

LDP Operation

1) LDP Discovery:

There are two types of LDP discoveries-

a) A basic discovery mechanism used to discover directly connected LSR neighbors

In basic discovery mechanism, an LSR sends out periodic Hello messages as UDP packets to “all routers on this subnet” i.e. 224.0.0.2 to discover neighbors. The Hello messages contain the LDP Identifier (Router ID: Label Space) it intends to use with other information. The LDP ID is a 6-byte value with 4-byte Router ID and 2-byte Label Space.

The 4-byte Router ID is configurable. By default, the Router ID is the highest Loopback interface on the router or in case the Loopback interface is not present, the highest active interface. Router ID can be configured using ***mpls ldp router-id <intf-type> <intf-number> [force]*** command.

The router determines the LDP router ID as follows, if the ***mpls ldp router-id*** command is not executed,

1. The router examines the IP addresses of all operational interfaces.
2. If these IP addresses include loopback interface addresses, the router selects the largest loopback address as the LDP router ID.
3. Otherwise, the router selects the largest IP address pertaining to an operational interface as the LDP router ID.

The normal (default) method for determining the LDP router ID may result in a router ID that is not usable in certain situations. For example, the router might select an IP address as the LDP router ID that the routing protocol cannot advertise to a neighboring router. The ***mpls ldp router-id*** command allows you to specify the IP address of an interface as the LDP router ID. Make sure the specified interface is operation also that its IP address can be used as the LDP router ID.

When you issue the ***mpls ldp router-id*** command without the ***force*** keyword, the router select selects the IP address of the specified interface (provided that the interface is operational) the next time it is necessary to select an LDP router ID, which is typically the next time the interface is shut down or the address is configured.

When you issue the ***mpls ldp router-id*** command with the ***force*** keyword, the effect of the ***mpls ldp router-id*** command depends on the current state of the specified interface:

- If the interface is up (operational) and if its IP address is not currently the LDP router ID, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears

down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

- If the interface is down (not operational) when the **mpls ldp router-id interface force** command is issued, when the interface transitions to up, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

The 2-byte Label Space indicates per-platform or per-interface label space. If the value is 0, it indicates per-platform label space. If the value is non-zero, it indicates per-interface label space.

To view LDP discoveries on all interfaces, use *show mpls ldp discovery [detail]* command. The output also shows the Hello and Holdtime intervals.

The *show mpls interfaces* command provides a concise view of which interfaces are running LDP.

To change the interval between sending Hello messages or change the Holdtime interval, use *mpls ldp discovery {hello {holdtime / interval}}* command from global configuration mode. The default Hello interval is 5 seconds. The default Holdtime is 15 seconds and default interval value is 5 seconds.

If two LDP peers are configured with different Holdtime values, then the smallest value of the two is used between them.

b) An extended discovery mechanism used to discover non-directly connected LSR neighbors

In extended discovery mechanism, an LSR sends out Targeted Hello (unicast) messages as UDP packets to specific address of the targeted LSR.

The targeted LDP neighborhood can be configured using *mpls ldp neighbor <ip-address> targeted ldp* command from global configuration mode.

2) LDP Session Establishment:

The exchange of LDP Hello messages between two LSRs (LSR1 and LSR2) results in Hello adjacency which triggers LDP Session establishment. It is a two step process-

A) Transport connection establishment

- a) LSR1 attempts to open a TCP session with LSR2. It will use the source address as the transport address that it used to advertise Hello messages to LSR2. Or it will use the transport address as the one mentioned using **mpls ldp discovery transport-address <interface name | IP Address>** command. These addresses will be advertised in Hello messages.

- b) LSR1 will determine whether it will play active or passive role by comparing the source address of itself and LSR2 used for TCP connection. The LSR with higher IP address becomes the active LSR and initiates the TCP connection.
- c) If LSR1 is active, it will attempt to establish a TCP connection by connecting to well-known TCP port (port 646) at transport-address of LSR2. If LSR1 is passive, it waits for LSR2 to open a TCP connection with itself.

B) Session Initialization

After LSR1 and LSR2 establish a TCP connection, they negotiate various session parameters using Initialization messages. The session parameters include LDP protocol version, label distribution method, timer values, VPI/VCI values for LC-ATM, DLCI range for label controlled FR, etc.

If LSR1 is the active LSR, it initiates negotiation of session parameters by sending Initialization message to LSR2. If LSR1 is passive, it waits for LSR2 to initiate negotiation of session parameters. The Initialization message carries the LDP Identifier of the active LSR and the passive LSR. If there are multiple links between LSR1 and LSR2 with different label spaces, the passive LSR would not know which label space to advertise over newly established TCP connection.

If LSR1 is a passive LSR-

- i) LSR1 receives Initialization message from LSR2, it compares the LDP Identifier with a Hello adjacency.
- ii) If it finds a matching Hello adjacency, the adjacency specifies the local label space for the session. Next, it checks whether the session parameters are acceptable. If yes, it replies with an Initialization message of its own with its own session parameters and a Keepalive message to notify that it accepts the session parameters. If the parameters are not acceptable, it responds with a **Session Rejected/ Parameters Error Notification message** and closes the TCP connection.
- iii) If it cannot find a matching Hello adjacency, it sends a **Session Rejected/ No Hello Error Notification message** to LSR2 and closes the TCP connection.
- iv) If LSR1 receives a Keepalive message in response to its Initialization message, the LDP session is operational.
- v) If LSR1 receives an Error Notification message, LSR2 has rejected LSR1's session parameters and LSR1 closes the TCP session.

If LSR1 is an active LSR,

- i) If LSR1 receives an Error Notification message, LSR2 has rejected its session parameters and LSR1 closes its TCP session.

ii) If LSR1 receives an Initialization message, it checks for session parameters. If they are acceptable, it sends a Keepalive message; otherwise, replies with a Session Rejected/ Parameters Error Notification message and closes the TCP connection.

iii) If LSR1 receives a Keepalive message, a session is operational.

An LSR throttles its session setup attempts with an exponential backoff in situations when Initialization messages are negatively acknowledged. In Cisco IOS, the throttled rate can be controlled using ***mpls ldp backoff <initial-backoff> <maximum-backoff>*** command. By default, initial-backoff value is 15 seconds and maximum-backoff value is 120 seconds.

3) Maintaining Hello Adjacencies:

LDP uses the regular receipt of LDP Discovery Hello messages to indicate a peer's intent to use a label space identified by a Hello. An LSR maintains a Hold Timer for each Hello adjacency which it resets with every arrival of Hello for that particular adjacency.

If the timer expires, LDP concludes that the peer does not wish to label switch with that label space or the peer is dead. The LSR then deletes the Hello adjacency which results in the LSR terminating the LDP session by sending a Notification message and closing the TCP connection.

4) Maintaining LDP Sessions:

After an LDP session is established, an LSR maintains the integrity of the session by sending Keepalive messages every 60 seconds by default. The Keepalive timer for each peer session resets whenever it receives any packet on that session. If the Keepalive timer expires (180 seconds by default), LDP concludes that the TCP connection is bad or the peer is dead.

An LSR can terminate the LDP session at any stage by sending a Shutdown message to its peer.

The LDP session holdtime value can be changed using ***mpls ldp holdtime <value>*** command. The default is 180 seconds.

5) Label Distribution and Management:

a) **Label distribution methods-** There are two types of Label distribution methods-

- Unsolicited Downstream- An LSR distributes label bindings to other LSRs that have not explicitly requested them.

- Downstream-on-demand- An LSR requests a label binding for a particular FEC from its next-hop LSR.

An LSR may run both these methods as it depends on the type of interfaces which are supported by a particular implementation.

b) **Label Retention modes**- There are two types of label retention modes-

- Liberal- An LSR retains all the label bindings for a particular FEC that it receives from different LSRs which are not its next-hop LSRs.

- Conservative- An LSR discards label bindings that are received from LSRs that are not its next-hop.

Liberal label retention mode adapts quickly to routing changes while conservative label retention mode requires an LSR to maintain fewer labels.

c) **Label Distribution Control modes**- There are two types of modes-

- Independent Label Distribution Control mode- When using Independent LSP control, an LSR can advertise label bindings to its next-hop LSRs at any time.

- Ordered Label Distribution Control mode- An LSR may initiate advertising label bindings to its neighbor LSRs only if it has received a label binding for that particular FEC from its downstream LSR, or if it is the egress LSR for that FEC.

6) LDP Identifier and Next-hop addresses:

An LSR maintains learned labels in a LIB. An LIB entry for a prefix associates a collection of (LDP Identifier, label) pairs with the prefix; one such pair for each peer advertising for that particular prefix.

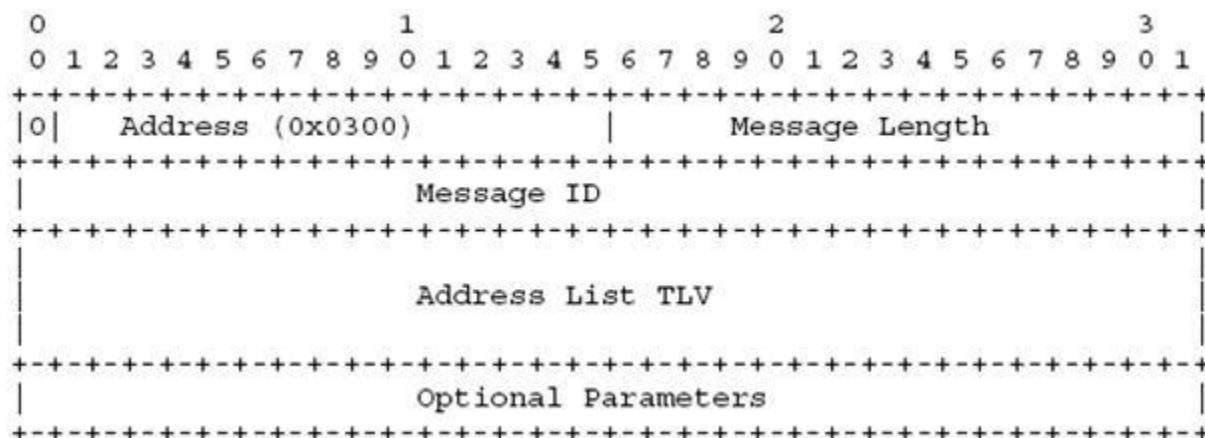
The LIB information can be viewed using *show mpls ldp bindings* command.

To enable an LSR to map between a peer LDP identifier and peer's addresses, LSRs advertise their addresses using **Address messages**. An LSR uses **Address Withdraw messages** to withdraw any addresses previously advertised to a peer.

Address messages and Address Withdraw messages:

When a new LDP session is established and before sending Label Mapping and Label Request messages, an LSR must advertise its interface addresses in one more Address messages. These interface addresses are called **Bounded Addresses**.

To view bounded addresses, use *show mpls ldp neighbor detail* command.



Message ID: 32-bit value used to identify this message

Address List TLV: The List of interface addresses being advertised by the sending LSR

If an LSR “deactivates” an interface, it must advertise an Address Withdraw message to withdraw the address.



Message ID: 32-bit value used to identify this message

Address List TLV: The List of interface addresses to be withdrawn by the sending LSR

7) Loop Detection:

Loop detection can be configured on LSRs for finding looping LSPs and prevent Label Request/ Label Mapping messages from looping. Loop detection on Cisco LSRs can be configured from global configuration mode using ***mpls ldp loop-detection*** command.

The mechanism uses Path Vector and Hop Count TLVs carried by Label Request and Label Mapping messages.

A Path Vector TLV contains a list of LSRs that its containing message has traversed. The LSR is identified by the first 4 octets of the LDP Identifier. When an LSR propagates a message containing Path Vector TLV, it adds its LSR ID to the Path Vector list. An LSR that receives a message that contains the Path Vector list with its LSR ID, interprets that a loop is detected.

A Hop Count TLV contains a count of the LSRs the containing message has traversed. The LSRs increment the hop-count when they propagate the message containing the HOP Count TLV. An LSR that detects the Hop Count has reached a configured value, interprets that the message has traversed the loop.

To configure maximum allowable hops, use *mpls ldp maxhops <value>* command from global configuration mode. The value range is 0 – 254.

Label Request Message-

Rules governing the HOP Count TLV in a Label Request Message for loop detection-

- a) A Label Request Message must contain a Hop Count TLV.
- b) If an LSR sending a Label Request message is ingress LSR for an FEC, it must set the hop count to 1.
- c) If an LSR has received a Label Request message from its upstream LSR with Hop Count TLV, it must increment the hop count by 1 and pass the resulting value in a Hop Count TLV to its next hop along with the Label Request message.

Rules governing the Path Vector TLV in a Label Request message for loop detection-

- a) If an LSR sending the Label Request message is ingress LSR for an FEC, it must include the Path Vector TLV of 1 with its LSR ID.
- b) If an LSR has received a Label Request message from its upstream LSR, it must include its own LSR ID and pass it to its downstream LSR; and if the incoming Label Request has no Path Vector TLV, this LSR must include the Path Vector with its LSR ID.

If the LSR receives a Label Request message with Hop Count TLV equal to maximum allowable hops by this LSR or Path Vector TLV containing its own LSR ID, it detects that the message has traversed a loop. It drops the packet and sends a **Loop Detected Notification message** back to the source of Label Request message.

Label Mapping message-

Rules governing the HOP Count TLV in a Label Mapping Message for loop detection-

- a) A Label Mapping message should contain Hop Count TLV

- b) If the LSR is the egress LSR, the hop count value is set to 1.
- c) If an LSR is at the edge of a domain where the domain-members cannot decrement TTL (e.g. ATM LSR or FR LSR domain), the LSR should reset the hop count value to 1 before propagating the message. If not, the LSR should increment the hop count received from the next-hop before propagating to its upstream LSR.
- d) If an LSR is running Independent label control mode, it has no knowledge of the hop-count and hence set the hop-count value to 0 which is interpreted as “unknown”.

Rules governing the Path Vector TLV in a Label Mapping message for loop detection-

- a) If an LSR is the egress LSR, the Label Mapping message need not include the Path Vector TLV.
- b) If an LSR receives a Label Mapping message not containing Path Vector TLV, it includes the Path Vector TLV of length 1 with its own LSR ID.

If the LSR receives a Label Mapping message with Hop Count TLV equal to maximum allowable hops by this LSR or Path Vector TLV containing its own LSR ID, it detects that the message has traversed a loop. It drops the packet and sends a **Loop Detected Notification message** back to the source of Label Mapping message.

If loop-detection is desired, then it should be enabled on all LSRs within the MPLS domain.

Further Reading:

- RFC 3036: LDP Specifications <http://www.faqs.org/ftp/rfc/pdf/rfc3036.txt.pdf>
- RFC 3037: LDP Applicability <http://4lang.com/rfcs/rfc3037.html>
- RFC 3031: MPLS Architecture <http://www.faqs.org/ftp/rfc/pdf/rfc3031.txt.pdf>

Source Credit: Cisco.com, <https://sites.google.com/site/amitsciscozone/>