Implementing MPLS Label Distribution Protocol on Cisco IOS XR Software

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

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

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

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

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

Feature History for Implementing MPLS LDP on Cisco IOS XR Software

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	Support was added for the Cisco XR 12000 Series Router. Support was added for conceptual and configuration information about LDP Label Advertisement Control (Outbound label filtering).
Release 3.3.0	Support was added for • Inbound Label Filtering • Local Label Allocation Control • Session Protection • LDP-IGP Synchronization
Release 3.4.0	No modification.
Release 3.4.1	No modification.

Release 3.5.0	Support was added for LDP Auto-configuration.
Release 3.6.0	Support was added for LDP nonstop routing (NSR).
Release 3.7.0	No modification.

Contents

Prerequisites for Implementing Cisco MPLS LDP

The following prerequisites are required to implement MPLS LDP:

- You must be in a user group associated with a task group that includes the proper task IDs for MPLS LDP commands.
- You must be running Cisco IOS XR software.
- You must install a composite mini-image and the MPLS package.
- You must activate IGP.

Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand the following concepts:

Overview of Label Distribution Protocol

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

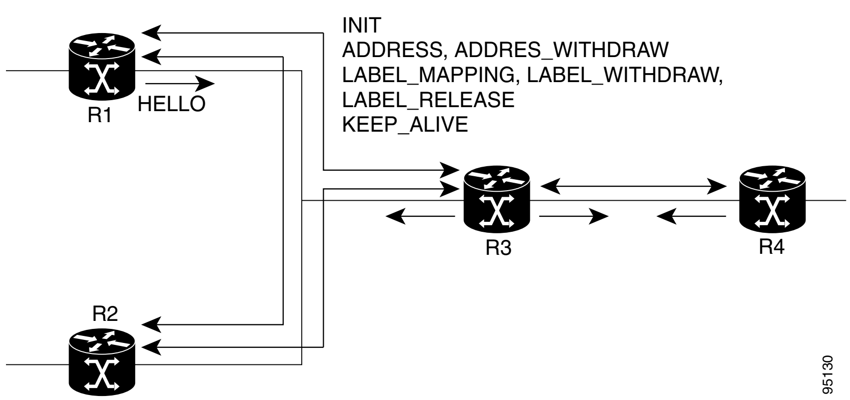
Label Switched Paths

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

LDP Control Plane

The control plane enables label switched routers (LSRs) to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information. Figure 1 shows the control messages exchanged between LDP peers.

Figure 1 LDP Control Protocol



LDP uses the hello discovery mechanism to discover its neighbor or peer on the network. When LDP is enabled on an interface, it sends hello messages to a link-local multicast address, and joins a specific multicast group to receive hellos from other LSRs present on the given link. When LSRs on a given link receive hellos, their neighbors are discovered and the LDP session (using TCP) is established.

Note Hellos are not only used to discover and trigger LDP sessions; they are also required to maintain LDP sessions. If a certain number of hellos from a given peer are missed in sequence, LDP sessions are brought down, until the peer is discovered again.

LDP also supports non-link neighbors that could be multiple hops away on the network, using the targeted hello mechanism. In these cases, hellos are sent on a directed, unicast address.

The first message in the session establishment phase is the initialization message, which is used to negotiate session parameters. After session establishment, LDP sends a list of all its interface addresses to its peers in an address message. Whenever a new address becomes available or unavailable, the peers are notified regarding such changes via ADDRESS or ADDRESS_WITHDRAW messages respectively.

When MPLS LDP learns an IGP prefix it allocates a label locally as the inbound label. The local binding between the prefix label is conveyed to its peers via LABEL_MAPPING message. If the binding breaks and becomes unavailable, a LABEL_WITHDRAW message is sent to all its peers, which respond with LABEL_RELEASE messages.

The local label binding and remote label binding received from its peer(s) is used to setup forwarding entries. Using routing information from the IGP protocol and the forwarding information base (FIB), the next active hop is selected. Label binding is learned from the next hop peer, and is used as the outbound label while setting up the forwarding plane.

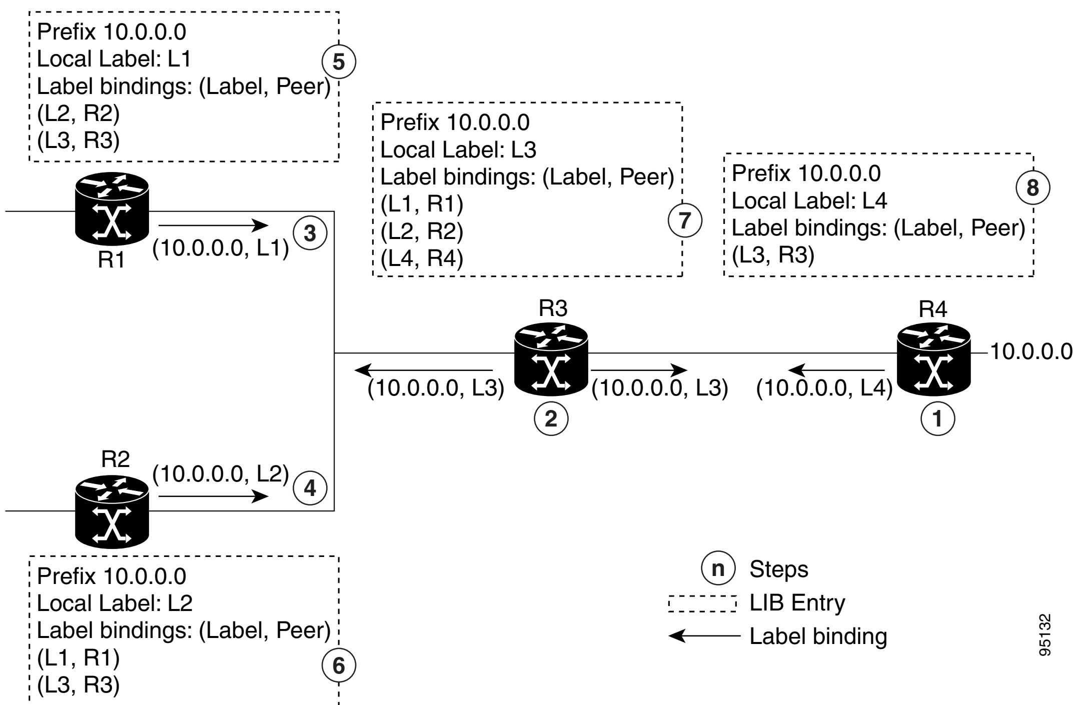
The LDP session is also kept alive using the LDP keepalive mechanism, where an LSR sends a keepalive message periodically to its peers. If no messages are received and a certain number of keepalive messages are missed from a peer, the session is declared dead, and brought down immediately.

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

Figure 2 Setting Up Label Switched Paths



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:

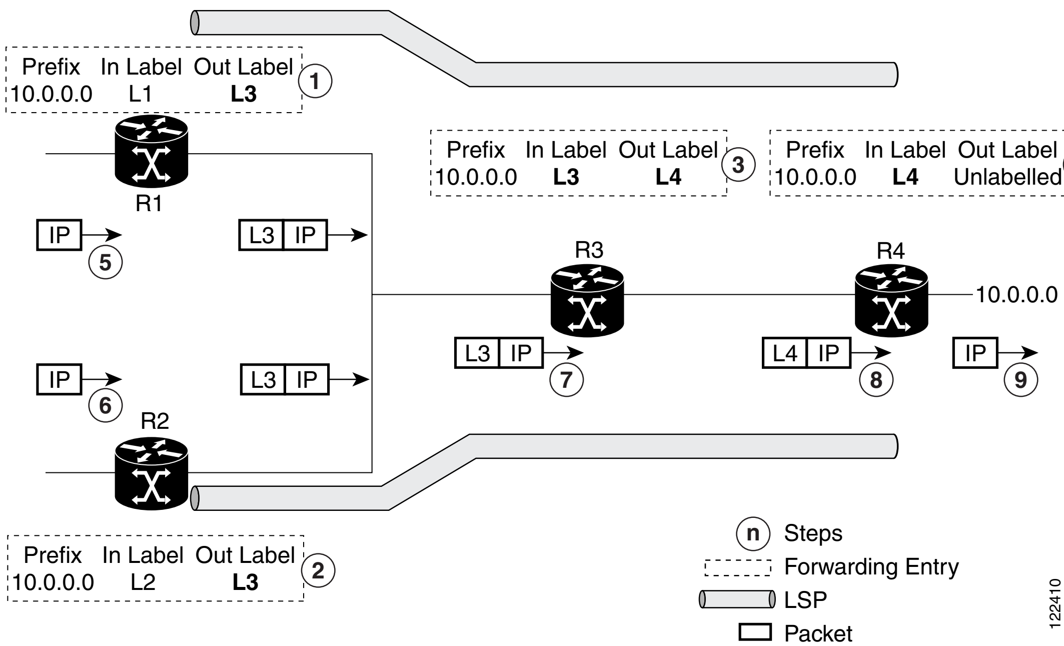
1. R4 allocates local label L4 for prefix 10.0.0.0 and advertises it to its neighbors (R3).
2. R3 allocates local label L3 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R2, R4).
3. R1 allocates local label L1 for prefix 10.0.0.0 and advertises it to its neighbors (R2, R3).
4. R2 allocates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

Setting Up LDP Forwarding

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in Figure 3.

Figure 3 Forwarding Setup



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

8. R4 receives an MPLS packet with label L4, looks up in the MPLS label forwarding table and finds that it should be Unlabeled, pops the top label, and passes it to the IP forwarding plane.
9. IP forwarding takes over and forwards the packet onward.

LDP Graceful Restart

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

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

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

- Reconnect time: the maximum time that other peer will wait for this LSR to reconnect after control channel failure.
- Recovery time: Max time that other peer has on its side to reinstate or refresh its states with this LSR. This time is used only during session reestablishment after earlier session failure.
- FT flag: This flag indicates whether a restart could restore the preserved (local) node state.

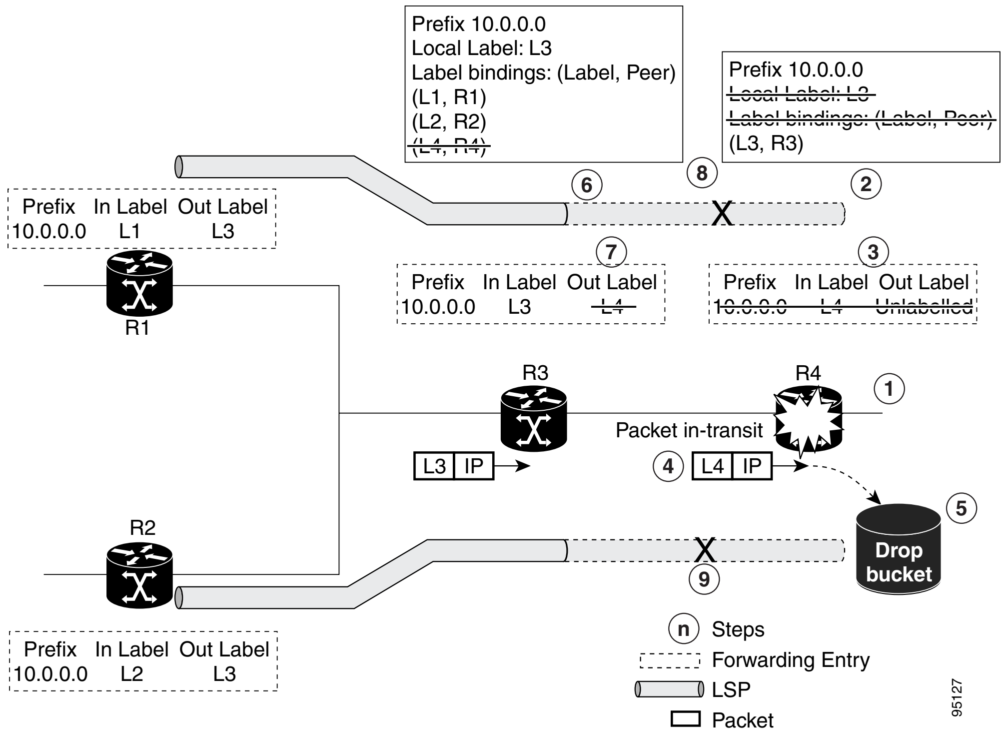
Once the graceful restart session parameters are conveyed and the session is up and running, graceful restart procedures are activated.

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

Figure 4 ***Control Plane Failure***



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

Phases in Graceful Restart

The graceful restart mechanism can be divided into different phases as follows:

- Control communication failure detection
- Forwarding state maintenance during failure
- Control state recovery

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.

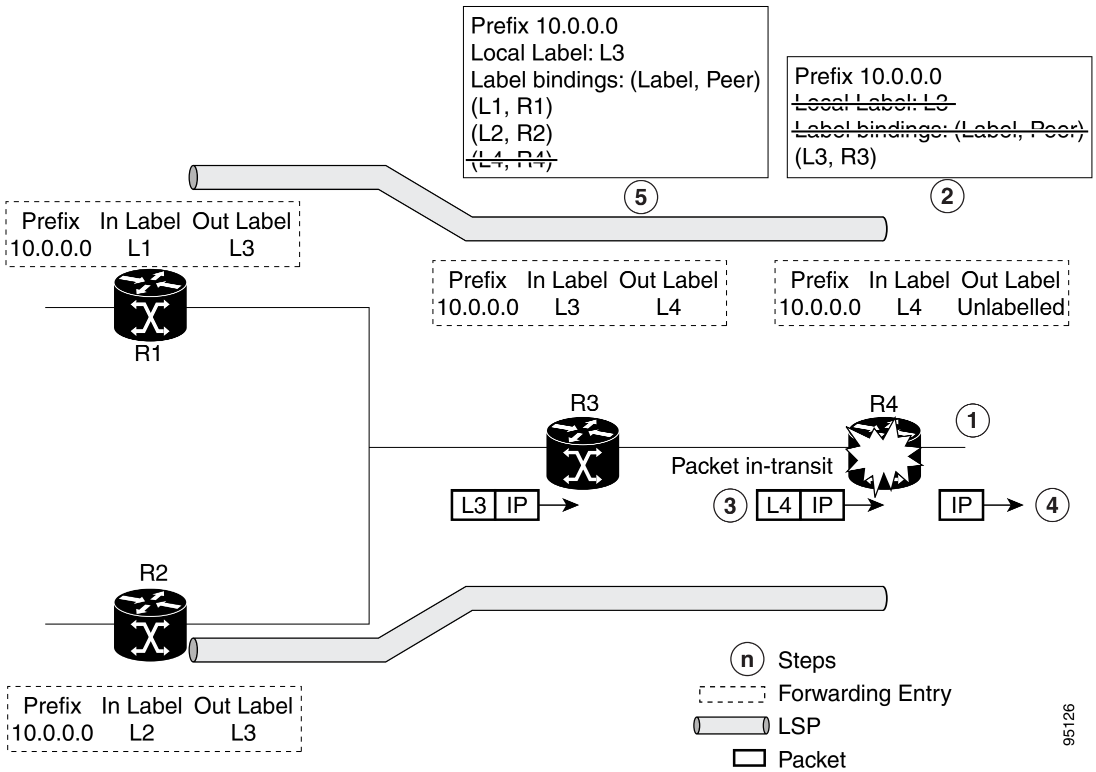
Control State Recovery

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 5 illustrates the process of failure recovery using graceful restart.

Figure 5 Recovering with 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 labelled 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, indicating 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.

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.*

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.

Local Label Allocation Control

By default, LDP allocates local labels for all prefixes that are not Border Gateway Protocol (BGP) prefixes[1]. This is acceptable when LDP is used for applications other than Layer 3 virtual private networks (L3VPN) core transport. When LDP is used to set up transport LSPs for L3VPN traffic in the core, it is not efficient or even necessary to allocate and advertise local labels for, potentially, thousands of IGP prefixes. In such a case, LDP is typically required to allocate and advertise local label for loopback /32 addresses for PE routers. This is accomplished using LDP local label allocation control, where an access list can be used to limit allocation of local labels to a set of prefixes. Limiting local label allocation provides several benefits, including reduced memory usage requirements, fewer local forwarding updates, and fewer network and peer updates.

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

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

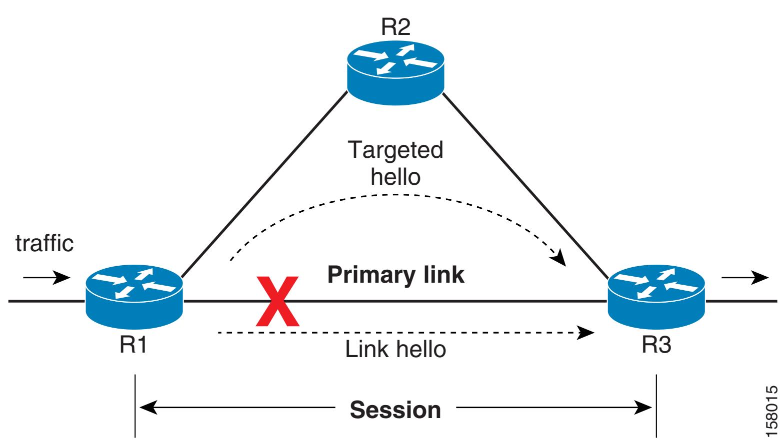
Session Protection

When a link comes up, IP converges earlier and much faster than MPLS LDP and may result in MPLS traffic loss until MPLS convergence. If a link flaps, the LDP session will also flap due to loss of link discovery. LDP session protection minimizes traffic loss and 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.

Figure 6 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 6 Session Protection



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

IGP Synchronization

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

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

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

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.

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

IGP Auto-configuration

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

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

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

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

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 the following 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 on Cisco IOS XR Software" in *Cisco IOS XR IP Addresses and Services Configuration Guide*.

How to Implement LDP on Cisco IOS XR Software

A typical MPLS LDP deployment requires coordination among several global neighbor routers. Various configuration tasks are required to implement MPLS LDP on Cisco IOS XR software, as follows:

- Configuring LDP Discovery Parameters, page MPC-13 (optional)
- Configuring LDP Discovery Over a Link, page MPC-15 (required)
- Configuring LDP Discovery for Active Targeted Hellos, page MPC-17 (required)
- Configuring LDP Discovery for Passive Targeted Hellos, page MPC-19 (required)
- Configuring Label Advertisement Control (Outbound Filtering), page MPC-21 (optional)
- Setting Up LDP Neighbors, page MPC-23 (optional)
- Setting Up LDP Forwarding, page MPC-25 (optional)
- Setting Up LDP NSF Using Graceful Restart, page MPC-27 (optional)
- Configuring Label Acceptance control (Inbound Filtering), page MPC-30 (optional)
- Configuring Local Label Allocation Control, page MPC-31 (optional)
- Configuring Session Protection, page MPC-33 (optional)
- Configuring LDP IGP Synchronization: OSPF, page MPC-34 (optional)
- Configuring LDP IGP Synchronization: ISIS, page MPC-36 (optional)
- Configuring LDP IGP Sync Delay Interval, page MPC-38 (optional)
- Enabling LDP Auto-configuration for a Specified OSPF Instance, page MPC-39 (optional)
- Enabling LDP Auto-configuration in an Area for a Specified OSPF Instance, page MPC-41
- Disabling LDP Auto-configuration, page MPC-42 (optional)
- Configuring LDP NSR, page MPC-44

Configuring LDP Discovery Parameters

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

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

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **router-id** {*type number* | *ip-address*}
4. **discovery** {**hello** | **targeted-hello**} **holdtime** *seconds*
5. **discovery** {**hello** | **targeted-hello**} **interval** *seconds*
6. **end**
   or
   commit
7. **show mpls ldp parameters**

DETAILED STEPS

	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 submode.
Step 3	**`router-id`** {*`type number`* | *`ip-address`*} **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id loopback 1`	Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface name or IP address. By default, LDP uses the global router ID (configured by the global router ID process).

Step 4	**`discovery`** {**`hello`** | **`targeted-hello`**} **`holdtime`** *`seconds`*
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# discovery hello holdtime 30`
	`RP/0/RP0/CPU0:router(config-ldp)# discovery targeted-hello holdtime 180`
		Specifies the time that a discovered neighbor is kept without receipt of any subsequent hello messages.
		• The default value for the seconds argument is 15 seconds for link hello and 90 seconds for targeted hello messages.

Step 5	**`discovery`** {**`hello`** | **`targeted-hello`**} **`interval`** *`seconds`*
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# discovery hello interval 15`
	`RP/0/RP0/CPU0:router(config-ldp)# discovery targeted-hello interval 20`
		Selects the period of time between the transmission of consecutive hello messages.
		• The default value for the seconds argument is 5 seconds for link hello messages and 10 seconds for targeted hello messages.

	Command or Action	Purpose

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• 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.

Step 7	**show mpls ldp parameters** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp parameters`	(Optional) Displays all the current MPLS LDP parameters.

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.

Prerequisites

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. **router-id** {*type number* | *ip-address*}
4. **interface** *type number*

5. **end**
   or
   commit
6. **show mpls ldp discovery**

DETAILED STEPS

	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	**`router-id`** {*type number* | *ip-address*} **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id loopback 1`	(Optional) Specifies the router ID of the local node. • In Cisco IOS XR, 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 number* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.

	Command or Action	Purpose

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit`
		Saves configuration changes.
		• 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.

Step 6	**show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	(Optional) 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.

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.

Prerequisites

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

- A 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. **router-id** {*type number* | *ip-address*}
4. **interface** *type number*
5. **end**
   or
   commit
6. **show mpls ldp discovery**

DETAILED STEPS

	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 submode.
Step 3	**router-id** [*type number* | *ip-address*] **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id loopback 1`	Specifies the router ID of the local node. In Cisco IOS XR, the router ID is specified as an interface name or IP address. By default, LDP uses the global router ID (configured by global router ID process).
Step 4	**interface** *type number* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol.

	Command or Action	Purpose

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit`
		Saves configuration changes.
		• 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.

Step 6	**show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	(Optional) 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.

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.

Prerequisites

A 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 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. **router-id** [*type number* | *ip-address*]
4. **discovery targeted-hello accept**

5. **end**
   or
   commit
6. **show mpls ldp discovery**

DETAILED STEPS

	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	**`router-id`** [*type number* | *ip-address*] **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id loopback 1`	(Optional) Specifies the router ID of the local node. • In Cisco IOS XR, the router ID is specified as an interface name or IP address. 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 tail-end node (with respect to a given MPLS TE tunnel). • You can control the targeted-hello acceptance using the **discovery targeted-hello accept** command.

	Command or Action	Purpose
Step 5	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-ospf-ar-if)# end or RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit	Saves configuration changes. • 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.
Step 6	**show mpls ldp discovery** **Example:** RP/0/RP0/CPU0:router# show mpls ldp discovery	(Optional) 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.

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.

Prerequisites

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** *interface-id*}
4. **end**
   or
   commit

DETAILED STEPS

	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** *interface-id*} **Example:** RP/0/RP0/CPU0:router(config-ldp)# label advertise **interface POS 0/1/0/0** **RP/0/RP0/CPU0:router(config-ldp)# for pfx_acl1 to peer_acl1**	Configures label advertisement as specified by one of the following arguments: • **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** *aclist* **to** *peer-acl*—Specifies neighbors that advertise and receive label advertisements.
Step 4	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-ldp)# end or RP/0/RP0/CPU0:router(config-ldp)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Setting Up LDP Neighbors

Perform this task to set up LDP neighbors.

Prerequisites

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. **interface** *type number*
4. **discovery transport-address** [*ip-address* | **interface**]
5. **end**
   or
   commit
6. **holdtime** *seconds*
7. **neighbor** *ip-address* **password** [*encryption*] *password*
8. **backoff** *initial maximum*
9. **end**
   or
   commit
10. **show mpls ldp neighbor**

DETAILED STEPS

	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 submode.

Step 3	`interface` *type number* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.

	Command or Action	Purpose

Step 4	**discovery transport-address** [*ip-address* | **interface**]
	
	Example:
	`RP/0/RP0/CPU0:router(onfig-ldp-if)# discovery transport-address 192.168.1.42`
	or
	`RP/0/RP0/CPU0:router(onfig-ldp)# discovery transport-address interface`
		Provides an alternative transport address for a TCP connection.
		• The default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.
		• The 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	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp-if)# commit`
		Saves configuration changes.
		• 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.

Step 6	**holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(onfig-ldp)# holdtime 30`	Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer. • The outgoing keepalive interval is adjusted accordingly (to make 3 keepalives in given holdtime) with a change in session holdtime value. • The 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.

	Command or Action	Purpose
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.

Step 9	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit`
		Saves configuration changes.
		• 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.

Step 10	**show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	(Optional) 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 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).

Prerequisites

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. **explicit-null**
4. **end**
   or
   commit
5. **show mpls ldp forwarding**
6. **show mpls forwarding**
7. **ping** *ip-address*

DETAILED STEPS

	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 submode.
Step 3	**`explicit-null`** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# 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).

	Command or Action	Purpose

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• 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.

Step 5	**show mpls ldp forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp forwarding`	(Optional) Displays the MPLS LDP view of installed forwarding states (rewrites).
Step 6	**show mpls forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls forwarding`	(Optional) Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).
Step 7	**ping** *ip-address* **Example:** `RP/0/RP0/CPU0:router# ping 192.168.2.55`	(Optional) Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the **show mpls forwarding** command).

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.

Prerequisites

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. **interface** {*type number*}
4. **end**
   or
   commit
5. **graceful-restart**
6. **graceful-restart forwarding-state-holdtime** *seconds*
7. **graceful-restart reconnect-timeout** *seconds*
8. **end**
   or
   commit
9. **show mpls ldp parameters**
10. **show mpls ldp neighbor**
11. **show mpls ldp graceful-restart**

Note Repeat these steps on neighboring routers.

DETAILED STEPS

	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 number* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.

	Command or Action	Purpose

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp-if)# commit`
		Saves configuration changes.
		• 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.

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(onfig-ldp)# graceful-restart forwarding state-holdtime 180`	(Optional) Specifies the length of time that forwarding can keep LDP-installed forwarding states and rewrites, and specifies when 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. • The recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer.
Step 7	**graceful-restart reconnect-timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(onfig-ldp)# graceful-restart reconnect timeout 169`	(Optional) The length of time a neighbor waits before restarting the node to reconnect before declaring an earlier graceful restart session as down. • This command is used to start a timer on the peer (upon a neighbor restart). This timer is referred to as Neighbor Liveness timer.

	Command or Action	Purpose

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit`
		Saves configuration changes.
		• 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.

Step 9	**show mpls ldp parameters** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp parameters`	(Optional) Displays all the current MPLS LDP parameters.
Step 10	**show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	(Optional) 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	**show mpls ldp graceful-restart** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp graceful-restart`	(Optional) 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. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters the 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-lbl-acpt)# 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	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)# commit`
		Saves configuration changes.
		• 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 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. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters the configuration mode.

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

	Command or Action	Purpose
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	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• 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 Session Protection

Perform this task to configure LDP session protection.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters the configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**session protection for** ***peer-acl*** **duration** ***seconds*** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# session protection for peer_acl_1 duration 60`	Configures LDP session protection for peers specified by peer-acl with a maximum duration in seconds.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• When you enter 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.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

Configuring LDP IGP Synchronization: OSPF

Perform this task to configure LDP IGP Synchronization under OSPF.

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

SUMMARY STEPS

1. **configure**
2. **router ospf** *interface-id*
3. **mpls ldp sync**
   or
   area *area-id* **mpls ldp sync**
   or
   area *area-id* **interface** *name* **mpls ldp sync**
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 100`	Enters OSPF configuration mode.

	Command or Action	Purpose
Step 3	**mpls ldp sync** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls ldp sync`	Enables LDP IGP synchronization on an interface.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf)# commit`
		Saves configuration changes.
		• When you enter 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.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

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* **interface** *name* **address-family ipv4 unicast**
3. **mpls ldp sync** [**level 1- 2**]
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure`** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**`router isis`** *`instance-id`* **`interface`** *`name`* **`address-family ipv4 unicast`** **Example:** `RP/0/RP0/CPU0:router(config)# router isis 100 interface POS 0/2/0/0 address-family ipv4 unicast`	Enters ISIS interface configuration mode for the IPv4 unicast address family.
Step 3	**`mpls ldp sync`** **Example:** `RP/0/RP0/CPU0:router(config-isis-if-af)# mpls ldp sync`	Enables LDP IGP synchronization.

Step 4	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-isis-if-af)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-isis-if-af)# commit`
		Saves configuration changes.
		• When you enter 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.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

Configuring LDP IGP Sync Delay Interval

Perform this task to configure the LDP IGP synchronization delay interval.

By default, LDP does not delay declaring sync up as soon as convergence conditions are met.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **igp sync delay** *seconds*
4. **end**
   or
   commit

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**igp sync delay** ***seconds*** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# igp sync delay 30`	Configures LDP IGP sync delay in seconds.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• When you enter 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.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

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 (see Disabling LDP Auto-configuration, page MPC-42).

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-id*
6. **end**
   or
   commit

DETAILED STEPS

	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`	Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.
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. The *area-id* argument is specified as either a decimal value or an IP address.
Step 5	**`interface`** *`type interface-id`* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`	Enables LDP auto-configuration on the specified interface. **Note** LDP configurable limit for maximum number of interfaces does not apply to IGP auto-configuration interfaces.

Step 6	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling LDP 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 (see, Disabling LDP Auto-configuration, page MPC-42).

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-id*
6. **end**
   or
   commit

DETAILED STEPS

	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`	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)# router ospf`	Configures an OSPF area and identifier. The *area-id* argument is specified as either a decimal value or an IP address.
Step 4	**mpls ldp auto-config** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls ldp auto-config`	Enables LDP auto-configuration.

	Command or Action	Purpose
Step 5	**interface** *type interface-id* **Example:Example:** `RP/0/RP0/CPU0:router(config-ospf)# interface pos 0/6/0/0`	Enables LDP auto-configuration on the specified interface. **Note** LDP configurable limit for maximum number of interfaces does not apply to IGP auto-config interfaces.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf)# commit`
		Saves configuration changes.
		• 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-id*
4. **igp auto-config disable**
5. **end**
   or
   commit

DETAILED STEPS

	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	**interface** *type interface-id* **Example:Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface pos 0/6/0/0`	Enters interface configuration mode and configures an interface.
Step 4	**igp auto-config disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# igp auto-config disable`	Disables auto-configuration on the specified interface.

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp-if)# commit`
		Saves configuration changes.
		• 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 LDP NSR

Perform this task to configure LDP NSR (see LDP Nonstop Routing, page MPC-12).

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

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **nsr**
4. **end**
   or
   commit
5. **show mpls ldp nsr statistics**

DETAILED STEPS

	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	**nsr** **Example:Example:** `RP/0/RP0/CPU0:router(config-ldp)# nsr`	Enables LDP nonstop routing.

	Command or Action	Purpose

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ldp)# commit`
		Saves configuration changes.
		• 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.

Step 5	**show mpls ldp nsr statistics** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# igp auto-config disable`	Shows MPLS LDP NSR statistics.

Configuration Examples for Implementing LDP

This section provides the following configuration examples:

Configuring LDP with Graceful Restart: Example

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

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

Configuring LDP Discovery: Example

The following example shows how to configure LDP discovery parameters:

```
mpls ldp
 router-id loopback0
 discovery hello holdtime 15
 discovery hello interval 5
!

show mpls ldp parameters
show mpls ldp discovery
```

Configuring LDP Link: Example

The following example shows how to configure LDP link parameters:

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

show mpls ldp discovery
```

Configuring LDP Discovery for Targeted Hellos: Example

The following example shows how to configure LDP Discovery to accept targeted hello messages:

Active (tunnel head)

```
mpls ldp
 router-id loopback0
 interface tunnel-te 12001
 !
!
```

Passive (tunnel tail)

```
mpls ldp
 router-id loopback0
 discovery targeted-hello accept
!
```

Configuring Label Advertisement (Outbound Filtering): Example

The following example shows how to configure LDP label advertisement control:

```
mpls ldp
    label
        advertise
            disable
            for pfx_acl_1 to peer_acl_1
            for pfx_acl_2 to peer_acl_2
            for pfx_acl_3
            interface POS 0/1/0/0
            interface POS 0/2/0/0
        !
    !
!
ipv4 access-list pfx_acl_1
    10 permit ip host 1.0.0.0 any
!
ipv4 access-list pfx_acl_2
    10 permit ip host 2.0.0.0 any
!
ipv4 access-list peer_acl_1
    10 permit ip host 1.1.1.1 any
    20 permit ip host 1.1.1.2 any
!
ipv4 access-list peer_acl_2
    10 permit ip host 2.2.2.2 any
!

show mpls ldp binding
```

Configuring LDP Neighbors: Example

The following example shows how to disable label advertisement:

```
mpls ldp
    router-id Loopback0
    neighbor 1.1.1.1 password encrypted 110A1016141E
    neighbor 2.2.2.2 implicit-withdraw
!
```

Configuring LDP Forwarding: Example

The following example shows how to configure LDP forwarding:

```
mpls ldp
 explicit-null
!

show mpls ldp forwarding
show mpls forwarding
```

Configuring LDP Nonstop Forwarding with Graceful Restart: Example

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

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

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

Configuring Label Acceptance (Inbound Filtering): Example

The following example shows how to configure inbound label filtering:

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

Configuring Local Label Allocation Control: Example

The following example shows how to configure local label allocation control:

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

Configuring LDP Session Protection: Example

The following example shows how to configure session protection:

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

Configuring LDP IGP Synchronization - OSPF: Example

The following example shows how to configure LDP IGP synchronization:

```
router ospf 100
mpls ldp sync
!
```

```
mpls ldp
 igp sync delay 30
!
```

Configuring LDP IGP Synchronization - ISIS: Example

The following example shows how to configure LDP IGP synchronization:

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

Configuring LDP Auto-configuration: Example

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

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

The following 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 interface pos 1/1/1/1
```

Additional References

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

Related Documents

Related Topic	Document Title
Cisco IOS XR LDP commands	*MPLS Label Distribution Protocol Commands on Cisco IOS XR Software*
Cisco CRS-1 router getting started material	Cisco IOS XR *Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*

Standards

Standards[1]	Title
Technical Assistance Center (TAC) home page, containing 30,000 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.	—

1. Not all supported standards are listed.

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[1]	Title
RFC 3031	*Multiprotocol Label Switching Architecture*
RFC 3036	*LDP Specification*
RFC 3037	LDP Applicability
RFC 3478	Graceful Restart Mechanism for Label Distribution Protocol
RFC3815	Definitions of Managed Objects for MPLS LDP

1. Not all supported RFCs are listed.

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