Implementing MPLS Label Distribution Protocol

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Implementing MPLS label distribution protocol	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-12G12X4Y-A • 8011-12G12X4Y-D
Implementing MPLS label distribution protocol	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Implementing MPLS label distribution protocol	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

In IP forwarding, when a packet arrives at a router the router looks at the destination address in the IP header, performs a route lookup, and then forwards the packet to the next hop. MPLS is a forwarding mechanism in

which packets are forwarded based on labels. Label Distribution Protocols assign, distribute, and install the labels in an MPLS environment. It is the set of procedures and messages by which Label Switched Routers (LSRs) establish LSPs through a network by mapping network-layer routing information directly to data-link layer switched paths. These LSPs may have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or may have an endpoint at a network egress node, enabling switching via all intermediary nodes.

LSPs 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 enables LSRs to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information. Once label bindings are learned, the LDP is ready to setup the MPLS forwarding plane.

For MPLS LDP, Graceful Out of Resource (OOR) handling is supported from Release 7.3.2 onwards.

Prerequisites for Implementing MPLS Label Distribution Protocol

The following are the prerequisites 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.

Note This point is not appplicable for a Cisco NCS 540 Series Router.

- 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.

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
- From Release 24.4.1 onwards, to improve user readability of MPLS route statistics summary when there is a large number of MPLS routes coming from Service Layer API (SL-API) clients, the **show service-layer mpls label**[*label-id* | **client** *client-id*] command now shows a shortened output per path under MPLS. Prior to this release, this command provided detailed output per path under MPLS.

  Example:

  Sample Output - Old Behavior

```
RP/0/RP0/CPU0:ios#show service-layer mpls label
MPLS Label :46002,, tag: 0, distance: 2
    path: 1, 1.1.1.1, via HundredGigE 0/0/0/0,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60001,
        interface name: HundredGigE 0/0/0/0,

    path: 2, 2.1.1.1, via HundredGigE 0/0/0/1,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60011,
```

```
        interface name: HundredGigE 0/0/0/1,

    path: 3, ……
MPLS Label :46003,, tag: 0, distance: 2
    path: 1, 1.1.1.1, via HundredGigE 0/0/0/0,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60001,
        interface name: HundredGigE 0/0/0/0,

    path: 2, 2.1.1.1, via HundredGigE 0/0/0/1,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60011,
        interface name: HundredGigE 0/0/0/1,

    path: 3, ……
MPLS Label :46004,, ……
……
```

Sample Output - New Behavior

```
MPLS Label :46002, tag: 0, distance: 2
    path: 1, 1.1.1.1 via HundredGigE 0/0/0/0
    path: 2, 2.1.1.1 via HundredGigE 0/0/0/1
    path: 3, 3.1.1.1 via HundredGigE 0/0/0/2
MPLS Label :46003, tag: 0, distance: 2
    path: 1, 1.1.1.1 via HundredGigE 0/0/0/0
    path: 2, 2.1.1.1 via HundredGigE 0/0/0/1
    path: 3, 3.1.1.1 via HundredGigE 0/0/0/2
.
.
.
```

Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand these concepts:

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.
- 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 8000 Series Routers*.

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).

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.

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*.

Conditional label advertisement in label-switched path networks

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Conditional label advertisement in label-switched path networks	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Conditional label advertisement in label-switched path networks	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Conditional label advertisement in label-switched path networks	Release 24.2.11	You can now enhance your network's stability and performance with the streamlined label management. This can be achieved by configuring LDP to advertise labels to peers only when at least one labeled path is available for a prefix. Previously, LDP would advertise local labels to peers even if all next-hop paths for a specific Forwarding Equivalence Class (FEC) had no labels. This release has the following changes: **CLI:** • Introduced a new keyword **unlabelled-all** in **show mpls ldp forwarding** command. • **conditional minimum-one-labelled-nexthop**

LDP Behavior in Label Advertisement

In an LDP-enabled network, when LDP isn't active on certain links, LDP assigns an unlabeled route to tunnels or interfaces which are inoperative. However, LDP continues to advertise labels to its peers even when all possible paths for a specific prefix are unlabeled. This behavior becomes problematic when the forwarding state of a Forwarding Equivalence Class (FEC) is unstable, causing frequent changes in the network's state (known as flapping). This flapping results in a high volume of label updates and withdrawals, creating unnecessary signaling traffic, known as churn, in the network.

When LDP assigns an unlabeled route to a prefix, it can disrupt Equal-Cost Multi-Path (ECMP) routing, leading to failed paths. Despite of these issues, LDP doesn’t automatically retract the assigned label even if the traffic is being forwarded without labels.

With this feature enabled, LDP advertises labels to its peers if there’s at least one label path is available for a prefix. If LDP detects that all outgoing paths for a prefix are unlabeled, it withdraws the previously advertised label.

Default Behavior of LSP Networks in Label Advertisement

This feature is disabled by default.

Feature History Table

ASR 9000

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
<Feature Title>	<Mention the Release re-use variable>	<Feature Description> The feature introduces these changes: **CLI:** • **abc** • The **abc** keyword is introduced in the **xyz** command. **YANG Data Models:** • `abc.yang` • New Xpaths for `abc.yang` (see GitHub, YANG Data Models Navigator)

NCS 5500

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
<Feature Title>	<Mention the Release re-use variable>	<Feature Description> The feature introduces these changes: **CLI:** • **abc** • The **abc** keyword is introduced in the **xyz** command. **YANG Data Models:** • `abc.yang` • New Xpaths for `abc.yang` (see GitHub, YANG Data Models Navigator)

Cisco 8000

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
<Feature Title>	<Mention the Release, use re-use variable	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100])(select variants only*); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) <Feature Description> The feature introduces these changes: **CLI:** • **abc** • The **abc** keyword is introduced in the **xyz** command. **YANG Data Models:** • `abc.yang` • New Xpaths for `abc.yang` (see GitHub, YANG Data Models Navigator) This feature is supported on: • <PID> • <PID>. Also, specify if there are requirements for a specific Fabric card (FC) or Route Processor (RP)

Limitations

This feature doesn't impact BGP route types as LDP allocates and installs no forwarding for the following scenarios:

- Interactions between autonomous systems in BGP, which involves different policies for label distribution and route propagation.
- Carrier Supporting Carrier (CSC) where one carrier provides MPLS services to another carrier.
- Seamless-MPLS to simplify network operations and support end-to-end services.
- Interworking of LDP-based MPLS systems with the BGP-SR (Segment Routing with BGP) systems.

Configuration

Configuration Example

```
Router# configure
Router(config)# mpls ldp
```

```
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# label local advertise conditional minimum-one-labelled-nexthop
Router(config-ldp-af)# exit
```

Verification

To view prefixes with all paths being unlabelled, use **show mpls ldp forwarding next-hop unlabelled-all** command.

```
Router# show mpls ldp forwarding next-hop unlabelled-all

Prefix          Label-In  Label(s)-Out  Outgoing-Interface Next Hop     Flags G S R E
--------------- ------- -------------- ------------ ---------------------------------
14.14.14.14/32  24006     Unlabelled     Gi0/2/0/2         13.13.13.2
15.15.15.15/32  24007     Unlabelled     Gi0/2/0/2         13.13.13.2
16.16.16.16/32  24008     Unlabelled     Gi0/2/0/0         10.10.10.2
                                         Unlabelled        Gi0/2/0/2
```

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.

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).

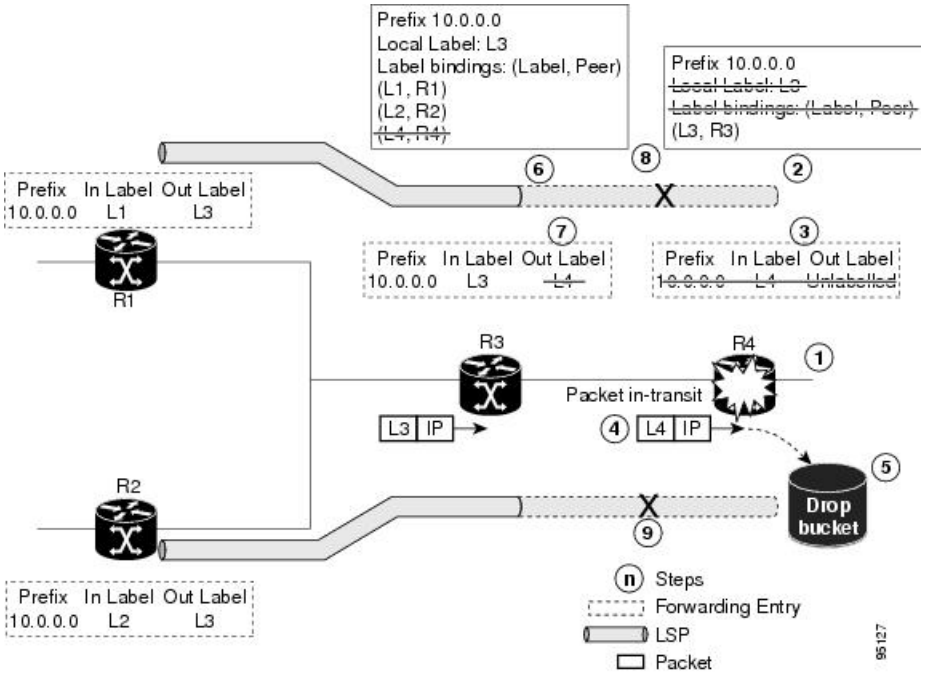
If an address family is not enabled on receiving LSR, LDP discards any bindings received from peer(s) for the address family. This means that when address family is enabled, LDP needs to reset existing sessions with the peers in order to re-learn the discarded bindings. The implementation is optimized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

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 1: 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.

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.

Label Distribution Protocol Discovery Parameters

Discovery parameter specifies the time periods between transmitted and not received hello messages.

Configuration Example

A discovery parameter specifies time of the discovered neighbor (15 seconds) which is kept without receipt of any subsequent hello messages. After the specified time period, there is an interval of 5 seconds between the transmission of consecutive hello messages.

Configuration of Label Distribution Protocol Discovery Parameters

```
Router(config)#mpls ldp
Router(config-ldp)#router-id 192.168.70.1
Router(config-ldp)#discovery hello holdtime 15
Router(config-ldp)#discovery targeted-hello holdtime 5
Router(config-ldp)#commit
```

Verification

Displays all the current MPLS LDP parameters.

```
RP/0/RP0/CPU0:router# show mpls ldp parameters
LDP Parameters:
Role: Active
Protocol Version: 1
Router ID: 192.168.70.1

Discovery:
Link Hellos:     Holdtime:15 sec, Interval:5 sec
Targeted Hellos: Holdtime:5 sec, Interval:10 sec
Quick-start: Enabled (by default)
Transport address:       IPv4: 192.168.70.1
```

Downstream on Demand

The 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.

Explicit-null and implicit-null labels

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Explicit-null and implicit-null labels	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
Explicit-null and implicit-null labels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The MPLS Label Distribution Protocol (LDP) is integrated to streamline label distribution and management in MPLS networks. This enhancement optimizes label switching by establishing LDP sessions between routers to exchange label mapping information efficiently. LDP supports the creation of LSPs (Label Switched Paths), ensuring effective and scalable data forwarding across the network. By facilitating dynamic label assignment, it reduces manual configuration efforts and enhances network flexibility, contributing to more reliable and efficient traffic management. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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.

Label distribution protocol interior gateway protocol synchronization

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Label distribution protocol interior gateway protocol synchronization	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Label distribution protocol interior gateway protocol synchronization	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100])(select variants only*). The MPLS Label Distribution Protocol (LDP) enables the establishment of label-switched paths (LSPs) by mapping network routes to labels, facilitating efficient packet forwarding. This feature streamlines the process of routing by dynamically distributing labels to network devices, enhancing scalability and reducing configuration complexity. LDP is integral for optimizing network performance in MPLS environments, ensuring reliable and efficient data transmission. *Previously this feature was supported on Q200 and Q100. It is now extended to Cisco 8712-MOD-M routers.

Label distribution protocol interior gateway protocol synchronization	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M

Lack of synchronization between LDP and Interior Gateway Protocol (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 coordinates 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, 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 check-point recovered LDP graceful restart session is used and treated as converged and is given an opportunity to connect and resynchronize.

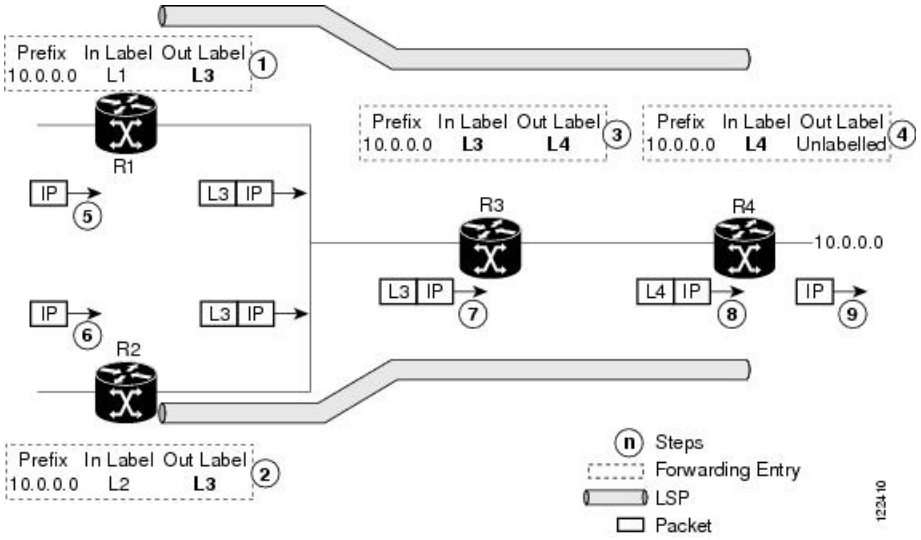
Under certain circumstances, it might be required to delay declaration of re-synchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.

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 (Layer 1, Layer 3).
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 (Layer 2, Layer 3).
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 (Layer 3, Layer 4).
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 (Layer 4); 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.

Setting up Label Distribution Protocol Graceful Restart

Configuration Example

This example shows how to configure LDP graceful restart. In this example, the amount of time that a neighboring router maintains the forwarding state about the gracefully restarting router is specified as 180 seconds. Also, the amount of time the LDP neighbor should wait for a reconnection from the gracefully restarting router in the event of a LDP session failure is specified as 169 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#interface TenGigE 0/0/0/5
Router(config-ldp-if)#exit
Router(config-ldp)#graceful-restart
Router(config-ldp)#graceful-restart forwarding-state-holdtime 180
Router(config-ldp)#graceful-restart reconnect-timeout 169
Router(config-ldp)#commit
```

Verification

```
RP/0/RP0/CPU0:router#show mpls ldp graceful-restart
Forwarding State Hold timer : Not Running
GR Neighbors : 1

Neighbor ID      Up  Connect Count  Liveness Timer      Recovery Timer
---------------  --  -------------  ------------------  ------------------
8.8.8.8          Y        1                -                   -

RP/0/RP0/CPU0:router#show mpls ldp parameters
Graceful Restart:Enabled
Reconnect Timeout:169 sec, Forwarding State Holdtime:180 sec
NSR: Disabled, Not Sync-ed
```

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.

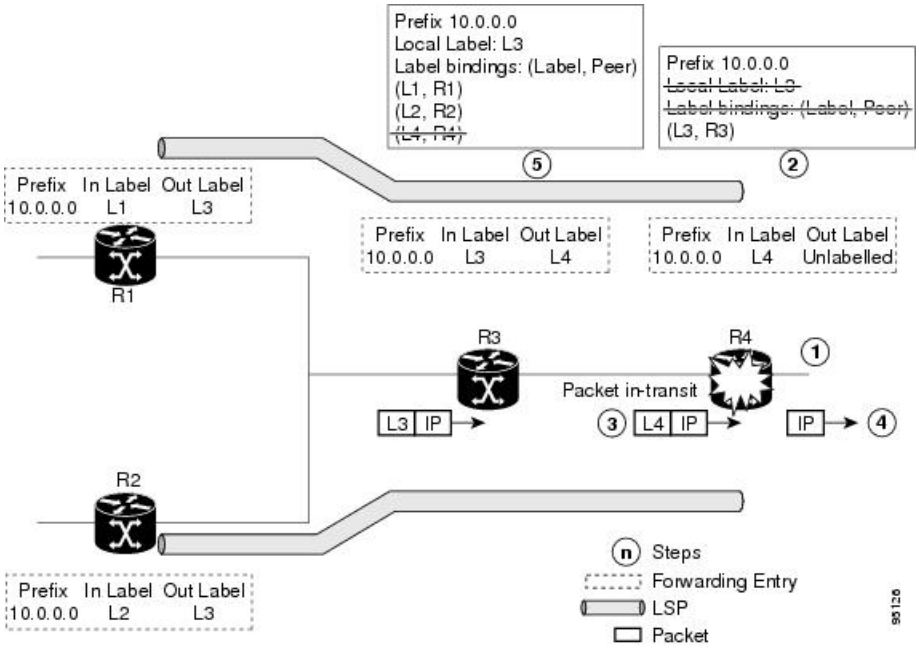
Control state recovery

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.

Recovery with Graceful-Restart

Figure 3: 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.

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 8000 Series Routers*.

Local Label Allocation Control

Label Distribution Protocol 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 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.

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

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/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**redistribute bgp** **Example:** `RP/0/RP0/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.

	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.
Step 5	**show run mpls ldp** **Example:** `RP/0/RP0/CPU0:router# show run mpls ldp`	Displays information about the redistributed route information.

Session protection

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Session protection	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*)(select variants only*) *This feature is supported on: • 8711-48Z-M
Session protection	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

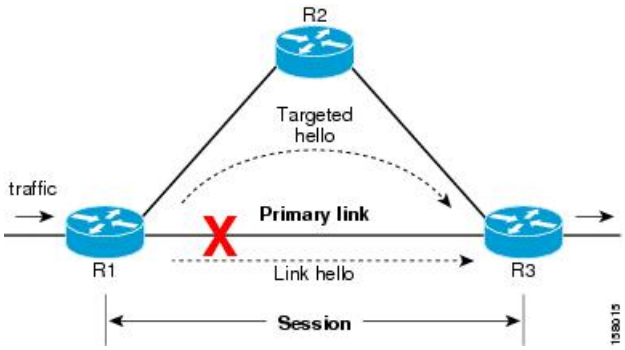
Session protection	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) Improve network robustness with MPLS LDP session protection, which safeguards Label Distribution Protocol sessions from interruptions. This feature maintains session continuity during network instability, reducing the risk of label distribution loss. It ensures stable MPLS operations by quickly recovering LDP sessions after disruptions, thereby enhancing overall network reliability and performance. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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.

This 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 4: Session protection



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

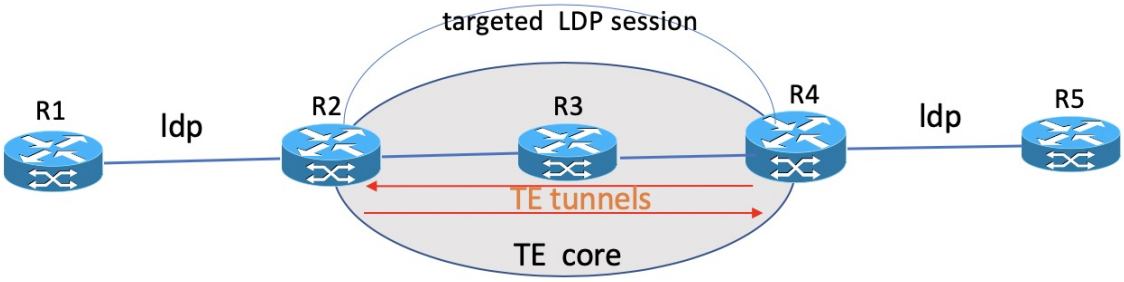
LDP over RSVP LSR support

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
LDP over RSVP LSR support	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
LDP over RSVP LSR support	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
LDP over RSVP LSR support	Release 7.3.1	With this feature, users can transport LDP traffic over an RSVP-TE network automatically, through a targeted LDP session. The automatic configuration for LDP over RSVP-TE supports 1000 TE tunnels.

Consider this topology of an RSVP-TE network spanning R2 to R4. LDP traffic is transported from R1 to R5. A targeted LDP session is established between R2 and R4 so that LDP traffic is transported over the TE tunnel network.

Figure 5: LDP Over RSVP



No additional configuration is required to enable the LDP over RSVP-TE function. Up to 1000 tunnels are supported by default. The **autoroute announce** command is enabled on the edge routers of the RSVP-TE network.

If you need more than 1000 TE tunnels, enable the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers R2 and R4. However, when you enable this command, the LDP over TE feature gets disabled.

The following configuration disables the LDP over TE function, and allows you run more than 1000 TE tunnels.

```
Router# configure terminal
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
Router# reload
```

Increase in RSVP-TE Tunnel Scale for LDP over TE

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 7.5.3	We allow you to use a maximum of 4000 RSVP-TE tunnels with LDP over TE using the **hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte** command. Earlier, only 1000 RSVP-TE tunnels were supported. If you further need to increase the number of RSVP-TE tunnels up to 8000, enable the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers. However, the LDP over TE is disabled when you enable this command.

You can scale up to 4000 RSVP-TE tunnels only on routers with line cards based on Q200 Silicon. In this scenario, LDP over TE feature is enabled which means you have the ability to steer LDP prefixes or labeled SR prefixes over a higher number of RSVP-TE tunnels.

To enable up to a maximum of 4000 RSVP-TE tunnels with LDP over TE feature, use the **hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte** command on the edge routers.

```
Router# configure
Router(config)# hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte
Router(config)# commit
Router# reload
```

In scenarios where you require additional tunnels without requiring to steer LDP prefixes or labeled SR prefixes over RSVP-TE tunnels, use the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers to configure up to a maximum of 8000 TE tunnels.

```
Router# configure terminal
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
Router# reload
```

Note Reload the line card for changes to take effect.

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 :

Implementing MPLS Label Distribution Protocol

MPLS (Multi Protocol Label Switching) is a forwarding mechanism based on label switching. In an MPLS network, data packets are assigned labels and packet-forwarding decisions are taken based on the contents of the label. To switch labeled packets across the MPLS network, predetermined paths are established for various source-destination pairs. These predetermined paths are known as Label Switched Paths (LSPs). To establish LSPs, MPLS signaling protocols are used. Label Distribution Protocol (LDP) is an MPLS signaling protocol used for establishing LSPs. This module provides information about how to configure MPLS LDP.

MPLS Restrictions

Labeled packets are not supported on Linux interfaces.

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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `Router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**mldp** **Example:**	Enables MLDP.

	Command or Action	Purpose

	`Router(config-ldp)# mldp`
	
	`Router(config-ldp-mldp)#`	

Step 4	**end** or **commit**
	Example:
	`Router(config-ldp-mldp)# end`
	or
	`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.
		• 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:**	Enters XR Config mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**make-before-break** [**delay** *seconds*] **Example:** `RP/0/RP0/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/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/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/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**mofrr** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# mofrr`	Enables MoFRR support.

	Command or Action	Purpose

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/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 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/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**recursive-fec** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# recursive-fec`	Enables recursive FEC support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/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/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static mp2mp** *ip-address* **Example:** `RP/0/RP0/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.

	Command or Action	Purpose

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/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 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/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static p2mp** *ip-address* **Example:** `RP/0/RP0/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.

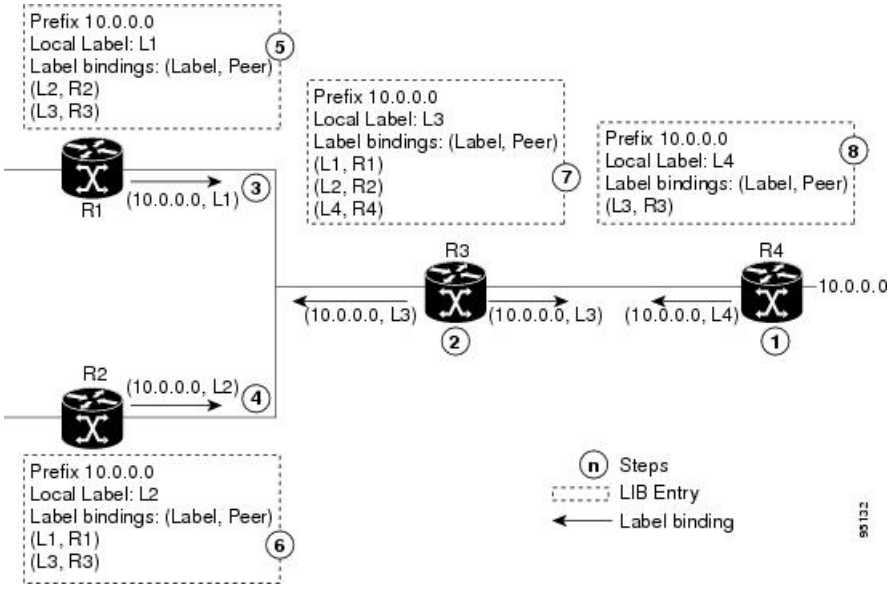
Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [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.

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 6: 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.

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. **label advertise** { **disable** | **for** *prefix-acl* [**to** *peer-acl*] | **interface** *type interface-path-id* }
4. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	**label advertise** { **disable** | **for** *prefix-acl* [**to** *peer-acl*] | **interface** *type interface-path-id* }
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# label advertise interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-ldp)# for pfx_acl1 to peer_acl1`
		Configures label advertisement by specifying one of the following options:
		disable
		Disables label advertisement to all peers for all prefixes (if there are no other conflicting rules).
		interface
		Specifies an interface for label advertisement of an interface address.
		for ***prefix-acl***
		to ***peer-acl***
		Specifies neighbors to advertise and receive label advertisements.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		end —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. **label**
4. **implicit-null-override for** *access-list*
5. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**label** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# label`	Configures the allocation, advertisement ,and acceptance of labels.
Step 4	**implicit-null-override for** *access-list* **Example:**	Configures implicit-null local label for non-egress prefixes. **Note**

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp-af-lbl)# implicit-null-override for 70`	This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. **explicit-null**
4. Use the **commit** or **end** command.
5. (Optional) **show mpls ldp forwarding**
6. (Optional) **show mpls forwarding**
7. (Optional) **ping** *ip-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**explicit-null** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# 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 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	(Optional) **show mpls ldp forwarding** **Example:** `RP/0/RP0/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 6	(Optional) **show mpls forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls forwarding`	Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).

	Command or Action	Purpose
Step 7	(Optional) **ping** *ip-address* **Example:** `RP/0/RP0/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).

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. **discovery transport-address** [*ip-address* | **interface**]
5. **exit**
6. **holdtime** *seconds*
7. **neighbor** *ip-address* **password** [*encryption*] *password*
8. **backoff** *initial maximum*
9. Use the **commit** or **end** command.
10. (Optional) **show mpls ldp neighbor**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:**	Enters interface configuration mode for the LDP protocol.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	
Step 4	**discovery transport-address** [*ip-address* | **interface**] **Example:** or `RP/0/RP0/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 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 6	**holdtime** *seconds* **Example:** `RP/0/RP0/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 7	**neighbor** *ip-address* **password** [*encryption*] *password* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor 192.168.2.44 password secretpasswd`	Configures password authentication (using the TCP MD5 option) for a given neighbor.
Step 8	**backoff** *initial maximum* **Example:** `RP/0/RP0/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.

	Command or Action	Purpose
Step 9	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/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.

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. Use the **commit** or **end** command.
9. (Optional) **show mpls ldp parameters**
10. (Optional) **show mpls ldp neighbor**
11. (Optional) **show mpls ldp graceful-restart**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol.

Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 5	**graceful-restart** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart`	Enables the LDP graceful restart feature.
Step 6	**graceful-restart forwarding-state-holdtime** *seconds* **Example:** `RP/0/RP0/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:**	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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169`	start a timer on the peer (upon a neighbor restart). This timer is referred to as *Neighbor Liveness* timer.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 9	(Optional) **show mpls ldp parameters** **Example:** `RP/0/RP0/CPU0:router # show mpls ldp parameters`	Displays all the current MPLS LDP parameters.
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/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/RP0/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.

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**}
5. **label remote accept from** *ldp-id* **for** *prefix-acl*
6. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**label accept for** *prefix-acl* **from** *ip-address*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1`
	`RP/0/RP0/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**}
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`
	`RP/0/RP0/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/RP0/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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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** } **unicast**
5. **mpls ldp sync**
6. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router isis** *instance-id*
	Example:
	`RP/0/RP0/CPU0:router(config)# router isis 100`
	`RP/0/RP0/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/RP0/CPU0:router(config-isis)# interface`	Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.

	Command or Action	Purpose
	HundredGigE 0/0/0/2 RP/0/RP0/CPU0:router(config-isis-if)#	
Step 4	**address-family** {**ipv4** } **unicast** **Example:** RP/0/RP0/CPU0:router(config-isis-if)# **address-family ipv4 unicast** RP/0/RP0/CPU0:router(config-isis-if-af)#	Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) address prefix.
Step 5	**mpls ldp sync** **Example:** RP/0/RP0/CPU0:router(config-isis-if-af)# **mpls ldp sync**	Enables LDP IGP synchronization.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`	Enables LDP IPv4 address family.
Step 4	**discovery transport-address** *ip-address* **Example:** `RP/0/RP0/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/RP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP0/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.

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/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id* **Example:** `RP/0/RP0/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/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/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.

	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.

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. Use the **commit** or **end** command.
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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/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/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/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.

	Command or Action	Purpose
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/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/RP0/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/RP0/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/RP0/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/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

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. Use the **commit** or **end** command.
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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/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/RP0/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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/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/RP0/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/RP0/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/RP0/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/RP0/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/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

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. Use the **commit** or **end** command.
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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:**	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	• 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/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.

Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/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/RP0/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/RP0/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/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.

	Command or Action	Purpose
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/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/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Configuring Downstream on Demand

By default, LDP uses downstream unsolicited mode in which label advertisements for all routes are received from all LDP peers. The 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.

In downstream on demand configuration, an ACL is used to specify the set of peers for downstream on demand mode. For down stream on demand to be enabled, it needs 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.

Configuration Example

This example shows how to configure LDP Downstream on Demand.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session downstream-on-demand with ACL1
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Configuring LDP Link: Example

The example shows how to configure LDP link parameters.

```
mpls ldp
 interface HundredGigE 0/0/0/2
 !
!

show mpls ldp discovery
```

Configuring Label Distribution Protocol Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) fail over, 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
- 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. L2VPN configuration is not supported on NSR. Process failures of active LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action.

Configuration Example

This example shows how to configure LDP Non-Stop Routing.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# nsr
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Verification

```
RP/0/RP0/CPU0:Router# show mpls ldp nsr summary
Mon Dec  7 04:02:16.259 UTC
Sessions:
Total: 1, NSR-eligible: 1, Sync-ed: 0
(1 Ready)
```

Configure Session Protection

Configuration Example

As per the configuration, LDP session protection for peers specified by peer-acl is configured to maximum duration of 60 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#session protection for peer_acl_1 duration 60
Router(config-ldp)#commit
```

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. **label allocate for** *prefix-acl*
4. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**label allocate for** *prefix-acl* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# label allocate for pfx_acl_1`	Configures label allocation control for prefixes as specified by prefix-acl.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection max-wait** *seconds* **Example:** `RP/0/RP0/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/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/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.

	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.

Configuring MPLS Label Security

Perform this task to configure the MPLS label security on an interface

Configuration Example

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# mpls label-security rpf
Router(config-if)# commit
```

Verification

Use the **show mpls forwarding label-security interface** command to view MPLS label security configuration on an interface.

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**default-vrf implicit-ipv4 disable** **Example:** `RP/0/RP0/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:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/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.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**mpls ldp**
	Example:
	`RP/0/RP0/CPU0:router(config)# mpls ldp`
	`RP/0/RP0/CPU0:router(config-ldp)#`
		Enters the MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	Enters interface configuration mode and configures an interface.
Step 4	**igp auto-config disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# igp auto-config disable`	Disables auto-configuration on the specified interface.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use one of the following commands:
   - **area** *area-id* **mpls ldp sync disable**
   - **area** *area-id* **interface** *name* **mpls ldp sync disable**
4. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 109`	Identifies the OSPF routing process and enters OSPF configuration mode.

Step 3	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/RP0/CPU0:router(config-ospf)# area 1 mpls ldp sync disable`
	`RP/0/RP0/CPU0:router(config-ospf)# area 1 interface HundredGigE 0/0/0/2 mpls ldp sync disable`
		Disables LDP IGP synchronization on an interface.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		• **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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** }
5. **igp mldp disable**
6. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	Enters interface configuration mode for the LDP protocol.
Step 4	**address-family** {**ipv4** } **Example:**	Enables the LDP IPv4 address family.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp-if)# address-family ipv4` or	
Step 5	**igp mldp disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if-af)# igp mldp disable`	Disables MLDP.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Auto-enable LDP on all TE tunnel interfaces

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Auto-enable LDP on all TE tunnel interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
Auto-enable LDP on all TE tunnel interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Auto-enable LDP on all TE tunnel interfaces	Release 7.11.1	You can now automatically enable all the configured LDP tunnels over IPv4 traffic-engineering (TE) with upto 2000 unique destinations to achieve time efficiency and consistency of LDP tunnel configurations in routers within an MPLS network. Earlier, you could enable the LDP tunnels over TE manually. This feature introduces the following changes: **CLI**: mpls ldp address-family ipv4 traffic-eng tunnels command. **YANG Data Model**: Cisco-IOS-XR-mpls-ldp-cfg (see GitHub, YANG Data Models Navigator)

Label Distribution Protocol over Traffic Engineering (LDPoTE) enables the establishment of label-switched paths (LSPs) in an MPLS network with specific traffic engineering requirements, such as link bandwidth, quality of service (QoS), and path optimization. LDPoTE enhances the scalability and efficiency of MPLS networks. It provides the ability to control traffic paths, allocate network resources, and optimize network performance based on traffic engineering policies.

From Release 7.11.1, you can automatically enable (auto-enable) all the configured tunnels over IPv4 TE at once upto 2000 unique destination routers using the **mpls ldp address-family ipv4 traffic-eng tunnels** command.

You can auto-enable:

- Named tunnels—Tunnels that are identified by specific identifiers or names. For example: ldp_tunnel
- Numbered tunnels—Tunnels that are identified by numerical values. For example: tunnel0
- Using a regular expression—Tunnels that are matched with the regular expression. For example: .*-1.

Limitations

Auto-enable LDP on all TE tunnel interfaces doesn't support:

- IPv6 LDP tunnel configuration
- Automatically establishing the LDP tunnels in a mesh network
- Auto-configuring of cloned tunnels

Configure Auto-enable LDP on all TE Tunnel Interfaces

Auto-enable LDP on all TE tunnel interfaces can be done for:

- All the named and numbered tunnels.
- All the tunnels which are matched with the regular expression.

Configuration Example for Auto-enabling LDP on all Named and Numbered TE Tunnel Interfaces

```
Router# configure
Router(config)# mpls ldp address-family ipv4 traffic-eng tunnels all
Router(config)# commit
Router(config)# end
```

Running Configuration

```
mpls ldp
 address-family ipv4
  traffic-eng
   tunnels all
  !
 !
```

Verification

Use the **show mpls ldp interface** command to verify the auto-enabled tunnels.

```
Router# show mpls ldp interface
Interface interface tunnel-te 1 (0xf0)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
Interface interface tunnel-te 2 (0x110)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
Interface interface tunnel-te 3 (0x210)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
```

Configuration Example for Auto-enabling LDP on TE Tunnel Interfaces With Regular Expression

```
Router# configure
Router(config)# mpls ldp address-family ipv4 traffic-eng tunnels regular-expression .*-1
Router(config)# end
```

Running Configuration

```
mpls ldp
 address-family ipv4
```

```
  traffic-eng
   tunnels regular-expression .*-1
  !
 !
```

Verification

```
Router# show mpls ldp interface
Interface cisco-1 (0x110)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All with regex '.*-1'
Interface tunnel-te1-1 (0x210)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All with regex '.*-1'
```

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 190`
	`RP/0/RP0/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/RP0/CPU0:router(config-ospf)# mpls ldp auto-config`	Enables LDP auto-configuration.
Step 4	**area** *area-id* **Example:** `RP/0/RP0/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/RP0/CPU0:router(config-ospf-ar)# interface HundredGigE 0/0/0/2`	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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 64

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 100`
	`RP/0/RP0/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/RP0/CPU0:router(config-ospf)# area 8`
	`RP/0/RP0/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/RP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config`	Enables LDP auto-configuration.

Step 5	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigE 0/0/0/2`
	`RP/0/RP0/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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 64

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 63

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)
```

Verify IP LDP Fast Reroute Loop Free Alternate: Example

Configuration Examples for Implementing MPLS LDP

These configuration examples are provided to implement LDP:

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
  !
```

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 HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path HundredGigE 0/0/0/0 will exclude the interface
   # HundredGigE 0/0/0/3 in LFA backup path computation.
```

```
   fast-reroute per-prefix exclude interface HundredGigE 0/0/0/3
!
 interface HundredGigE 0/0/0/1
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/2
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/3
  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 HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path HundredGigE 0/0/0/0 will exclude the interface
   # HundredGigE 0/0/0/3 in LFA backup path computation. TE tunnel 1001
   # is using the link HundredGigE 0/0/0/3.
   fast-reroute per-prefix exclude interface HundredGigE 0/0/0/3
   fast-reroute per-prefix lfa-candidate interface tunnel-te1001
!
 interface HundredGigE 0/0/0/3
  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 HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface HundredGigE 0/0/0/1.1
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/1.2
  point-to-point
  address-family ipv4 unicast
```

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

```
  mpls ldp
   label
  allocate for pfx_acl_1
   !
  !
```

Configuring LDP with Graceful Restart: Example

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

interface.

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

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
```

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 HundredGigE 0/0/0/2
!

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

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
   !
  !
 !
```

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 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 HundredGigE 0/0/0/2
```

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 HundredGigE 0/0/0/2
```

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

```
mpls ldp
    router-id 192.168.70.1
    neighbor 10.0.0.1 password encrypted 110A1016141E
    neighbor 172.16.0.1 implicit-withdraw
!
```

Configuring LDP IGP Synchronization—ISIS: Example

The example shows how to configure LDP IGP synchronization.

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

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
!
```

Label Distribution Protocol Interior Gateway Protocol Auto-configuration

Interior Gateway Protocol (IGP) auto-configuration allows you to 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.

Controlling State Advertisements in an mLDP-Only Setup

Table 12: Feature History Table

Feature Name	Release Information	Description
Controlling State Advertisements in an mLDP-Only Setup	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Controlling State Advertisements in an mLDP-Only Setup	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Description
Controlling State Advertisements in an mLDP-Only Setup	Release 7.5.2	In conformance with RFC 7473, you can control state advertisements of non-negotiated Label Distribution Protocol (LDP) applications in a Multipoint LDP (mLDP)-only environment. In such an environment, participating routers don't need to exchange any unicast binding information. As a result, the flow of LDP state information in an mLDP-only setup is faster. Also, when routers come up after a network event, the network convergence time is shorter.

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.

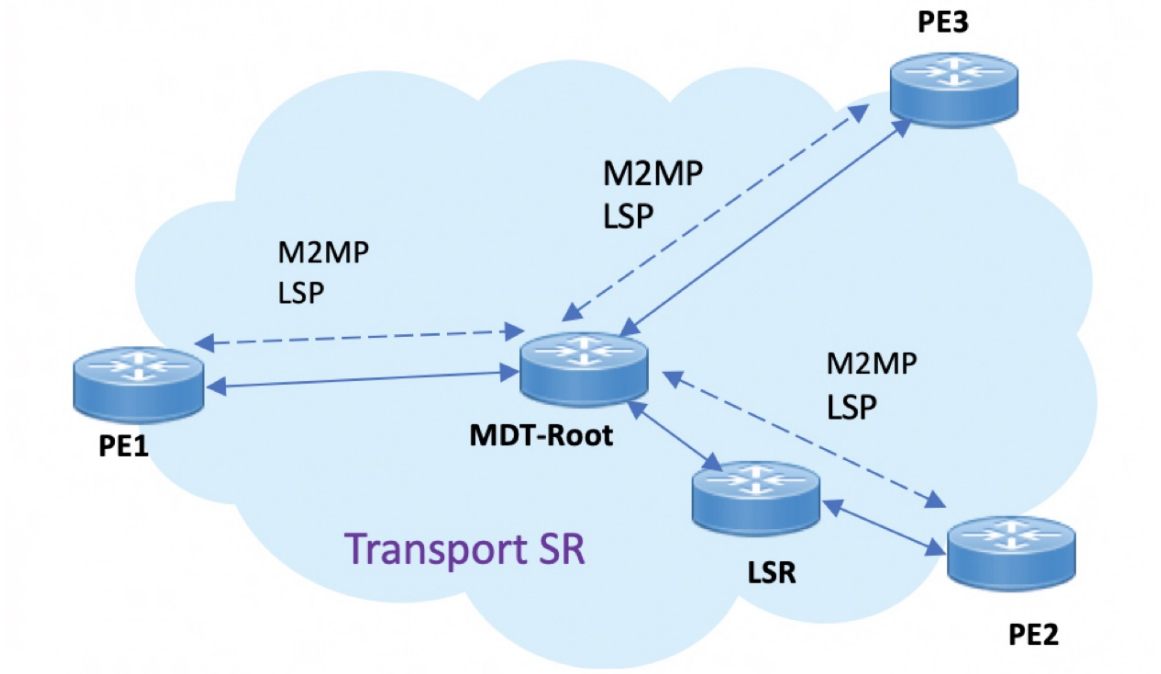
Use Cases For Controlling State Advertisements

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 7: 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)
```

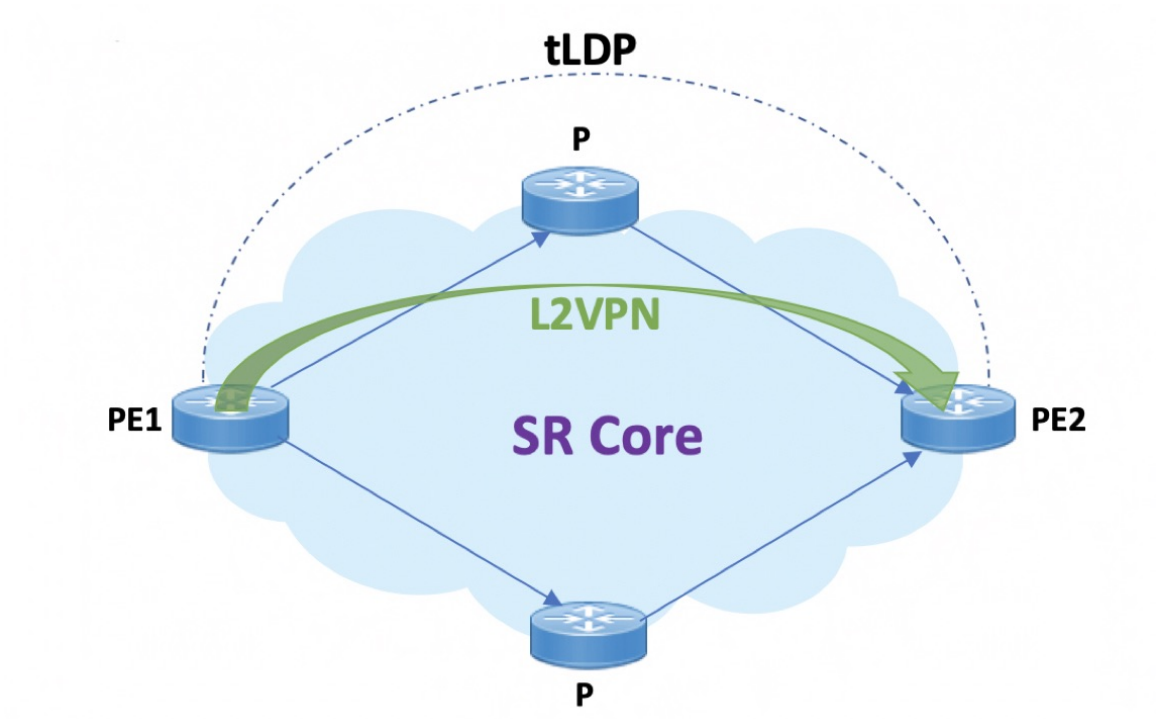
Capabilities Sent shows that **mldp-only** option disables all other advertisements.

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 8: 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
```

ECMP and Bundle Hashing with Entropy Label

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

Note The routers do not support imposition or disposition of EL.

Traffic load balancing over Equal Cost Multipath (ECMP) or Link Aggregation Groups (LAGs) is based on a hashing function. To arrive at the hash calculations, the node that performs the load balancing must read the 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 uses the topmost MPLS labels in the label stack as keys to balance load. This may result in an 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 an entropy label and performs the load balancing and forwards the MPLS packet on a selected path.

The ingress LSR of an LSP computes the hash that is based on appropriate fields from a given packet and places the result in a label that is called an entropy label as part of the MPLS label stack. Using the entropy label in the hash keys reduces the need of a DPI 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.

For more information on EL, see *RFC 6790*.

MPLS Hashing

The hashing uses the label stack and the payload when the label stack contains EL. However, load balancing functionality considers EL like any other label.

Starting from the Cisco IOS-XR Release 7.3.1, hashing is performed as described in the following table:

Table 13: MPLS Hashing

MPLS Payload	Fields Considered for Hashing	Description
IPv4	Router ID + Label stack + Src IP + Dst IP + L4 Protocol + Src Port + Dst Port	If IPv4 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
IPv6	Router ID + Label stack + Src IP + Dst IP + Flow-label + L4 Protocol + Src Port + Dst Port	If IPv6 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
GTP-u (IPv4 or IPv6)	Router ID + Label stack + GTP TEID	GPRS tuneling protocol. GTP-u uses UDP destination port 2152. The same fields are used for both IPv4 and IPv6. GTP TEID is a 32-bit tunnel end-point identifier.

MPLS Payload	Fields Considered for Hashing	Description
Ethernet	Router ID + Label stack + Dest MAC + Src MAC + Ether-type + VLAN	If Ethernet is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • For untagged Ethernet, destination MAC address, source MAC address, and Ethernet type are used. • For tagged Ethernet, destination MAC address, source MAC address, Ethernet type, and first VLAN tags are used. • When L2VPN is configured with the Control word, destination MAC address and source MAC address are used. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.

Load Balancing based on the Position of Entropy Label

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
Load Balancing based on the Position of Entropy Label	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Load Balancing based on the Position of Entropy Label	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Load Balancing based on the Position of Entropy Label	Release 7.5.2	The router achieves multi-pathing or traffic load balancing based on the Entropy Label (EL) and Entropy Label Indicator (ELI) position. The load balancing is based on the router ID and label stack if the ELI is in any of the three bottom labels. If not, the load balancing is based on all labels in the label stack plus the MPLS payload.

MPLS label stack contains Entropy Label (EL) and Entropy Label Indicator (ELI). The label immediately preceding an EL in the MPLS label stack is an ELI. The ELI uses a reserved label value of 7.

Starting from the Cisco IOS-XR Release 7.5.2, the load balancing is based on the placement of the EL and ELI. If the ELI and EL are placed at the bottom three label entry positions, load balancing uses all labels, else uses the MPLS payload along with other labels.

For GTP payload, the presence of EL does not change the hashing mechanism and it remains the same as described in the *MPLS Hashing* table.

The following table shows how load balancing is performed based on the position of ELI.

Position of ELI (Label Value 7)	Description
Bottom three labels	If there is ELI in any of the three bottom label positions, the load balacing is based on the router ID and label stack. The load balacing uses up to 14 labels in the label stack.
Anywhere else in the stack	The load balacing is performed based on the entropy label as described in the *MPLS Hashing* table. The load balacing uses all labels in the label stack and MPLS payload.

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 8000 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