls ldp discovery
```

Related Topics

Configuring LDP Discovery Over a Link, on page 36
LDP Control Plane, on page 3

Configuring LDP Discovery for Targeted Hellos: Example

The examples show how to configure LDP Discovery to accept targeted hello messages.

Active (tunnel head)

```
mpls ldp
 router-id 192.168.70.1
 interface tunnel-te 12001
 !
!
```

Passive (tunnel tail)

```
mpls ldp
 router-id 192.168.70.2
 discovery targeted-hello accept
!
```

Related Topics

Configuring LDP Discovery for Active Targeted Hellos, on page 38
Configuring LDP Discovery for Passive Targeted Hellos, on page 40
LDP Control Plane, on page 3

Configuring Label Advertisement (Outbound Filtering): Example

The example shows how to configure LDP label advertisement control.

```
 mpls ldp
    address-family ipv4
      label local advertise
         disable
         for pfx_acl_1 to peer_acl_1
         for pfx_acl_2 to peer_acl_2
         for pfx_acl_3
         interface POS 0/1/0/0
         interface POS 0/2/0/0
      !
   !

ipv4 access-list pfx_acl_1
    10 permit ipv4 host 10.0.0.4 any
!
ipv4 access-list pfx_acl_2
    10 permit ipv4 host 10.20.0.4 any
!
ipv4 access-list peer_acl_1
    10 permit ipv4 host 10.0.0.1 any
    20 permit ipv4 host 10.1.1.2 any
!
ipv4 access-list peer_acl_2
    10 permit ipv4 host 172.16.0.1 any
!
!

show mpls ldp binding
```

Related Topics

Configuring Label Advertisement Control (Outbound Filtering), on page 42

Label Advertisement Control (Outbound Filtering), on page 9

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

```
mpls ldp
   address-family ipv4
     label local advertise
       disable
     !
   !
!
```

Related Topics

Setting Up LDP Neighbors, on page 43

Configuring LDP Forwarding: Example

The example shows how to configure LDP forwarding.

```
mpls ldp
 address-family ipv4
 label local advertise explicit-null
!

show mpls ldp forwarding
show mpls forwarding
```

Related Topics

Setting Up LDP Forwarding, on page 46
LDP Forwarding, on page 4

Configuring LDP Nonstop Forwarding with Graceful Restart: Example

The example shows how to configure LDP nonstop forwarding with graceful restart.

```
mpls ldp
log
graceful-restart
!
 graceful-restart
 graceful-restart forwarding state-holdtime 180
 graceful-restart reconnect-timeout 15
 interface pos0/1/0/0
!

show mpls ldp graceful-restart
show mpls ldp neighbor gr
show mpls ldp forwarding
show mpls forwarding
```

Related Topics

Setting Up LDP NSF Using Graceful Restart, on page 50
LDP Graceful Restart, on page 5
Phases in Graceful Restart, on page 7
Recovery with Graceful-Restart, on page 7

Configuring Label Acceptance (Inbound Filtering): Example

The example shows how to configure inbound label filtering.

```
mpls ldp
 label
accept
 for pfx_acl_2 from 192.168.2.2
!
 !
!
```

```
mpls ldp
 address-family ipv4
  label remote accept from 192.168.1.1:0 for pfx_acl_2
  !
 !
!
```

Related Topics

Configuring Label Acceptance Control (Inbound Filtering), on page 53
Label Acceptance Control (Inbound Filtering), on page 9

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

```
mpls ldp
 address-family ipv4
  label local allocate for pfx_acl_1
  !
 !
```

Related Topics

Configuring Local Label Allocation Control, on page 54
Local Label Allocation Control, on page 9

Configuring LDP Session Protection: Example

The example shows how to configure session protection.

```
mpls ldp
 session protection duration 60 for peer_acl_1
```

```
!
```

Related Topics

Configuring Session Protection, on page 55
Session Protection, on page 10

Configuring LDP IGP Synchronization—OSPF: Example

The example shows how to configure LDP IGP synchronization for OSPF.

```
router ospf 100
mpls ldp sync
!
mpls ldp
 igp sync delay 30
!
```

Related Topics

Configuring LDP IGP Synchronization: OSPF, on page 56
IGP Synchronization, on page 11

Configuring LDP IGP Synchronization—ISIS: Example

The example shows how to configure LDP IGP synchronization.

```
router isis 100
 interface POS 0/2/0/0
address-family ipv4 unicast
mpls ldp sync
!
 !
!
mpls ldp
 igp sync delay 30
!
```

Related Topics

Configuring LDP IGP Synchronization: ISIS, on page 60
IGP Synchronization, on page 11

Configuring LDP Auto-Configuration: Example

The example shows how to configure the IGP auto-configuration feature globally for a specific OSPF interface ID.

```
router ospf 100
 mpls ldp auto-config
 area 0
  interface pos 1/1/1/1
```

The example shows how to configure the IGP auto-configuration feature on a given area for a given OSPF interface ID.

```
router ospf 100
 area 0
  mpls ldp auto-config
  interface pos 1/1/1/1
```

Related Topics

Enabling LDP Auto-Configuration for a Specified OSPF Instance, on page 61
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance, on page 62
Disabling LDP Auto-Configuration, on page 63
IGP Auto-configuration, on page 11

Configure IP LDP Fast Reroute Loop Free Alternate: Examples

This example shows how to configure LFA FRR with default tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide

 interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path GigabitEthernet0/6/0/13 will exclude the interface
   # GigabitEthernet0/6/0/33 in LFA backup path computation.
   fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
!
 interface GigabitEthernet0/6/0/23
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/6/0/24
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/6/0/33
  point-to-point
  address-family ipv4 unicast
!
```

This example shows how to configure TE tunnel as LFA backup:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
```

```
 interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path GigabitEthernet0/6/0/13 will exclude the interface
   # GigabitEthernet0/6/0/33 in LFA backup path computation. TE tunnel 1001
   # is using the link GigabitEthernet0/6/0/33.
   fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
   fast-reroute per-prefix lfa-candidate interface tunnel-te1001
!
 interface GigabitEthernet0/6/0/33
  point-to-point
  address-family ipv4 unicast
!
```

This example shows how to configure LFA FRR with configurable tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  fast-reroute per-prefix tiebreaker ?
  downstream            Prefer backup path via downstream node
  lc-disjoint           Prefer line card disjoint backup path
  lowest-backup-metric  Prefer backup path with lowest total metric
  node-protecting       Prefer node protecting backup path
  primary-path          Prefer backup path from ECMP set
  secondary-path        Prefer non-ECMP backup path

  fast-reroute per-prefix tiebreaker lc-disjoint index ?
  <1-255>  Index
  fast-reroute per-prefix tiebreaker lc-disjoint index 10

Sample configuration:

router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  fast-reroute per-prefix tiebreaker downstream index 60
  fast-reroute per-prefix tiebreaker lc-disjoint index 10
  fast-reroute per-prefix tiebreaker lowest-backup-metric index 40
  fast-reroute per-prefix tiebreaker node-protecting index 30
  fast-reroute per-prefix tiebreaker primary-path index 20
  fast-reroute per-prefix tiebreaker secondary-path index 50
!
interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface GigabitEthernet0/1/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface GigabitEthernet0/3/0/0.1
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/3/0/0.2
  point-to-point
  address-family ipv4 unicast
```

Related Topics

IP LDP Fast Reroute Loop Free Alternate, on page 12

Verify IP LDP Fast Reroute Loop Free Alternate: Example

The following examples show how to verify the IP LDP FRR LFA feature on the router.

The following example shows how to verify ISIS FRR output:

```
RP/0/RSP0/CPU0:router#show isis fast-reroute summary

IS-IS 1 IPv4 Unicast FRR summary

                         Critical   High       Medium     Low        Total
                         Priority   Priority   Priority   Priority
Prefixes reachable in L1
  All paths protected     0          0          4          1008       1012
  Some paths protected    0          0          0          0          0
  Unprotected             0          0          0          0          0
  Protection coverage     0.00%      0.00%      100.00%    100.00%    100.00%
Prefixes reachable in L2
  All paths protected     0          0          1          0          1
  Some paths protected    0          0          0          0          0
  Unprotected             0          0          0          0          0
  Protection coverage     0.00%      0.00%      100.00%    0.00%      100.00%
```

The following example shows how to verify the IGP route 10.21.1.1/24 in ISIS Fast Reroute output:

```
RP/0/RSP0/CPU0:router#show isis fast-reroute 10.21.1.1/24

L1 10.21.1.1/24 [40/115]
     via 12.0.0.2, GigabitEthernet0/6/0/13, NORTH
       FRR backup via 14.0.2.2, GigabitEthernet0/6/0/0.3, SOUTH

RP/0/RSP0/CPU0:router#show isis fast-reroute 10.21.1.1/24 detail

L1 10.21.1.1/24 [40/115] low priority
     via 12.0.0.2, GigabitEthernet0/6/0/13, NORTH
       FRR backup via 14.0.2.2, GigabitEthernet0/6/0/0.3, SOUTH
       P: No, TM: 130, LC: No, NP: Yes, D: Yes
     src sr1.00-00, 173.1.1.2
   L2 adv [40] native, propagated
```

The following example shows how to verify the IGP route 10.21.1.1/24 in RIB output:

```
RP/0/RSP0/CPU0:router#show route 10.21.1.1/24

Routing entry for 10.21.1.0/24
  Known via "isis 1", distance 115, metric 40, type level-1
  Installed Nov 27 10:22:20.311 for 1d08h
  Routing Descriptor Blocks
    12.0.0.2, from 173.1.1.2, via GigabitEthernet0/6/0/13, Protected
      Route metric is 40
    14.0.2.2, from 173.1.1.2, via GigabitEthernet0/6/0/0.3, Backup
      Route metric is 0
```

```
  No advertising protos.
```

The following example shows how to verify the IGP route 10.21.1.1/24 in FIB output:

```
RP/0/RSP0/CPU0:router#show cef 10.21.1.1/24
10.21.1.0/24, version 0, internal 0x40040001 (ptr 0x9d9e1a68) [1], 0x0              \
(0x9ce0ec40), 0x4500 (0x9e2c69e4)
 Updated Nov 27 10:22:29.825
 remote adjacency to GigabitEthernet0/6/0/13
 Prefix Len 24, traffic index 0, precedence routine (0)
   via 12.0.0.2, GigabitEthernet0/6/0/13, 0 dependencies, weight 0, class 0,        \
protected [flags 0x400]
    path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
    next hop 12.0.0.2
     local label 16080      labels imposed {16082}
   via 14.0.2.2, GigabitEthernet0/6/0/0.3, 3 dependencies, weight 0, class 0,       \
backup [flags 0x300]
    path-idx 1
    next hop 14.0.2.2
    remote adjacency
     local label 16080      labels imposed {16079}

RP/0/RSP0/CPU0:router#show cef 10.21.1.1/24 detail
10.21.1.0/24, version 0, internal 0x40040001 (ptr 0x9d9e1a68) [1], 0x0              \
(0x9ce0ec40), 0x4500 (0x9e2c69e4)
 Updated Nov 27 10:22:29.825
 remote adjacency to GigabitEthernet0/6/0/13
 Prefix Len 24, traffic index 0, precedence routine (0)
  gateway array (0x9cc622f0) reference count 1158, flags 0x28000d00, source lsd     \
(2),
                [387 type 5 flags 0x101001 (0x9df32398) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x9ce0ec40, sh-ldi=0x9df32398]
   via 12.0.0.2, GigabitEthernet0/6/0/13, 0 dependencies, weight 0, class 0,        \
protected [flags 0x400]
    path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
    next hop 12.0.0.2
     local label 16080      labels imposed {16082}
   via 14.0.2.2, GigabitEthernet0/6/0/0.3, 3 dependencies, weight 0, class 0,       \
backup [flags 0x300]
    path-idx 1
    next hop 14.0.2.2
    remote adjacency
     local label 16080      labels imposed {16079}


    Load distribution: 0 (refcount 387)

    Hash  OK  Interface                 Address
    0     Y   GigabitEthernet0/6/0/13   remote
```

The following example shows how to verify the IGP route 10.21.1.1/24 in MPLS LDP output:

```
RP/0/RSP0/CPU0:router#show mpls ldp forwarding 10.21.1.1/24

Prefix           Label   Label      Outgoing     Next Hop            GR Stale
                 In      Out        Interface
---------------- ------- ---------- ------------ ------------------- -- -----
10.21.1.0/24     16080   16082      Gi0/6/0/13   12.0.0.2            Y  N
```

```
                          16079      Gi0/6/0/0.3  14.0.2.2 (!)        Y  N


RP/0/RSP0/CPU0:router#show mpls ldp forwarding 10.21.1.1/24 detail

Prefix           Label   Label      Outgoing     Next Hop            GR Stale
                 In      Out        Interface
---------------- ------- ---------- ------------ ------------------- -- -----
10.21.1.0/24     16080   16082      Gi0/6/0/13   12.0.0.2            Y  N
                         [ Protected; path-id 1 backup-path-id 33;
                           peer 20.20.20.20:0 ]
                         16079      Gi0/6/0/0.3  14.0.2.2 (!)        Y  N
                         [ Backup; path-id 33; peer 40.40.40.40:0 ]
  Routing update   : Nov 27 10:22:19.560 (1d08h ago)
  Forwarding update: Nov 27 10:22:29.060 (1d08h ago)
```

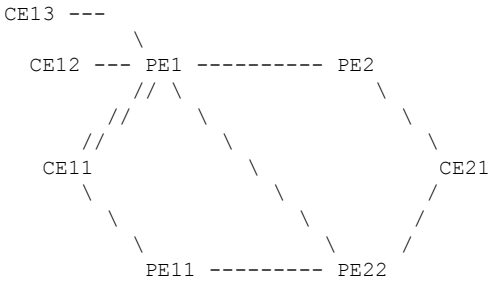
Related Topics

IP LDP Fast Reroute Loop Free Alternate, on page 12

MPLS LDP CSC for Multiple VRFs Configuration: Examples

This figure shows a L3VPN LDP CSC topology that uses either BGP or LDP between PE and CE routers to distribute routes and MPLS labels.

L3VPN CSC VPN: LDP / BGP



VRF red: CE11, CE21

VRF blue: CE12, CE13 (local only switching)

Multi-home CEs: CE11, CE21

LDP CSC: PE1/PE11 with CE1x

BGP CSC: PE2/PE22 with CE2x

CSC-CE11 Configuration

```
hostname ce11

interface Loopback0
 ipv4 address 198.51.100.254 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 192.168.1.11 255.255.255.0
```

```
!
interface POS0/2/0/1
 ipv4 address 192.168.2.11 255.255.255.0
!
interface POS0/2/0/2
 ipv4 address 192.168.3.11 255.255.255.0
!
router ospf 100
 log adjacency changes
 router-id 198.51.100.254
 area 0
  interface Loopback0
  !
  interface POS0/2/0/0
  !
  interface POS0/2/0/1
  !
  interface POS0/2/0/2
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 router-id 198.51.100.254
 address-family ipv4
 !
 interface POS0/2/0/0
  address-family ipv4
  !
 !
 interface POS0/2/0/1
  address-family ipv4
  !
 !
 interface POS0/2/0/2
  address-family ipv4
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
no interface POS0/2/0/2 shut
end
```

CSC-CE12 Configuration

```
hostname ce12

interface Loopback0
 ipv4 address 198.51.100.252 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 192.169.1.12 255.255.255.0
!
router ospf 100
 log adjacency changes
 router-id 198.51.100.252
 area 0
  interface Loopback0
```

```
  !
  interface POS0/2/0/0
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 router-id 198.51.100.252
 address-family ipv4
 !
 interface POS0/2/0/0
  address-family ipv4
  !
 !
!
no interface POS0/2/0/0 shut
end
```

CSC-CE13 Configuration

```
hostname ce13

interface Loopback0
 ipv4 address 198.51.100.254 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 192.170.1.12 255.255.255.0
!
router ospf 100
 log adjacency changes
 router-id 198.51.100.254
 area 0
  interface Loopback0
  !
  interface POS0/2/0/0
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 router-id 198.51.100.254
 address-family ipv4
 !
 interface POS0/2/0/0
  address-family ipv4
  !
 !
!
no interface POS0/2/0/0 shut
end
```

CSC-CE21 Configuration

```
hostname ce21

interface Loopback0
 ipv4 address 10.20.20.21 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 192.168.1.21 255.255.255.0
!
interface POS0/2/0/1
 ipv4 address 192.169.1.21 255.255.255.0
!
route-policy pass-all
  pass
end-policy
!
router static
 address-family ipv4 unicast
  192.168.1.2/32 POS0/2/0/0
  192.169.1.2/32 POS0/2/0/1
 !
 address-family ipv6 unicast
  1:1:1::1/128 POS0/2/0/0
 !
!
router bgp 2
 bgp router-id 10.20.20.21
 address-family ipv4 unicast
  redistribute connected
  allocate-label all
 !
 neighbor 192.168.1.2
  remote-as 100
  address-family ipv4 labeled-unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 192.169.1.22
  remote-as 100
  address-family ipv4 labeled-unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
end
```

CSC-PE1 Configuration

```
hostname pe1

vrf red
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
```

```
!
vrf blue
 address-family ipv4 unicast
 !
!
interface Loopback0
 ipv4 address 10.0.0.1 255.255.255.255
 ipv6 address 1:1:1::1/128
!
interface Loopback1
 vrf red
 ipv4 address 10.0.0.1 255.255.255.255
!
interface Loopback2
 vrf blue
 ipv4 address 10.0.0.1 255.255.255.255
!
interface Loopback11
 ipv4 address 10.0.0.2 255.255.255.255
 ipv6 address 1:1:1::2/128
!
interface Loopback112
 vrf blue
 ipv4 address 10.0.0.112 255.255.255.255
!
interface POS0/2/0/0
 vrf red
 ipv4 address 192.168.1.1 255.255.255.0
!
interface POS0/2/0/1
 vrf red
 ipv4 address 192.168.2.1 255.255.255.0
!
interface POS0/2/0/2
 vrf blue
 ipv4 address 192.169.1.1 255.255.255.0
!
interface POS0/2/0/3
 vrf blue
 ipv4 address 192.170.1.1 255.255.255.0
!
interface POS0/2/0/4
 ipv4 address 12.10.0.1 255.255.255.0
 ipv6 address 12:1::1/120
!
interface POS0/2/0/5
 ipv4 address 122.1.0.1 255.255.255.0
!
router static
 address-family ipv6 unicast
  2:2:2::2/128 POS0/2/0/4
 !
!
router ospf 100
 log adjacency changes
 router-id 10.0.0.1
 area 0
  interface Loopback0
  !
  interface POS0/2/0/4
  !
  interface POS0/2/0/5
  !
 !
```

```
 vrf red
  router-id 10.0.0.1
  redistribute bgp 100
  area 0
   interface Loopback1
   !
   interface POS0/2/0/0
   !
   interface POS0/2/0/1
   !
  !
 !
 vrf blue
  router-id 10.0.0.1
  area 0
   interface Loopback2
   !
   interface POS0/2/0/2
   !
   interface POS0/2/0/3
   !
  !
 !
!
router bgp 100
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 172.16.0.1
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 neighbor 172.16.0.12
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 vrf red
  rd 1:1
  address-family ipv4 unicast
   maximum-paths eibgp 8
   redistribute ospf 100
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 nsr
 router-id 10.0.0.1
 address-family ipv4
  label
   local
    advertise
     explicit-null
    !
   !
  !
```

```
 !
 interface POS0/2/0/4
  address-family ipv4
  !
 !
 interface POS0/2/0/5
  address-family ipv4
  !
 !
 vrf red
  address-family ipv4
  !
  interface POS0/2/0/0
   address-family ipv4
   !
  !
  interface POS0/2/0/1
   address-family ipv4
   !
  !
 !
 vrf blue
  router-id 10.0.0.2
  address-family ipv4
   discovery transport-address 10.0.0.1
   label
    local
     allocate for host-routes
    !
   !
  !
  interface POS0/2/0/2
   address-family ipv4
   !
  !
  interface POS0/2/0/3
   address-family ipv4
   !
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
no interface POS0/2/0/2 shut
no interface POS0/2/0/3 shut
no interface POS0/2/0/4 shut
no interface POS0/2/0/5 shut
end
```

CSC-PE2 Configuration

```
hostname pe2

vrf red
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
```

```
!
interface Loopback0
 ipv4 address 172.16.0.1 255.255.255.255
 ipv6 address 2:2:2::2/128
!
interface Loopback1
 vrf red
 ipv4 address 172.16.0.1 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 12.1.0.2 255.255.255.0
 ipv6 address 12:1::2/120
!
interface POS0/2/0/1
 vrf red
 ipv4 address 192.168.1.2 255.255.255.0
!
route-policy pass-all
  pass
end-policy
!
router static
 address-family ipv6 unicast
  1:1:1::1/128 POS0/2/0/0
  1:1:1::2/128 POS0/2/0/0
 !
 vrf red
  address-family ipv4 unicast
   192.168.1.21/32 POS0/2/0/1
  !
 !
!
router ospf 100
 log adjacency changes
 router-id 172.16.0.1
 area 0
  interface Loopback0
  !
  interface POS0/2/0/0
  !
 !
!
router bgp 100
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 10.0.0.1
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 vrf red
  rd 1:1
  address-family ipv4 unicast
   allocate-label all
  !
  neighbor 192.168.1.21
   remote-as 2
   address-family ipv4 labeled-unicast
    route-policy pass-all in
    route-policy pass-all out
   !
```

```
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 router-id 172.16.0.1
 address-family ipv4
  label
   local
    advertise
     explicit-null
    !
   !
  !
 !
 interface POS0/2/0/0
  address-family ipv4
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
end
```

CSC-PE11 Configuration

```
hostname pe11

vrf red
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
!
interface Loopback0
 ipv4 address 10.0.0.11 255.255.255.255
!
interface Loopback1
 vrf red
 ipv4 address 10.0.0.11 255.255.255.255
!
interface POS0/2/0/0
 vrf red
 ipv4 address 192.168.3.1 255.255.255.0
!
interface POS0/2/0/1
 ipv4 address 10.12.0.1 255.255.255.0
!
router ospf 100
 log adjacency changes
 router-id 10.0.0.11
 area 0
  interface Loopback0
  !
  interface POS0/2/0/1
```

```
  !
 !
 vrf red
  router-id 10.0.0.11
  redistribute bgp 100
  area 0
   interface Loopback1
   !
   interface POS0/2/0/0
   !
  !
 !
!
router bgp 100
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 172.16.0.12
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 vrf red
  rd 1:1
  address-family ipv4 unicast
   maximum-paths eibgp 8
   redistribute ospf 100
  !
 !
!
mpls ldp
 log
  adjacency
  neighbor
 !
 router-id 10.0.0.11
 address-family ipv4
 !
 interface POS0/2/0/1
  address-family ipv4
  !
 !
 vrf red
  address-family ipv4
  !
  interface POS0/2/0/0
   address-family ipv4
   !
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
end
```

CSC-PE22 Configuration

```
hostname pe22

vrf red
```

```
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
!
interface Loopback0
 ipv4 address 172.16.0.12 255.255.255.255
!
interface Loopback1
 vrf red
 ipv4 address 172.16.0.12 255.255.255.255
!
interface POS0/2/0/0
 ipv4 address 122.1.0.22 255.255.255.0
!
interface POS0/2/0/1
 vrf red
 ipv4 address 192.169.1.22 255.255.255.0
!
interface POS0/2/0/2
 ipv4 address 10.10.1.22 255.255.255.0
!
route-policy pass-all
  pass
end-policy
!
router static
 vrf red
  address-family ipv4 unicast
   192.169.1.21/32 POS0/2/0/1
  !
 !
!
router ospf 100
 log adjacency changes
 router-id 172.16.0.12
 area 0
  interface Loopback0
  !
  interface POS0/2/0/0
  !
  interface POS0/2/0/2
  !
 !
!
router bgp 100
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 10.0.0.1
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 neighbor 10.0.0.11
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
```

```
  !
 !
 vrf red
  rd 1:1
  address-family ipv4 unicast
   allocate-label all
  !
  neighbor 192.169.1.21
   remote-as 2
   address-family ipv4 labeled-unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
!
mpls ldp
 router-id 172.16.0.12
 address-family ipv4
 !
 interface POS0/2/0/0
  address-family ipv4
  !
 !
 interface POS0/2/0/2
  address-family ipv4
  !
 !
!
no interface POS0/2/0/0 shut
no interface POS0/2/0/1 shut
no interface POS0/2/0/2 shut
end
```

The following example shows the output for the **show running-config mpls ldp** command.

```
RP/0/RSP0/CPU0:router# show running-config mpls ldp

mpls ldp
 log
  adjacency
  neighbor
  nsr
  graceful-restart
  session-protection
 !
 nsr
 graceful-restart
 graceful-restart reconnect-timeout 60
 graceful-restart forwarding-state-holdtime 180
 igp sync delay on-proc-restart 300
 igp sync delay on-session-up 15
 discovery
  quick-start disable
  instance-tlv disable
  hello holdtime 30
  hello interval 10
  targeted-hello holdtime 180
  targeted-hello interval 20
 !
 session backoff 5 15
 session holdtime 300
```

```
signalling dscp 48
mldp
 logging notifications
 address-family ipv4
  static p2mp 10.0.0.1 1
  static mp2mp 10.10.20.10 1
  make-before-break delay 10
  mofrr
  recursive-fec
 !
!
router-id 10.0.0.1
neighbor
 password encrypted 01100F17580454
 172.16.0.1:0 password disable
 192.168.0.1:0 password encrypted 02050D480809
!
session downstream-on-demand with peer_acl1
session protection for peer_acl2 duration 30
address-family ipv4
 discovery targeted-hello accept from peer_acl1
 neighbor 172.16.0.1 targeted
 traffic-eng
  auto-tunnel mesh
   group all
   group 10
   group 20
  !
 !
 redistribute
  bgp
   as 100
   advertise-to peer_acl1
  !
 !
 label
  local
   default-route
   implicit-null-override for pfx_acl1
   allocate for pfx_acl
   advertise
    disable
    for pfx_acl1 to peer_acl1
    for pfx_acl2 to peer_acl2
    interface GigabitEthernet0/0/0/0
    explicit-null for pfx_acl1 to peer_acl1
   !
  !
  remote
   accept
    from 172.16.0.1:0 for pfx_acl2
    from 192.168.0.1:0 for pfx_acl3
   !
  !
 !
!
interface GigabitEthernet0/0/0/0
 igp sync delay on-session-up disable
 discovery quick-start disable
 discovery hello holdtime 30
 discovery hello interval 10
 address-family ipv4
  igp auto-config disable
  discovery transport-address interface
```

```
   mldp disable
  !
 !
 interface GigabitEthernet0/0/0/1
  igp sync delay on-session-up 10
  address-family ipv4
   discovery transport-address 10.0.0.1
  !
 !
 interface GigabitEthernet0/0/0/2
 !
!
```

LDP IPv6 Configuration: Examples

The following example shows how to enable LDP IPv6 native under LDP. The user must enable IPv6 address family under LDP submodes.

```
configure
  mpls ldp
   address-family ipv6
   !
  !
```

The following example shows how to enable LDP IPv6 control plane on an LDP interface:

```
configure
  mpls ldp
    interface pos 0/6/0/0
      address-family ipv6
    !
  !
```

The following examples shows how to configure IPv6-only LSR:

Note IPv4 is implicitly enabled under default VRF and any LDP interfaces under default VRF. In order to operate as an IPv6-only LSR, the user must also explicitly disable IPv4 address family.

Example 1:

Note In this example, there is no explicit IPv6 export address. The loopback's IPv6 address is used as the export address (6:6:6::6/128).

The router ID configured in MPLS LDP is not used in anyway for export. It is used only for LDP LSR identification.

```
configure
interface Loopback0
 ipv6 address 6:6:6::6/128
!
```

```
interface GigabitEthernet0/0/0/0
 ipv6 address 16:1::6/120
!
router isis 100
 net 49.0000.0000.0000.0006.00
 interface Loopback0
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/0
  address-family ipv6 unicast
  !
 !
mpls ldp
 default-vrf implicit-ipv4 disable
 router-id 6.6.6.6
 address-family ipv6
 !
 interface GigabitEthernet0/0/0/0
  address-family ipv6
 !
!
```

Example 2:

Note In this example, there is an explicit IPv6 export address. However, there is no IPv6 loopback.

There is no router-id configured, but the loopback IPv4 address is used.

```
configure
interface Loopback0
 ipv4 address 6.6.6.6/32
!
interface GigabitEthernet0/0/0/0
 ipv6 address 16:1::6/120
!
router isis 100
 net 49.0000.0000.0000.0006.00
 interface Loopback0
  address-family ipv6 unicast
  !
 !
 interface GigabitEthernet0/0/0/0
  address-family ipv6 unicast
  !
 !
mpls ldp
 default-vrf implicit-ipv4 disable
 address-family ipv6
  discovery transport-address 6:6:6::6
 !
 interface GigabitEthernet0/0/0/0
  address-family ipv6
 !
!
```

Entropy Label Support for Transit Routers

Entropy label (EL) in MPLS is a mechanism that improves load balancing across a network. Load balancing helps in planning the capacity of a network by distributing traffic across multiple paths based on hashing functions.



Note On ASR 9000 series routers, entropy label is supported only on transit routers. The data plane of the transit router does not have the functionality to impose entropy label on the MPLS packet if there are any equal cost paths available for a given LSP.

Traffic load balancing over Equal Cost Multipath (ECMP) or Link Aggregation Groups (LAGs) is usually based on a hashing function. To arrive at the hash calculations, the node that performs the load balancing must read header fields in the incoming packets. Currently, Label Switching Routers (LSRs) at each transit point must do a Deep Packet Inspection (DPI) along the path of a given Label Switched Path (LSP). This includes extracting the appropriate keys for load balancing. If the LSR is unable to infer the protocol, it will use the topmost MPLS labels in the label stack as keys to balance the load. This may result in unbalanced distribution of traffic.

Entropy labels enhance load balancing by eliminating the need for DPI at the transit LSRs. The transit router recognizes the incoming MPLS packets with entropy label and performs the load balancing and forwards the MPLS packet on a selected path. The input packets are assumed to have valid EL labels within the first seven labels. Else, either IP header or other MPLS labels are used for load balancing.

The ingress LSR of an LSP computes the hash based on appropriate fields from a given packet and places the result in a label called entropy label as part of the MPLS label stack. Using the entropy label in the hash keys reduces the need of a DPI inspection in the LSR. The transit LSR can use the entire label stack of the MPLS packet to perform load balancing, as the entropy label introduces the right level of order into the label stack.

Advantages of Entropy Label

The advantages of using entropy labels in MPLS networks are:

- Ingress LSRs operate at lower bandwidths than transit LSRs, and are hence the ideal choice for load balancing.
- Transit LSRs do not need to perform DPI and can effectively load balance the packets as decided by the Ingress LSRs.
- Transit LSRs are spared from the problem of misinterpreting the protocol denoted in the label stack and thereby causing inequitable distribution of traffic across equal cost paths exiting from the LSR.

Enable Entropy Label Support on Transit Routers

Entropy label (EL) supports an orderly method for routers to signal entropy label capability (ELC) in the network. When enabled, the routers wait for the ELC signal from all downstream routers before passing their ELC to the next upstream routers in the chain. This ensures that routers report their status in an order, and not in random. Random reporting might require to and fro signaling before ELC can be established in the network. If one router in the chain does not support EL, the network does not use EL for load balancing.

Procedure

Step 1 Enable Entropy label LDP signalling.

```
Router# configure
Router(config)# mpls ldp
Router(config)# entropy-label
Router(config-ldp)# router-id {type number | ip-address}
router(config-ldp)# interface type number
router(config-ldp)# commit
router(config-ldp)# end
```

The router ID is specified as an interface name or IP address. By default, LDP uses the global router ID that is configured by the global router ID process.

Step 2 Enable using entropy label value as a field in the hash calculation for load balancing during forwarding.

```
Router# configure
Router(config)# cef load-balancing fields mpls entropy-label
router(config-ldp)# commit
```

The **cef load-balancing fields mpls entropy-label** command configures the hash tuple with the following fields.

- entropy label
- router ID
- ingress interface

Note
The **cef load-balancing fields mpls entropy-label** command is supported only on Cisco ASR 9000 enhanced ethernet line cards.

Step 3 Display the running configuration that contains the load balancing information.

```
Router# show running config
```

Step 4 Determine load balancing using entropy label. These commands provide the output interface chosen as a result of hashing with MPLS entropy label:

ECMP Path Selection

The following example shows the output for ECMP path selection:

```
Router# show mpls forwarding exact-route label 24001 entropy-label 1234
ingress-interface tenGigE 0/0/0/1/0 location 0/0/CPU0

Local    Outgoing      Prefix      Outgoing        Next Hop        Bytes
Label    Label         or ID       Interface                       Switched
------ ----------- -------------- ------------ ---------------   -----------
24001    64002       194.0.0.1/32  Te0/0/0/1/0.1   25.2.11.1         N/A
    Via: Te0/0/0/1/0.1, Next Hop: 25.2.11.1
    Label Stack (Top -> Bottom): { 64002 }
    NHID: 0x4, Encap-ID: N/A, Path idx: 2, Backup path idx: 0, Weight: 0
    Hash idx: 2
    MAC/Encaps: 18/22, MTU: 1500
    Outgoing Interface: TenGigE0/0/0/1/0.1 (ifhandle 0x00000500)
```

Bundle Member Selection

The following example shows the output for Bundle member selection:

```
Router# bundle-hash bundle-ether 5001 location 0/0/CPU0
Calculate Bundle-Hash for L2 or L3 or sub-int based: 2/3/4 [3]: 3
Enter traffic type (1:IPv4-inbound, 2:MPLS-inbound, 3:IPv6-inbound, 4:IPv4-MGSCP, 5:IPv6-MGSCP): [1]:
 2
Entropy label: y/n [n]: y
Enter Entropy Label (in decimal): 1997
Enter the source interface name (Enter to skip interface details): TenGigE0/0/0/1/0
Entropy Label 1997 -- Link hashed to is TenGigE0/1/0/29, (raw hash 0xb5703292, LAG hash 2, ICL (),
LON 2, IFH 0x06001740)

Another? [y]:
```

Additional References

For additional information related to Implementing MPLS Label Distribution Protocol, refer to the following references:

Related Documents

Related Topic	Document Title
LDP Commands	*MPLS Label Distribution Protocol Commands* module in *MPLS Command Reference for Cisco ASR 9000 Series Routers*.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs **Note** Not all supported RFCs are listed.	Title
RFC 3031	*Multiprotocol Label Switching Architecture*
RFC 3036	*LDP Specification*
RFC 3037	*LDP Applicability*
RFC 3478	*Graceful Restart Mechanism for Label Distribution Protocol*
RFC 3815	*Definitions of Managed Objects for MPLS LDP*
RFC 5036	*Label Distribution and Management* *Downstream on Demand Label Advertisement*
RFC 5286	*Basic Specification for IP Fast Reroute: Loop-Free Alternates*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport