MPLS Label Distribution Protocol MIB Version 8 Upgrade

The Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) MIB Version 8 Upgrade feature enhances the LDP MIB to support the Internet Engineering Task Force (IETF) draft version 8.

Feature History for MPLS Label Distribution Protocol MIB Version 8 Upgrade

Release	Modification
12.0(11)ST	This feature was introduced to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series routers.
12.2(2)T	This feature was added to this release to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers.
12.0(21)ST	This feature was added to this release to provide SNMP agent and LDP notification support when using the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series Internet routers.
12.0(22)S	This feature (Version 1) was integrated into Cisco IOS Release 12.0(22)S.
12.0(24)S	This feature was upgraded to Version 8 in Cisco IOS Release 12.0(24)S.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for MPLS LDP MIB Version 8 Upgrade

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switch routers (LSRs).
- MPLS must be enabled on the LSRs.
- LDP must be enabled on the LSRs.

Restrictions for MPLS LDP MIB Version 8 Upgrade

This implementation of the MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object mplsLdpSessionUpDownTrapEnable, which, for purposes of this release, has been extended to be writable by the SNMP agent.

Setting this object to a value of true enables both the mplsLdpSessionUp and mplsLdpSessionDown notifications on the LSR; conversely, setting this object to a value of false disables both of these notifications.

For a description of notification events, see the "Events Generating MPLS LDP MIB Notifications in the MPLS LDP MIB Version 8 Upgrade" section on page 11.

Most MPLS LDP MIB objects are set up automatically during the LDP peer discovery (hello) process and the subsequent negotiation of parameters and establishment of LDP sessions between the LDP peers.

The following tables are not implemented in this release:

- mplsLdpEntityFrParmsTable
- mplsLdpEntityConfFrLRTable
- mplsLdpFrameRelaySesTable
- mplsFecTable
- mplsLdpSesInLabelMapTable
- mplsXCsFecsTable
- mplsLdpSesPeerAddrTable

Information About the MPLS LDP MIB Version 8 Upgrade

To configure the MPLS LDP MIB Version 8 Upgrade, you need to understand the following concepts:

Feature Design of the MPLS LDP MIB Version 8 Upgrade

Multiprotocol Label Switching (MPLS) is a packet forwarding technology that uses a short, fixed-length value called a label in packets to determine the next hop for packet transport through an MPLS network by means of label switch routers (LSRs).

A fundamental MPLS principle is that LSRs in an MPLS network must agree on the definition of the labels being used for packet forwarding operations. Label agreement is achieved in an MPLS network by means of procedures defined in the Label Distribution Protocol (LDP).

LDP operations begin with a discovery (hello) process during which an LDP entity (a local LSR) finds a cooperating LDP peer in the network and negotiates basic operating procedures between them. The recognition and identification of a peer by means of this discovery process results in a hello adjacency, which represents the context within which label binding information is exchanged between the local LSR and its LDP peer. LDP functionality then creates an active LDP session between the two LSRs to effect the exchange of label binding information. The result of this process, when carried to completion with respect to all the LSRs in an MPLS network, is a label-switched path (LSP), which constitutes an end-to-end packet transmission pathway between the communicating network devices.

By means of LDP, LSRs can collect, distribute, and release label binding information to other LSRs in an MPLS network, thereby enabling the hop-by-hop forwarding of packets in the network along normally routed paths.

The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco IOS. Providing this capability requires SNMP agent code to execute on a designated network management station (NMS) in the network. The NMS serves as the medium for user interaction with the network management objects in the MPLS LDP MIB.

The SNMP agent embodies a layered structure that is compatible with Cisco IOS and presents a network administrative and management interface to the objects in the MPLS LDP MIB and, thence, to the rich set of label switching capabilities supported by Cisco IOS.

By means of an SNMP agent, you can access MPLS LDP MIB objects using standard SNMP GET operations to accomplish a variety of network management tasks. All the objects in the MPLS LDP MIB follow the conventions defined in the IETF draft MIB entitled *draft-ietf-mpls-ldp-mib-08.txt,* which defines network management objects in a structured and standardized manner. This draft MIB is continually being evolved toward the status of a standard. Accordingly, the MPLS LDP MIB will be implemented in a manner that tracks the evolution of this IETF document.

Slight differences that exist between the IETF draft MIB and the implementation of equivalent functions in Cisco IOS require some minor translations between the MPLS LDP MIB objects and the internal data structures of Cisco IOS. Such translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low priority process.

The extensive label switching capabilities supported in Cisco IOS provide an integrated approach to managing the large volumes of traffic carried by WANs. These capabilities are integrated into the Layer 3 network services, thus optimizing the routing of high-volume traffic through Internet service provider backbones while, at the same time, ensuring the resiliency of the network to link or node failures.

This release of Cisco IOS supports the following functionality in relation to the MPLS LDP MIB:

- Tag Distribution Protocol (TDP)
- Generation and sending of event notification messages to signal changes in the status of LDP sessions
- Enabling and disabling of event notification messages by means of extensions to existing SNMP CLI commands
- Specification of the name or the IP address of an NMS workstation in the operating environment to which Cisco IOS event notification messages are to be sent to serve network administrative and management purposes
- Storage of the configuration pertaining to an event notification message into NVRAM of the NMS

The structure of the MPLS LDP MIB conforms to Abstract Syntax Notation One (ASN.1), thereby forming a highly structured and idealized database of network management objects.

Using any standard SNMP application, you can retrieve and display information from the MPLS LDP MIB by means of standard SNMP GET operations; similarly, you can traverse and display information in the MIB by means of SNMP GETNEXT operations.

Note Because the MPLS LDP MIB was not given an Internet Assigned Numbers Authority (IANA) Experimental object identifier (OID) at the time of its implementation, Cisco chose to implement the MIB under the ciscoExperimental OID number, as follows:

ciscoExperimental 1.3.6.1.4.1.9.10

mplsLdpMIB 1.3.6.1.4.1.9.10.65

If the MPLS LDP MIB is assigned an IANA Experimental OID number, Cisco will replace all objects in the MIB under the ciscoExperimental OID and reposition the objects under the IANA Experimental OID.

Enhancements in Version 8 of the MPLS LDP MIB

Version 8 of the MPLS LDP MIB contains the following enhancements since the release of Version 1:

- TDP support
- Upgraded objects
- New indexing that is no longer based on the number of sessions

Benefits of the MPLS LDP MIB Version 8 Upgrade

- Supports TDP and LDP
- Establishes LDP sessions between peer devices in an MPLS network

- Retrieves MIB parameters relating to the operation of LDP entities, such as:
  - Well-known LDP discovery port
  - Maximum transmission unit (MTU)
  - Proposed keepalive timer interval
  - Loop detection
  - Session establishment thresholds
  - Range of virtual path identifier/virtual channel identifier (VPI/VCI) pairs to be used in forming labels
- Gathers statistics related to LDP operations, such as error counters (Table 5)
- Monitors the time remaining for hello adjacencies
- Monitors the characteristics and status of LDP peers, such as:
  - Internetwork layer address of LDP peers
  - Loop detection of the LDP peers
  - Default MTU of the LDP peer
  - Number of seconds the LDP peer proposes as the value of the keepalive interval
- Monitors the characteristics and status of LDP sessions, such as:
  - Displaying the error counters (Table 10)
  - Determining the LDP version being used by the LDP session
  - Determining the keepalive hold time remaining for an LDP session
  - Determining the state of an LDP session (whether the session is active or not)
  - Displaying the label ranges (Table 2) for platform-wide and interface-specific sessions
  - Displaying the ATM parameters (Table 3)

Description of MPLS LDP MIB Elements for the MPLS LDP MIB Version 8 Upgrade

LDP operations related to an MPLS LDP MIB involve the following functional elements:

- LDP Entity—Relates to an instance of LDP for purposes of exchanging label spaces; describes a potential session.
- LDP Peer—Refers to a remote LDP entity (that is, a nonlocal LSR).
- LDP Session—Refers to an active LDP process between a local LSR and a remote LDP peer.
- Hello Adjacency—Refers to the result of an LDP discovery process which affirms the state of two LSRs in an MPLS network as being adjacent to each other (that is, as being LDP peers). A hello adjacency constitutes the working context between two LSRs in an MPLS network for purposes of exchanging label binding information.

These MPLS LDP MIB elements are briefly described under separate headings below.

In effect, the MPLS LDP MIB provides a network management database that supports real-time access to the various MIB objects within, reflecting the current state of MPLS LDP operations in the network. This network management information database is accessible by means of standard SNMP commands issued from an NMS in the MPLS LDP operating environment.

The MPLS LDP MIB supports the following network management and administrative activities:

- Retrieving MPLS LDP MIB parameters pertaining to LDP operations
- Monitoring the characteristics and the status of LDP peers
- Monitoring the status of LDP sessions between LDP peers
- Monitoring hello adjacencies in the network
- Gathering statistics regarding LDP sessions

LDP Entities

An LDP entity is uniquely identified by an LDP identifier that consists of the mplsLdpEntityLdpId and the mplsLdpEntityIndex as shown in Figure 1. The mplsLdpEntityLdpId consists of the local LSR ID (four octets) and the label space ID (two octets). The mplsLdpEntityIndex consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the peer LSR. The label space ID identifies a specific label space available within the LSR.

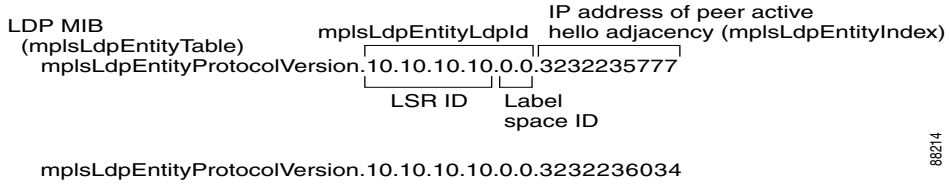
The mplsldpEntityProtocolVersion is a sample object from the mplsLdpEntityTable.

Figure 1 shows the following indexing:

- mplsLdpEntityLdpId = 10.10.10.10.0.0
- LSR ID = 10.10.10.10
- Label space ID = 0.0

  The mplsLdpEntityLdpId or the LDP ID consists of the LSR ID and the label space ID.
- The IP address of peer active hello adjacency or the mplsLdpEntityIndex = 3232235777, which is the 32-bit representation of the IP address assigned to the peer's active hello adjacency.

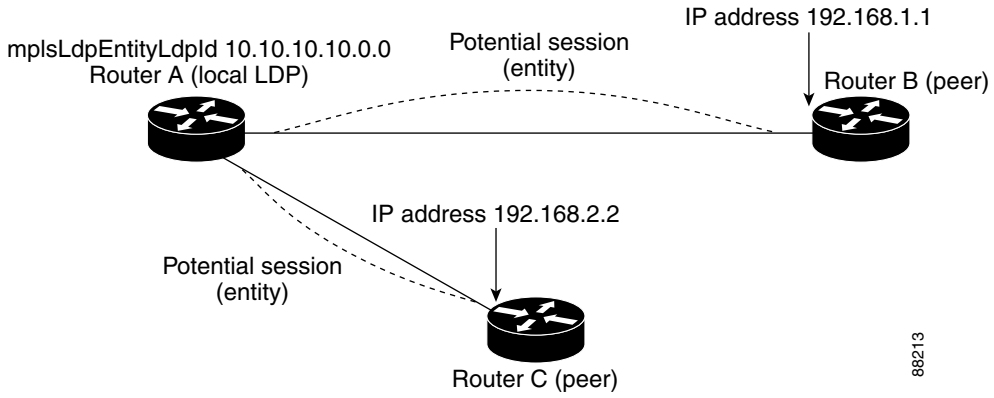
Figure 1 Sample Indexing for an LDP Entity



An LDP entity represents a label space that has the potential for a session with an LDP peer. An LDP entity is set up when a hello adjacency receives a hello message from an LDP peer.

In Figure 2, Router A has potential sessions with two remote peers, Routers B and C. The mplsLdpEntityLdpId = 10.10.10.10.0.0, and the IP address of the peer active hello adjacency (mplsLdpEntityIndex) = 3232235777, which is the 32-bit representation of the IP address 192.168.1.1 for Router B.

Figure 2 LDP Entity



LDP Sessions and Peers

LDP sessions exist between local entities and remote peers for the purpose of distributing label spaces. There is always a one-to-one correspondence between an LDP peer and an LDP session. A single LDP session is a label distribution protocol instance that communicates across one or more network links with a single LDP peer.

LDP supports the following types of sessions:

- Interface-specific—An interface-specific session uses interface resources for label space distributions. For example, each label-controlled ATM (LC-ATM) interface uses its own VPIs/VCIs for label space distributions. Depending on its configuration, an LDP platform can support zero, one, or more interface-specific sessions. Each LC-ATM interface has its own interface-specific label space and a non-zero label space ID.
- Platform-wide—An LDP platform supports a single platform-wide session for use by all interfaces that can share the same global label space. For Cisco platforms, all interface types except LC-ATM use the platform-wide session and have a label space ID of zero.

When a session is established between two peers, entries are created in the mplsLdpPeerTable and the mplsLdpSessionTable because they have the same indexing.

In Figure 3, Router A has two remote peers, Routers B and C. Router A has a single platform-wide session that consists of two serial interfaces with Router B and another platform-wide session with Router C. Router A also has two interface-specific sessions with Router B.

Figure 3 LDP Session

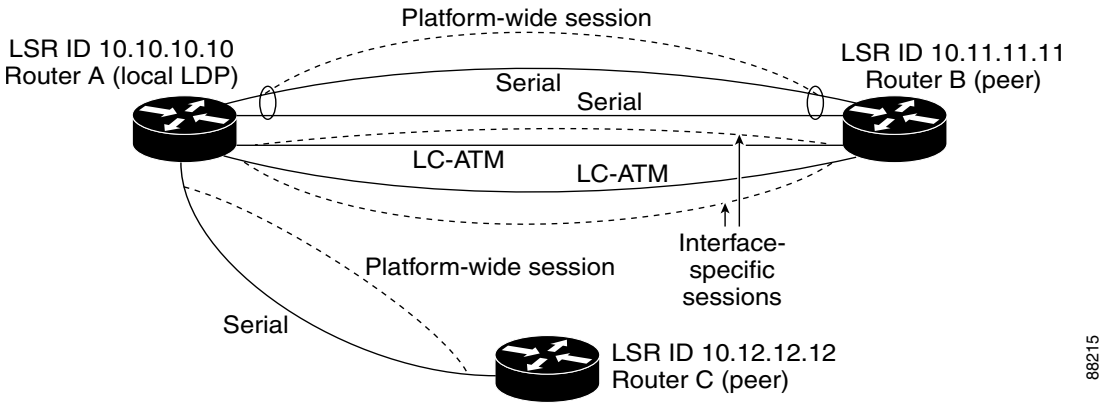


Figure 4 shows entries that correspond to the mplsLdpPeerTable and the mplsLdpSessionTable in Figure 3.

In Figure 4, mplsLdpSesState is a sample object from the mplsLdpSessionTable on Router A. There are four mplsLdpSesState sample objects shown (top to bottom). The first object represents a platform-wide session associated with two serial interfaces. The next two objects represent interface-specific sessions for the LC-ATM interfaces on Routers A and B. Note that these interface-specific sessions have non-zero peer label space IDs. The last object represents a platform-wide session for the next peer, Router C.

The indexing is based on the entries in the mplsLdpEntityTable. It begins with the indexes of the mplsLdpEntityTable and adds the following:

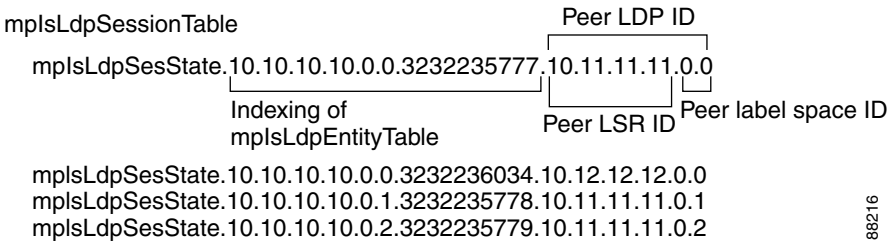
- Peer LDP ID = 11.11.11.11.0.0

  The peer LDP ID consists of the peer LSR ID (four octets) and the peer label space ID (two octets).

- Peer LSR ID = 11.11.11.11
- Peer label space ID = 0.0

  The peer label space ID identifies a specific peer label space available within the LSR.

Figure 4 Sample Indexing for an LDP Session



LDP Hello Adjacencies

An LDP hello adjacency is a network link between a router and its peers. The purpose of an LDP hello adjacency is to exchange label binding information between two LSRs that are adjacent to each other in a network.

An LDP hello adjacency exists for each link on which LDP runs. Multiple LDP hello adjacencies exist whenever there is more than one link in a session between a router and its peer, such as in a platform-wide session.

A hello adjacency is considered active if it is currently engaged in a session, or nonactive if it is not currently engaged in a session.

A targeted hello adjacency is not directly connected to its peer and has an unlimited number of hops between itself and its peer. A linked hello adjacency is directly connected between two routers.

In Figure 5, Router A has two remote peers, Routers B and C. Router A has a platform-wide session that consists of three serial interfaces, one of which is active, with Router B and another platform-wide (targeted) session with Router C.

Figure 5 Hello Adjacency

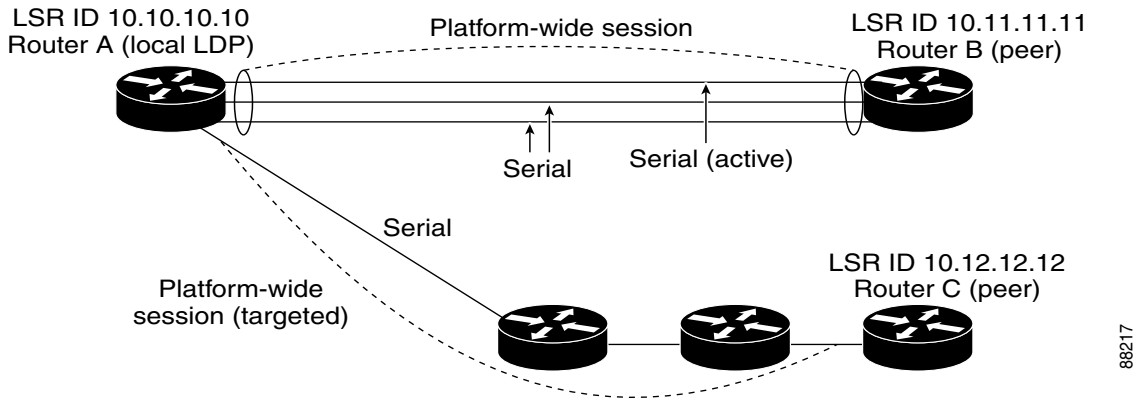
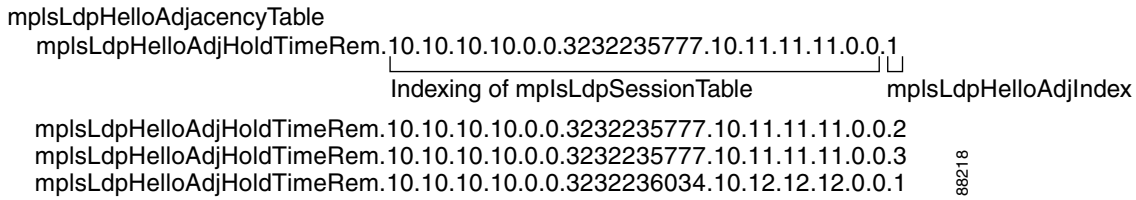


Figure 6 shows entries in the mplsLdpHelloAdjacencyTable. There are four mplsLdpHelloAdjHoldTime sample objects (top to bottom). They represent the two platform-wide sessions and the four serial links shown in Figure 5.

The indexing is based on the mplsLdpSessionTable. When the mplsLdpHelloAdjIndex enumerates the different links within a single session, the active link is mplsLdpHelloAdjIndex = 1.

Figure 6 Sample Indexing for an LDP Hello Adjacency



Events Generating MPLS LDP MIB Notifications in the MPLS LDP MIB Version 8 Upgrade

When you enable MPLS LDP MIB notification functionality by issuing the **snmp-server enable traps mpls ldp** command, notification messages are generated and sent to a designated NMS in the network to signal the occurrence of specific events within Cisco IOS.

The MPLS LDP MIB objects involved in LDP status transitions and event notifications include the following:

- mplsLdpSessionUp—This message is generated when an LDP entity (a local LSR) establishes an LDP session with another LDP entity (an adjacent LDP peer in the network).
- mplsLdpSessionDown—This message is generated when an LDP session between a local LSR and its adjacent LDP peer is terminated.
- mplsLdpPathVectorLimitMismatch—This message is generated when a local LSR establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits.

  The value of the path vector limit can range from 0 through 255; a value of 0 indicates that loop detection is off; any value other than zero up to 255 indicates that loop detection is on and, in addition, specifies the maximum number of hops through which an LDP message can pass before a loop condition in the network is sensed.

  We recommend that all LDP-enabled routers in the network be configured with the same path vector limit. Accordingly, the mplsLdpPathVectorLimitMismatch object exists in the MPLS LDP MIB to provide a warning message to the NMS when two routers engaged in LDP operations have a dissimilar path vector limit.

Note This notification is generated only if the distribution method is downstream-on-demand.

- mplsLdpFailedInitSessionThresholdExceeded—This message is generated when a local LSR and an adjacent LDP peer attempt to set up an LDP session between them, but fail to do so after a specified number of attempts. The default number of attempts is 8. This default value is implemented in Cisco IOS and cannot be changed using either the command-line interface (CLI) or an SNMP agent.

  Eight failed attempts to establish an LDP session between a local LSR and an LDP peer, due to any type of incompatibility between the devices, causes this notification message to be generated.

  In general, Cisco routers support the same features across multiple platforms. Therefore, the most likely incompatibility to occur between Cisco LSRs is a mismatch of their respective ATM VPI/VCI label ranges.

  For example, if you specify a range of valid labels for an LSR that does not overlap the range of its adjacent LDP peer, the routers will try eight times to create an LDP session between themselves before the mplsLdpFailedInitSessionThresholdExceeded notification is generated and sent to the NMS as an informational message.

  Operationally, the LSRs whose label ranges do not overlap continue their attempt to create an LDP session between themselves after the eight retry threshold is exceeded. In such cases, the LDP threshold exceeded notification alerts the network administrator to the existence of a condition in the network that may warrant attention.

  RFC 3036, *LDP Specification*, details the incompatibilities that can exist between Cisco routers and/or other vendor LSRs in an MPLS network. Among such incompatibilities, for example, are the following:

  - Nonoverlapping ATM VPI/VCI ranges (as noted above) or nonoverlapping Frame-Relay DLCI ranges between LSRs attempting to set up an LDP session
  - Unsupported label distribution method
  - Dissimilar protocol data unit (PDU) size
  - Dissimilar LDP feature support

MIB Tables in the MPLS LDP MIB Version 8 Upgrade

The MPLS LDP MIB consists of the following tables:

- mplsLdpEntityTable (Table 1)—Contains entries for every active LDP hello adjacency. Nonactive hello adjacencies appear in the mplsLdpHelloAdjacencyTable, rather than this table. This table is indexed by the local LDP identifier for the interface and the IP address of the peer active hello adjacency. (See Figure 1.)

  The advantage of showing the active hello adjacency in this table instead of sessions is that the active hello adjacency can exist even if an LDP session is not active (cannot be established). Previous implementations of the IETF MPLS-LDP MIB used sessions as the entries in this table. This approach was inadequate because as sessions went down, the entries in the entity table would disappear completely because the agent code could no longer access them. This resulted in the MIB failing to provide information about failed LDP sessions.

  Directed adjacencies are also shown in this table. These entries, however, are always administratively (adminStatus) and operationally (operStatus) up because the adjacencies disappear if the directed session were to fail. Nondirected adjacencies may disappear from the MIB on some occasions as well because adjacencies are deleted if the underlying interface becomes operationally down, for example.

- mplsLdpEntityConfGenLRTable (Table 2)—Contains entries for every LDP-enabled interface that is in the global label space. (For Cisco, this applies to all interfaces except LC-ATM.) LC-ATM entities are shown in the mplsLdpEntityConfAtmLRTable instead. Indexing is the same as the mplsLdpEntityTable plus the additional indexes mplsLdpEntityConfGenLRMin and mplsLdpEntityConfGenLRMax. These additional indexes allow more than one label range to be defined. However, in current Cisco IOS implementation, only one global label range is allowed.

- mplsLdpEntityAtmParmsTable (Table 3)—Contains entries for every LDP-enabled LC-ATM interface. This table is indexed the same as the mplsLdpEntityTable although only LC-ATM interfaces are shown.

- mplsLdpEntityConfAtmLRTable (Table 4)—Contains entries for every LDP-enabled LC-ATM interface. Indexing is the same as the mplsLdpEntityTable plus the additional indexes mplsLdpEntityConfAtmLRMinVpi and mplsLdpEntityConfAtmLRMinVci. These additional indexes allow more than one label range to be defined. However, in current Cisco IOS implementation only one label range per LC-ATM interface is allowed.

- mplsLdpEntityStatsTable (Table 5)—Augments the mplsLdpEntityTable and shares the exact same indexing for performing **GET** and **GETNEXT** operations. This table shows additional statistics for entities.

- mplsLdpPeerTable (Table 6)—Contains entries for every peer session that exists. This table is indexed by the local LDP identifier of the session, the IP address of the peer active hello adjacency, and the peer's LDP identifier. (See Figure 4.)

- mplsLdpHelloAdjacencyTable (Table 7)—Contains entries for all of the hello adjacencies. This table is indexed by the local LDP identifier of the associated session, the IP address of the peer active hello adjacency, the LDP identifier for the peer, and an arbitrary index that is set to the list position of the adjacency. (See Figure 6.)
- mplsLdpSessionTable (Table 8)—Augments the mplsLdpPeerTable and shares the exact same indexing for performing **GET** and **GETNEXT** operations. This table shows all sessions that exist.
- mplsLdpAtmSesTable (Table 9)—Contains entries for every LC-ATM session. Indexing is the same as the mplsLdpPeerTable plus the additional indexes mplsLdpSesAtmLRLowerBoundVpi and mplsLdpSesAtmLRLowerBoundVci. These additional indexes allow more than one label range to be defined. However, in current Cisco IOS implementation, only one label range per LC-ATM interface is allowed.
- mplsLdpSesStatsTable (Table 10)—Augments the mplsLdpPeerTable and shares the exact same indexing for performing **GET** and **GETNEXT** operations. This table shows additional statistics for sessions.

mplsLdpEntityTable

Table 1 lists the mplsLdpEntityTable objects and their descriptions.

Table 1 mplsLdpEntityTable Objects and Descriptions

Objects	Description
mplsLdpEntityEntry	Represents an LDP entity, which is a potential session between two peers.
mplsLdpEntityLdpId	The LDP identifier (not accessible) consists of the local LSR ID (four octets) and the label space ID (two octets).
mplsLdpEntityIndex	A secondary index to identify this row uniquely. It consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the LSR (not accessible).
mplsLdpEntityProtocolVersion	The version number of the LDP protocol to be used in the session initialization message.
mplsLdpEntityAdminStatus	The administrative status of this LDP entity is always up. If the hello adjacency fails, then this entity disappears from the mplsLdpEntityTable.
mplsLdpEntityOperStatus	The operational status of this LDP entity. Values are unknown(0), enabled(1), and disabled(2).
mplsLdpEntityTcpDscPort	The TCP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityUdpDscPort	The UDP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityMaxPduLength	The maximum PDU length that is sent in the common session parameters of an initialization message.
mplsLdpEntityKeepAliveHoldTimer	The two-octet value that is the proposed keepalive hold timer for this LDP entity.

Table 1 mplsLdpEntityTable Objects and Descriptions (continued)

Objects	Description
mplsLdpEntityHelloHoldTimer	The two-octet value that is the proposed hello hold timer for this LDP entity.
mplsLdpEntityInitSesThreshold	The threshold for notification when this entity and its peer are engaged in an endless sequence of initialization messages. The default value is 8 and cannot be changed by SNMP or CLI.
mplsLdpEntityLabelDistMethod	The specified method of label distribution for any given LDP session. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpEntityLabelRetentionMode	Can be configured to use either conservative(1) for LC-ATM or liberal(2) for all other interfaces.
mplsLdpEntityPVLMisTrapEnable	Indicates whether the mplsLdpPVLMismatch trap should be generated. If the value is enabled(1), the trap is generated. If the value is disabled(2), the trap is not generated. The default is disabled(2). **Note** The mplsLdpPVLMismatch trap is generated only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).
mplsLdpEntityPVL	If the value of this object is 0, loop detection for path vectors is disabled. Otherwise, if this object has a value greater than zero, loop detection for path vectors is enabled, and the path vector limit is this value. **Note** The mplsLdpEntityPVL object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).
mplsLdpEntityHopCountLimit	If the value of this object is 0, loop detection using hop counters is disabled. If the value of this object is greater than 0, loop detection using hop counters is enabled, and this object specifies this entity's maximum allowable value for the hop count. **Note** The mplsLdpEntityHopCountLimit object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).
mplsLdpEntityTargPeer	If this LDP entity uses a targeted adjacency, then this object is set to true(1). The default value is false(2).
mplsLdpEntityTargPeerAddrType	The type of the internetwork layer address used for the extended discovery. This object indicates how the value of mplsLdpEntityTargPeerAddr is to be interpreted.
mplsLdpEntityTargPeerAddr	The value of the internetwork layer address used for the targeted adjacency.

Table 1 mplsLdpEntityTable Objects and Descriptions (continued)

Objects	Description
mplsLdpEntityOptionalParameters	Specifies the optional parameters for the LDP initialization message. If the value is generic(1), then no optional parameters will be sent in the LDP initialization message associated with this entity. LC-ATM uses atmParameters(2) to specify that a row in the mplsLdpEntityAtmParmsTable corresponds to this entry. **Note** Frame Relay parameters are not supported in this release.
mplsLdpEntityDiscontinuityTime	The value of sysUpTime on the most recent occasion at which any one or more of this entity's counters suffered a discontinuity. The relevant counters are the specific instances associated with this entity of any Counter32, or Counter64 object contained in the mplsLdpEntityStatsTable. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, then this object contains a 0 value.
mplsLdpEntityStorType	The storage type for this entry in this release is a read-only implementation that is always volatile.
mplsLdpEntityRowStatus	This object is a read-only implementation that is always active in this release.

mplsLdpEntityConfGenLRTable

Table 2 lists the mplsLdpEntityConfGenLRTable objects and their descriptions.

Table 2 mplsLdpEntityConfGenLRTable Objects and Descriptions

Objects	Description
mplsLdpEntityConfGenLREntry	A row in the LDP Entity Configurable Generic Label Range Table. One entry in this table contains information on a single range of labels represented by the configured upper and lower bounds pairs. The current implementation supports one label range per entity.
mplsLdpEntityConfGenLRMin	The minimum label configured for this range (not accessible).
mplsLdpEntityConfGenLRMax	The maximum label configured for this range (not accessible).
mplsLdpEntityConfGenIfIndxOrZero	This value represents the SNMP IF-MIB index for the platform-wide entity. If the active hello adjacency is targeted, then the value is 0.

Table 2 mplsLdpEntityConfGenLRTable Objects and Descriptions (continued)

Objects	Description
mplsLdpEntityConfGenLRStorType	The storage type for this entry in this release is a read-only implementation that is always volatile.
mplsLdpEntityConfGenLRRowStatus	This object is a read-only implementation that is always active in this release.

mplsLdpEntityAtmParmsTable

Table 3 lists the mplsLdpEntityAtmParmsTable objects and their descriptions.

Table 3 mplsLdpEntityAtmParmsTable Objects and Descriptions

Objects	Description
mplsLdpEntityAtmParmsEntry	Represents the ATM parameters and ATM information for this LDP entity.
mplsLdpEntityAtmIfIndxOrZero	This value represents the SNMP IF-MIB index for the interface-specific LC-ATM entity.
mplsLdpEntityAtmMergeCap	Denotes the merge capability of this entity.
mplsLdpEntityAtmLRComponents	Number of label range components in the initialization message. This also represents the number of entries in the mplsLdpEntityConfAtmLRTable that correspond to this entry. **Note** This release supports only one component.
mplsLdpEntityAtmVcDirectionality	If the value of this object is bidirectional(0), a given VCI within a given VPI, is used as a label for both directions independently. If the value of this object is unidirectional(1), a given VCI within a VPI designates one direction.
mplsLdpEntityAtmLsrConnectivity	The peer LSR can be connected indirectly by means of an ATM VP so that the VPI values may be different on either endpoint so the label must be encoded entirely within the VCI field. Values are direct(1), the default, and indirect(2).
mplsLdpEntityDefaultControlVpi	The default VPI value for the non-MPLS connection.
mplsLdpEntityDefaultControlVci	The default VCI value for the non-MPLS connection.
mplsLdpEntityUnlabTrafVpi	VPI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.
mplsLdpEntityUnlabTrafVci	VCI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.

Table 3 mplsLdpEntityAtmParmsTable Objects and Descriptions (continued)

Objects	Description
mplsLdpEntityAtmStorType	The storage type for this entry in this release is a read-only implementation that is always volatile.
mplsLdpEntityAtmRowStatus	This object is a read-only implementation that is always active in this release.

mplsLdpEntityConfAtmLRTable

Table 4 lists the mplsLdpEntityConfAtmLRTable objects and their descriptions.

Table 4 mplsLdpEntityConfAtmLRTable Objects and Descriptions

Objects	Description
mplsLdpEntityConfAtmLREntry	A row in the LDP Entity Configurable ATM Label Range Table. One entry in this table contains information on a single range of labels represented by the configured upper and lower bounds VPI/VCI pairs. These are the same data used in the initialization message. This label range should overlap the label range of the peer.
mplsLdpEntityConfAtmLRMinVpi	The minimum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMinVci	The minimum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVpi	The maximum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVci	The maximum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRStorType	The storage type for this entry in this release is a read-only implementation that is always volatile.
mplsLdpEntityConfAtmLRRowStatus	This object is a read-only implementation that is always active in this release.

mplsLdpEntityStatsTable

Table 5 lists the mplsLdpEntityStatsTable objects and their descriptions.

Table 5 mplsLdpEntityStatsTable Objects and Descriptions

Objects	Description
mplsLdpEntityStatsEntry	These entries augment the mplsLdpEntityTable by providing additional information for each entry.
mplsLdpAttemptedSessions	Not supported in this release.
mplsLdpSesRejectedNoHelloErrors	A count of the session rejected/no hello error notification messages sent or received by this LDP entity.

Table 5 mplsLdpEntityStatsTable Objects and Descriptions (continued)

Objects	Description
mplsLdpSesRejectedAdErrors	A count of the session rejected/parameters advertisement mode error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedMaxPduErrors	A count of the session rejected/parameters max PDU length error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedLRErrors	A count of the session rejected/parameters label range notification messages sent or received by this LDP entity.
mplsLdpBadLdpIdentifierErrors	This object counts the number of bad LDP identifier fatal errors detected by the session associated with this LDP entity.
mplsLdpBadPduLengthErrors	This object counts the number of bad PDU length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadMessageLengthErrors	This object counts the number of bad message length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadTlvLengthErrors	This object counts the number of bad Type-Length-Value (TLV) length fatal errors detected by the session associated with this LDP entity.
mplsLdpMalformedTlvValueErrors	This object counts the number of malformed TLV value fatal errors detected by the session associated with this LDP entity.
mplsLdpKeepAliveTimerExpErrors	This object counts the number of session keepalive timer expired errors detected by the session associated with this LDP entity.
mplsLdpShutdownNotifReceived	This object counts the number of shutdown notifications received related to the session associated with this LDP entity.
mplsLdpShutdownNotifSent	This object counts the number of shutdown notifications sent related to the session associated with this LDP entity.

mplsLdpPeerTable

Table 6 lists the mplsLdpPeerTable objects and their descriptions.

Table 6 mplsLdpPeerTable Objects and Descriptions

Objects	Description
mplsLdpPeerEntry	Information about a single peer that is related to a session (not accessible). **Note** This table is augmented by the mplsLdpSessionTable.
mplsLdpPeerLdpId	The LDP identifier of this LDP peer (not accessible) consists of the peer LSR ID (four octets) and the peer label space ID (two octets).

Table 6 mplsLdpPeerTable Objects and Descriptions (continued)

Objects	Description
mplsLdpPeerLabelDistMethod	For any given LDP session, the method of label distribution. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpPeerLoopDetectionForPV	An indication of whether loop detection based on path vectors is disabled or enabled for this peer. For downstream unsolicited distribution (mplsLdpPeerLabelDistMethod is downstreamUnsolicited(2)), this object always has a value of disabled(0) and loop detection is disabled. For downstream-on-demand distribution (mplsLdpPeerLabelDistMethod is downstreamOnDemand(1)), this object has a value of enabled(1), provided that loop detection based on path vectors is enabled.
mplsLdpPeerPVL	If the value of mplsLdpPeerLoopDetectionForPV for this entry is enabled(1), this object represents that path vector limit for this peer. If the value of mplsLdpPeerLoopDetectionForPV for this entry is disabled(0), this value should be 0.

mplsLdpHelloAdjacencyTable

Table 7 lists the mplsLdpHelloAdjacencyTable objects and their descriptions.

Table 7 mplsLdpHelloAdjacencyTable Objects and Descriptions

Objects	Description
mplsLdpHelloAdjacencyEntry	Each row represents a single LDP hello adjacency. An LDP session can have one or more hello adjacencies (not accessible).
mplsLdpHelloAdjIndex	An identifier for this specific adjacency (not accessible). The active hello adjacency has mplsLdpHelloAdjIndex equal to 1.
mplsLdpHelloAdjHoldTimeRem	The time remaining for this hello adjacency. This interval changes when the next hello message, which corresponds to this hello adjacency, is received.
mplsLdpHelloAdjType	This adjacency is the result of a link hello if the value of this object is link(1). Otherwise, this adjacency is a result of a targeted hello and its value is targeted(2).

mplsLdpSessionTable

Table 8 lists the mplsLdpSessionTable objects and their descriptions.

Table 8 mplsLdpSessionTable Objects and Descriptions

Objects	Description
mplsLdpSessionEntry	An entry in this table represents information on a single session between an LDP entity and LDP peer. The information contained in a row is read-only. This table augments the mplsLdpPeerTable.
mplsLdpSesState	The current state of the session, all of the states are based on the LDP or TDP state machine for session negotiation behavior. The states are as follows: • nonexistent(1) • initialized(2) • openrec(3) • opensent(4) • operational(5)
mplsLdpSesProtocolVersion	The version of the LDP protocol which this session is using. This is the version of the LDP protocol that has been negotiated during session initialization.
mplsLdpSesKeepAliveHoldTimeRem	The keepalive hold time remaining for this session.
mplsLdpSesMaxPduLen	The value of maximum allowable length for LDP PDUs for this session. This value may have been negotiated during the session initialization.
mplsLdpSesDiscontinuityTime	The value of sysUpTime on the most recent occasion at which any one or more of this session's counters suffered a discontinuity. The relevant counters are the specific instances associated with this session of any Counter32 or Counter64 object contained in the mplsLdpSesStatsTable. The initial value of this object is the value of sysUpTime when the entry was created in this table.

mplsLdpAtmSesTable

Table 9 lists the mplsLdpAtmSesTable objects and their descriptions.

Table 9 mplsLdpAtmSesTable Objects and Descriptions

Objects	Description
mplsLdpAtmSesEntry	An entry in this table represents information on a single label range intersection between an LDP entity and an LDP peer (not accessible).
mplsLdpSesAtmLRLowerBoundVpi	The minimum VPI number for this range (not accessible).
mplsLdpSesAtmLRLowerBoundVci	The minimum VCI number for this range (not accessible).

Table 9 mplsLdpAtmSesTable Objects and Descriptions (continued)

Objects	Description
mplsLdpSesAtmLRUpperBoundVpi	The maximum VPI number for this range (read-only).
mplsLdpSesAtmLRUpperBoundVci	The maximum VCI number for this range (read-only).

mplsLdpSesStatsTable

Table 10 lists the mplsLdpSesStatsTable objects and their descriptions.

Table 10 mplsLdpSesStatsTable Objects and Descriptions

Objects	Description
mplsLdpSesStatsEntry	An entry in this table represents statistical information on a single session between an LDP entity and LDP peer. This table augments the mplsLdpPeerTable.
mplsLdpSesStatsUnkMesTypeErrors	This object counts the number of unknown message type errors detected during this session.
mplsLdpSesStatsUnkTlvErrors	This object counts the number of unknown TLV errors detected during this session.

VPN Contexts in the MPLS LDP MIB Version 8 Upgrade

Within an MPLS Border Gateway Protocol (BGP) 4 Virtual Private Network (VPN) environment, separate LDP processes can be created for each VPN. These processes and their associated data are called VPN contexts. Each context is independent from all others and contains data specific only to that context. The IETF MPLS-LDP MIB is capable only of showing information about a single context at one time.

Note This release supports a global VPN context only.

How to Configure the MPLS LDP MIB Version 8 Upgrade

This section contains the following procedures:

- Enabling the SNMP Agent, page 22 (required)
- Enabling Cisco Express Forwarding (CEF), page 23 (required)
- Enabling MPLS Globally, page 23 (required)
- Enabling LDP Globally, page 24 (required)
- Enabling MPLS on an Interface, page 25 (required)
- Enabling LDP on an Interface, page 26 (required)
- Verifying the MPLS LDP MIB Version 8 Upgrade, page 27 (optional)

Enabling the SNMP Agent

Perform this task to enable the SNMP agent.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*]
5. **end**
6. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router to determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) community strings for the MPLS LDP MIB. • The *string* argument functions like a password permitting access to SNMP functionality on label switched routers (LSRs) in an MPLS network. • The optional **ro** keyword configures read-only (ro) access to the objects in the MPLS LDP MIB.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

Enabling Cisco Express Forwarding (CEF)

Perform this task to enable CEF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed CEF (dCEF).
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling MPLS Globally

Perform this task to enable MPLS globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for the platform.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling LDP Globally

Perform this task to enable LDP globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol** {**ldp** | **tdp**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol** {**ldp** | **tdp**} **Example:** `Router(config)# mpls label protocol ldp`	Specifies the platform default label distribution protocol.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling MPLS on an Interface

Perform this task to enable MPLS on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** [*type number*]
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** [*type number*] **Example:** `Router(config)# interface Ethernet1`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.
Step 4	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for a particular interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Enabling LDP on an Interface

Perform this task to enable LDP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** [*type number*]
4. **mpls label protocol** {**ldp** | **tdp** | **both**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** [*type number*] **Example:** `Router(config-if)# interface Ethernet1`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.

	Command or Action	Purpose
Step 4	**mpls label protocol** {**ldp** | **tdp** | **both**} **Example:** `Router(config-if)# mpls label protocol ldp`	Specifies the label distribution protocol to be used on a given interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Verifying the MPLS LDP MIB Version 8 Upgrade

Perform a MIB walk using your SNMP management tool to verify that the MPLS LDP MIB Version 8 Upgrade feature is functioning.

Note This release supports SNMP v1 and v2c.

Configuration Examples for the MPLS LDP MIB Version 8 Upgrade

This section provides the following configuration examples:

- MPLS LDP MIB Version 8 Upgrade Example, page 27

MPLS LDP MIB Version 8 Upgrade Example

The following example shows how to enable an SNMP agent on the host NMS:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# snmp-server community
```

The following example shows how to enable SNMPv1 and SNMPv2C on the host NMS. The configuration permits any SNMP agent to access all MPLS LDP MIB objects with read-only permission using the community string public.

```
Router(config)# snmp-server community public
```

The following example shows how to allow read-only access to all MPLS LDP MIB objects relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any of the MPLS LDP MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to enable LDP globally and then on an interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mpls label protocol ldp
Router(config)# interface Ethernet1
Router(config-if)# mpls label protocol ldp
Router(config-if)# end
```

Additional References

The following sections provide references related to the MPLS LDP MIB Version 8 Upgrade feature:

Related Documents

Related Topic	Document Title
MPLS LDP configuration tasks	*MPLS Label Distribution Protocol (LDP)*
A description of SNMP agent support in Cisco IOS for the MPLS Label Switching Router MIB (MPLS-LSR-MIB)	*MPLS Label Switching Router MIB*
A description of SNMP agent support in Cisco IOS for the MPLS Traffic Engineering MIB (MPLS TE MIB)	*MPLS Traffic Engineering (TE) MIB*
Configuration tasks for MPLS ATM network enhancements	*MPLS Scalability Enhancements for the ATM LSR*
MPLS automatic bandwidth adjustment configuration tasks	*MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for TE Tunnels*
A description of MPLS differentiated types of service across an MPLS network	*MPLS Class of Service*

Standards

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

MIBs

MIBs	MIBs Link
• *MPLS Label Distribution Protocol MIB (draft-ietf-mpls-ldp-mib-08.txt)*	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2233 The LDP implementation supporting the MPLS LDP MIB fully complies with the provisions of Section 10 of RFC 2026, which, in effect, states that the implementation of LDP is recommended for network devices that perform MPLS forwarding along normally routed paths, as determined by destination-based routing protocols.	*Interfaces MIB*

Technical Assistance

Description	Link
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.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- **snmp-server community**
- **snmp-server enable traps**
- **snmp-server host**

snmp-server community

To configure read-only Simple Network Management Protocol (SNMP) community strings for the MPLS LDP MIB, use the **snmp-server community** command in global configuration mode on the host network management station (NMS). To change the community string to its default value, use the **no** form of this command.

snmp-server community *string* [**view** *view-name*] [**ro**] [*number*]

no snmp-server community *string*

Syntax Description

string	Consists of 1 to 32 alphanumeric characters and functions much like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. Blank spaces are not permitted in the community string.
view *view-name*	(Optional) The name of a previously defined view denoting the objects available to the SNMP community.
ro	(Optional) Configures read-only (RO) access to the objects in the MPLS LDP MIB, thus limiting NMS functions to the retrieval of objects from the MIB.
number	(Optional) An integer from 1 to 99, specifying an access list of IP addresses for LSRs that are allowed to use the community string to gain access to the SNMPv1 agent in an MPLS network.

Defaults

The default value of the read/write keyword is read-only (***ro***).

The default value of the read-only community string is *public*.

The default value of the read/write community string is *private*.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(17)ST	This command was integrated into Cisco IOS Release 12.0(17)ST.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(24)S	This command was integrated into Cisco IOS Release 12.0(24)S.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.

Usage Guidelines

The **no snmp-server** command disables both SNMPv1 and SNMPv2.

The first **snmp-server** command issued enables both SNMPv1 and SNMPv2.

Examples

The following example shows how to set the read/write community string to newstring:

```
Router(config)# snmp-server community newstring rw
```

The following example shows how to assign the string comaccess to SNMPv1, allowing read-only access and specifying that IP access list 4 can use the community string:

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to assign the string mgr to SNMPv1, allowing read/write access to the objects in the restricted view:

```
Router(config)# snmp-server community mgr view restricted rw
```

The following example shows how to remove the community string comaccess.

```
Router(config)# no snmp-server community comaccess
```

The following example shows how to disable both SNMP versions:

```
Router(config)# no snmp-server
```

Related Commands

Command	Description
snmp-server enable traps	Enables an MPLS LSR to send SNMP notification messages to a designated NMS workstation in an MPLS network.
snmp-server host	Specifies the targeted recipient of an SNMP notification message.

snmp-server enable traps

To enable a label switch router (LSR) to send Simple Network Management Protocol (SNMP) notifications or informs to an SNMP host, use the **snmp-server enable traps** command in global configuration mode. To disable notifications or informs, use the **no** form of this command.

snmp-server enable traps [*notification-type*] [*notification-option*]

no snmp-server enable traps [*notification-type*] [*notification-option*]

Syntax Description

notification-type	(Optional) Specifies the particular type of SNMP notification(s) to be enabled on the LSR. If a notification type is not specified, all SNMP notifications applicable to the LSR are enabled and sent to the SNMP host. Any one or all of the following keywords can be specified in any combination as the *notification-type* (family name) in the **snmp-server enable traps** command: • **bgp**—Sends Border Gateway Protocol (BGP) state change notifications. • **config**—Sends configuration notifications. • **entity**—Sends Entity MIB modification notifications. • **envmon**—Sends Cisco enterprise-specific environmental monitor notifications whenever certain environmental thresholds are exceeded. *Notification-option* arguments (below) can be specified in combination with this keyword. • **frame-relay**—Sends Frame Relay notifications. • **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications. • **isdn**—Sends ISDN notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword. • **repeater**—Sends Ethernet hub repeater notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword. • **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications. • **rtr**—Sends Service Assurance Agent/Response Time Reporter (RTR) notifications. • **snmp** [**authentication**]—Sends RFC 1157 SNMP notifications. Using the **authentication** keyword produces the same effect as not using it. Both the **snmp-server enable traps snmp** and the **snmp-server enable traps snmp authentication** forms of this command globally enable the following SNMP notifications (or, if using the **no** form of the command, disables such notifications): **authenticationFailure**, **linkUp**, **linkDown**, or **warmstart**. • **syslog**—Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command. • **mpls ldp**—Sends notifications about status changes in LDP sessions. Note that this keyword is specified as **mpls ldp**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword. • **mpls traffic-eng**—Sends notifications about status changes in MPLS label distribution tunnels. Note that this keyword is specified as **mpls traffic-eng**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword.

notification-option	(Optional) Defines the particular options associated with the specified *notification-type* that are to be enabled on the LSR. • **envmon** [**voltage** | **shutdown** | **supply** | **fan** | **temperature**] When you specify the **envmon** keyword, you can enable any one or all of the following environmental notifications in any combination: **voltage**, **shutdown**, **supply**, **fan**, or **temperature**. If you do not specify an argument with the **envmon** keyword, all types of system environmental notifications are enabled on the LSR. • **isdn** [**call-information** | **isdn u-interface**] When you specify the **isdn** keyword, you can use either the **call-information** argument (to enable an SNMP ISDN call information option for the ISDN MIB subsystem) or the **isdn u-interface** argument (to enable an SNMP ISDN U interface option for the ISDN U Interfaces MIB subsystem), or both. If you do not specify an argument with the **isdn** keyword, both types of isdn notifications are enabled on the LSR. • **repeater** [**health** | **reset**] When you specify the **repeater** keyword, you can use either the **health** argument (to enable the IETF Repeater Hub MIB (RFC 1516 notification) or the **reset** argument (to enable the IETF Repeater Hub MIB (RFC 1516 notification), or both. If you do not specify an argument with the **repeater** keyword, both types of notifications are enabled on the LSR. • **mpls ldp** [**session-up** | **session-down** | **pv-limit** | **threshold**] When you specify the **mpls ldp** keyword, you can use any one or all of the following arguments in any combination to indicate status changes in LDP sessions: **session-up**, **session-down**, **pv-limit**, or **threshold**. If you do not specify an argument with the **mpls ldp** keyword, all four types of LDP session notifications are enabled on the LSR. • **mpls traffic-eng** [**up** | **down** | **reroute**] When you specify the **mpls traffic-eng** keyword, you can use any one or all of the following arguments in any combination to enable the sending of notifications regarding status changes in MPLS label distribution tunnels: **up**, **down**, or **reroute**. If you do not specify an argument with the **mpls traffic-eng** keyword, all three types tunnel notifications are enabled on the LSR.

Defaults

If you issue this command on an LSR without specifying any *notification-type* keywords, the default behavior of the LSR is to enable all notification types controlled by the command (some notification types cannot be controlled by means of this command).

Command Modes

Global configuration

Command History

Release	Modification
11.1	This command was introduced.
11.3	The **snmp-server enable traps snmp authentication** form of this command was introduced to replace the **snmp-server trap-authentication** command.
12.0(17)ST	The **mpls traffic-eng** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
12.0(21)ST	The **mpls ldp** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(24)S	This command was integrated into Cisco IOS Release 12.0(24)S.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.

Usage Guidelines

To configure an LSR to send SNMP LDP notifications, you must issue at least one **snmp-server enable traps** command on the router.

To configure an LSR to send either notifications (traps) or informs to a designated NMS, you must issue the **snmp-server host** command on that device using the desired keyword (**traps** or **informs**) that suits your purposes.

If you issue the **snmp-server enable traps** command without keywords, all SNMP notification types are enabled on the LSR. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled on the LSR.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. You use the latter command to specify the NMS host (or hosts) targeted as the recipient(s) of the SNMP notifications generated by SNMP-enabled LSRs in the network. To enable an LSR to send such notifications, you must issue at least one **snmp-server host** command on the LSR.

Examples

In the following example, the router is enabled to send all notifications to the host specified as myhost.cisco.com, using the community string defined as public:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, the router is enabled to send Frame Relay and environmental monitor notifications to the host specified as myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps frame-relay

Router(config)# snmp-server enable traps envmon temperature

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, notifications are not sent to any host. BGP notifications are enabled for all hosts, but the only notifications enabled to be sent to a host are ISDN notifications (which are not enabled in this example).

```
Router(config)# snmp-server enable traps bgp

Router(config)# snmp-server host bob public isdn
```

In the following example, the router is enabled to send all inform requests to the host specified as myhost.cisco.com, using the community string defined as public:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, HSRP MIB notifications are sent to the host specified as myhost.cisco.com using the community string public*:*

```
Router(config)# snmp-server enable hsrp

Router(config)# snmp-server host myhost.cisco.com traps version 2c public hsrp
```

Related Commands

Command	Description
snmp-server host	Specifies the intended recipient of an SNMP notification (that is, the designated NMS workstation in the network).

snmp-server host

To specify a network management station (NMS) in the network as the intended recipient of Simple Network Management Protocol (SNMP) notifications or informs, use the **snmp-server host** command in global configuration mode. To disable the configuration of the NMS as an SNMP host, use the **no** form of this command.

snmp-server host *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

no snmp-server host *host-addr* [**traps** | **informs**]

Syntax Description

host-addr	Specifies the name or the IP address of the host NMS workstation on which the SNMP agent is running (thus serving as the recipient of SNMP notifications or informs).
traps	(Optional) Sends SNMP notifications to the specified NMS host. This is the default assumption of the **snmp-server host** command.
informs	(Optional) Sends SNMP informs to the specified NMS host.
version	(Optional) Indicates the SNMP version to be used in sending LDP notifications or informs to the NMS host. Version 3 is most secure, because it allows packet encryption by means of the **priv** keyword (below). If you use the **version** keyword, you must also specify one of the following arguments: • **1**—SNMPv1. This option is not available when using the **informs** keyword. • **2c**—SNMPv2C. • **3**—SNMPv3. The following optional arguments can be used with the **version 3** keyword: – **auth**—Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication. – **noauth**—Indicates the noAuthNoPriv security level. This argument is assumed as the default if the [**auth** | **noauth** | **priv**] keyword argument is not specified. – **priv**—Enables Data Encryption Standard (DES) packet encryption (also called privacy encryption).
community-string	The community string, functioning much like a password, is sent with the notification or informs operation. Although you can set this string using the **snmp-server host** command by itself, we recommend that you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.
udp-port *port*	User Datagram Protocol (UDP) port number of the NMS host to which SNMP notifications or informs are to be sent. The default UDP port number is 162.

notification-type	(Optional) Specifies the particular type of SNMP notifications or informs to be sent to the NMS host. If no notification type is specified, all applicable SNMP notifications or informs are sent. Any one or more of the following can be specified as a keyword in this command: • **bgp**—Sends Border Gateway Protocol (BGP) state change notifications. • **config**—Sends configuration notifications. • **dspu**—Sends downstream physical unit (DSPU) notifications. • **entity**—Sends entity MIB modification notifications. • **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when a specified system environmental threshold is exceeded. • **frame-relay**—Sends Frame Relay notifications. • **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications. • **isdn**—Sends ISDN notifications. • **llc2**—Sends Logical Link Control, Type 2 (LLC2) notifications. • **mpls-ldp**—Sends notifications indicating status changes in LDP sessions. This keyword (embodying a dash) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as **mpls ldp** (embodying a space), which is seen as two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command for consistency with other MPLS commands. • **mpls-traffic-eng**—Sends notifications indicating status changes in label distribution tunnels. This keyword (embodying dashes) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as **mpls traffic-eng** (with an embedded space and dash), which is seen as two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command for consistency with other MPLS commands. • **repeater**—Sends standard repeater (hub) notifications. • **rsrb**—Sends remote source-route bridging (RSRB) notifications. • **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications. • **rtr**—Sends SA Agent (RTR) notifications. • **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications. • **sdllc**—Sends SDLLC notifications. • **snmp**—Sends SNMP notifications (as defined in RFC 1157). • **stun**—Sends serial tunnel (STUN) notifications. • **syslog**—Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command. • **tty**—Sends Cisco enterprise-specific notifications when a Transmission Control Protocol (TCP) connection terminates. • **x25**—Sends X.25 event notifications.

Defaults

This command is disabled by default, in which case, no SNMP notifications are sent.

If you enter this command without keywords, the default is to send all notification types to the NMS host. No informs will be sent to the host.

If no **version** keyword is specified, the default is version 1. Issuing the **no snmp-server host** command without keywords disables notifications, but not informs, to the NMS host. To disable informs, use the **no snmp-server host informs** command.

Note If the *community-string* is not defined using the **snmp-server community** command prior to issuing the **snmp-server host** command, the default form of the **snmp-server community** command is automatically inserted into the configuration. The password (*community-string*) used for this automatic configuration of the **snmp-server community** will be the same as specified in the **snmp-server host** command. This is the default behavior for Cisco IOS Release 12.0(3) and later.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(17)ST	The **mpls-traffic-eng** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable sending traffic engineering notifications reflecting status changes in label distribution tunnels.
12.0(21)ST	The **mpls-ldp** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable sending LDP notifications reflecting status changes in LDP sessions.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(24)S	This command was integrated into Cisco IOS Release 12.0(24)S.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.

Usage Guidelines

To configure a label switch router (LSR) to send SNMP notifications to an NMS, you must enter at least one **snmp-server host** command on the LSR.

If you issue the **snmp-server host** command without keywords, all SNMP notification types are enabled for the specified NMS host. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled for the NMS host.

To enable multiple NMS hosts, you must issue a separate **snmp-server host** command for each targeted NMS host. You can specify multiple notification types in the command for each NMS.

When multiple **snmp-server host** commands are issued for the same NMS host and notification type (trap or inform request), each succeeding such command issued overwrites the previous command. For example, if you issue an **snmp-server host inform** command for an NMS host, and then issue another **snmp-server host inform** command for the same NMS host, the second command overrides the first command.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. You use the **snmp-server enable** command to specify which SNMP notifications are to be sent globally. For an NMS host to receive most SNMP notifications, at lease one **snmp-server enable** command and the **snmp-server host** command for that NMS host must be enabled on the LSR.

Examples

If you want to configure a unique SNMP community string for notifications, but you want to prevent SNMP polling access with this particular string, the configuration should include an access-list. In the following example, the community string is named comaccess and the access list is numbered 10:

```
Router(config)# snmp-server community comaccess ro 10

Router(config)# snmp-server host 172.20.2.160 comaccess

Router(config)# access-list 10 deny any
```

In the following example, SNMP notifications are sent to the NMS host specified as myhost.cisco.com. The community string is defined as comaccess.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com comaccess snmp
```

In the following example, SNMP and the Cisco environmental monitor (envmon) enterprise-specific notifications are sent to the NMS host identified by IP address 172.30.2.160:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host 172.30.2.160 public snmp envmon
```

In the following example, the LSR is enabled to send all notifications to the SNMP host identified as myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, notifications will not be sent to any SNMP host. The BGP notifications are enabled for all hosts, but only the ISDN notifications are enabled for sending to the host NMS.

```
Router(config)# snmp-server enable traps bgp

Router(config)# snmp-server host bob public isdn
```

In the following example, the LSR is enabled to send all inform requests to the NMS host specified as myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, HSRP MIB notifications are sent to the NMS host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable hsrp

Router(config)# snmp-server host myhost.cisco.com traps version 2c public hsrp
```

Related Commands

Command	Description
snmp-server enable traps	Enables the LSR on which this command is executed to send SNMP notifications to a designated NMS host.

Glossary

ATM—Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

informs—A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, while a trap notification does not.

label—A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

label distribution—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

LDP—Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP—label-switched path. A configured connection between two label switch routers (LSRs) in which label-switching techniques are used for packet forwarding; a specific path through an MPLS network.

LSR—label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

MIB—Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by using SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MPLS label distribution—A constraint-based routing algorithm for routing label-switched path (LSP) tunnels.

NMS—network management station. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

notification—A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS has occurred. *See also* trap.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

SNMP—Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, manage configurations, collect statistics, monitor performance, and ensure network security.

TDP—Tag Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the tags (addresses) used to forward packets. *See also* LDP.

TLV—Type-Length-Value. A block of information embedded in Cisco Discovery Protocol (CDP) advertisements.

trap—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS has occurred. Traps (notifications) are less reliable than inform requests, because the receiver of the trap does not send an acknowledgment of receipt; furthermore, the sender of the trap cannot determine if the trap was received. *See also* notification.

VCC—virtual channel connection. A logical circuit, made up of virtual channel links (VCLs), that carries data between two endpoints in an ATM network. Sometimes called a *virtual circuit connection*.

VCI—virtual channel identifier. A 16-bit field in the header of an ATM cell. The VCI, together with the virtual path identifier (VPI), is used to identify the next network virtual channel link (VCL) as the cell passes through a series of ATM switches on its way to its final destination.

VCL—virtual channel link. The logical connection that exists between two adjacent switches in an ATM network.

VPI—virtual path identifier. An 8-bit field in the header of an ATM cell. The VPI, together with the virtual channel identifier (VCI), is used to identify the next network virtual channel link (VCL) as the cell passes through a series of ATM switches on its way to its final destination.

VPN—Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.