Network Working Group
Internet Draft
Expires: September 2000

Cheenu Srinivasan
Tachion Networks, Inc.

Arun Viswanathan
Force10 Networks

Thomas D. Nadeau
Cisco Systems, Inc.

MPLS Label Switch Router Management Information Base Using SMIv2

draft-ietf-mpls-lsr-mib-02.txt

Status of this Memo

Abstract

This memo defines an experimental portion of the Management
Information Base (MIB) for use with network management protocols
in the Internet community. In particular, it describes managed
objects for modeling a Multi-Protocol Label Switching (MPLS)
[MPLSArch, MPLSFW] Label Switch Router (LSR).

1. Introduction

This memo defines an experimental portion of the Management
Information Base (MIB) for use with network management protocols
in the Internet community. In particular, it describes managed
objects for modeling a Multi-Protocol Label Switching (MPLS)
[MPLSArch, MPLSFW] Label Switch Router (LSR).

Comments should be made directly to the MPLS mailing list at
mpls@uu.net.

This memo does not, in its draft form, specify a standard for the
Internet community.

2. Terminology

This document uses terminology from the document describing the MPLS architecture [MPLSArch]. A label switched path (LSP) is modeled as a connection consisting of one or more incoming segments (in-segments) and/or one or more outgoing segments (out-segments) at a label switch router (LSR). The association or interconnection of the in-segments and out-segments is accomplished by using a cross-connect. We use the terminology "connection" and "LSP" interchangeably where the meaning is clear from the context.

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2271 [SNMPArch].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [SMIv1], RFC 1212 [SNMPv1MIBDef] and RFC 1215 [SNMPv1Traps]. The second version, called SMIv2, is described in RFC 1902 [SMIv2], RFC 1903 [SNMPv2TC] and RFC 1904 [SNMPv2Conf].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [SNMPv1]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [SNMPv2c] and RFC 1906 [SNMPv2TM]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [SNMPv2TM], RFC 2272 [SNMPv3MP] and RFC 2574 [SNMPv3USM].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [SNMPv1]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [SNMPv2PO].

- A set of fundamental applications described in RFC 2273 [SNMPv3App] and the view-based access control mechanism described in RFC 2575 [SNMPv3VACM].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI. This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine-readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine-readable information is not considered to change the semantics of the MIB.

3.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One

(ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to also refer to the object type.

4. Feature Checklist

The MPLS label switch router MIB (LSR-MIB) is designed to satisfy the following requirements and constraints:

- The MIB should be able to support both manually configured LSPs as well as those configured via LDP and/or RSVP signaling.

- The MIB must support the enabling and disabling of MPLS capability on MPLS capable interfaces of an LSR.

- The MIB should allow resource sharing between two or more LSPs.

- Both per-platform and per-interface label spaces must be supported.

- MPLS packets must be forwarded solely based on an incoming top label [MPLSArch, LblStk].

- Support must be provided for next-hop resolution when the outgoing interface is a shared media interface. In the point-to-multipoint case, each outgoing segment can reside on a different shared media interface.

- The MIB must support point-to-point, point-to-multipoint and multipoint-to-point connections at an LSR.

- For multipoint-to-point connections all outgoing packets must have the same top label.

- For multipoint-to-point connections, the outgoing resources of the merged connections must be shared.

- For multipoint-to-point connections, packets from different incoming connections may have distinct outgoing label stacks beneath the (identical) top label.

- In the point-to-multipoint case each outgoing connection can have a distinct label stack including the top label.

- All the members of a point-to-multipoint connection share the resources allocated for the ingress segments.

- The MIB must provide cross-connect capability to "pop" an incoming label and forward the packet with the remainder of the label stack unchanged and without pushing any labels ("pop-and-go") [LblStk].

- It must be possible to assign or re-map the Class of Service (COS) bits [LblStk] on the outgoing label. In the multipoint-to-point case, each in-segment can have a different outgoing COS value. In the point-to-multipoint case, each out-segment can have a different outgoing COS value.

- It should be possible to support persistent as well as non-persistent LSPs.

- Performance counters must be provided for in-segments and out-segments as well as for measuring MPLS performance on a per-interface basis.

5. Outline

Configuring LSPs through an LSR involves the following steps:

- Enabling MPLS on MPLS capable interfaces.

- Configuring in-segments and out-segments.

- Setting up the cross-connect table to associate segments and/or to indicate connection origination and termination.

- Optionally specifying label stack actions.

- Optionally specifying segment traffic parameters.

5.1. Summary of LSR MIB

The MIB objects for performing these actions consist of the following tables:

- The interface configuration table (mplsInterfaceConfTable), which is used for enabling the MPLS protocol on MPLS-capable interfaces.

- The in-segment (mplsInSegmentTable) and out-segment (mplsOutSegmentTable) tables, which are used for configuring LSP segments at an LSR.

- The cross-connect table (mplsXCTable), which is used to associate in and out segments together, in order to form a cross-connect.

- The label stack table (mplsLabelStackTable), which is used for specifying label stack operations.

- The TSpec table (mplsTSpecTable), which is used for specifying LSP-related traffic parameters.

Further, the MPLS in-segment and out-segment performance tables, mplsInSegmentPerfTable and mplsOutSegmentPerfTable, contain the objects necessary to measure the performance of LSPs, and mplsInterfacePerfTable has objects to measure MPLS performance on a per-interface basis.

These tables are described in the subsequent sections.

6. Brief Description of MIB Objects

Sections 6.1-6.3 describe objects pertaining to MPLS capable interfaces of an LSR. The objects described in Sections 6.4-6.9, when considered together, are equivalent to the tables described in the MPLS architecture document [MPLSArch], that is, the Incoming Label Map (ILM) and the Next Hop Label Forwarding Entry (NHLFE) tables. Section 6.10 describes objects for specifying traffic parameters for in and out segments.

6.1. mplsInterfaceConfTable

This table represents the interfaces that are MPLS capable. An LSR creates an entry in this table for every MPLS capable interface on that LSR. Each entry contains information about per-interface label ranges. The administrator can specify the desired MPLS status (enable/up, disable/down, testing) of an interface by writing the object mplsInterfaceAdminStatus. The actual status is indicated by the object mplsInterfaceOperStatus.

6.2. mplsInterfaceResTable

This table provides resource information such as available and allocated bandwidth and buffers on each MPLS capable interface for each priority level.

6.3. mplsInterfacePerfTable

This table contains objects to measure the MPLS performance of MPLS capable interfaces and is an AUGMENT to mplsInterfaceConfTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

6.4. mplsInSegmentTable

This table contains a description of the incoming MPLS segments to an LSR and their associated parameters.

6.5. mplsInSegmentPerfTable

The MPLS In-Segment Performance Table has objects to measure the performance of an incoming segment configured on an LSR. It is an AUGMENT to mplsInSegmentTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

6.6. mplsOutSegmentTable

The Out-Segment Table contains a description of the outgoing MPLS segments at an LSR and their associated parameters.

6.7. mplsOutSegmentPerfTable

The MPLS Out-Segment Table contains objects to measure the performance of an outgoing segment configured on an LSR. It is an AUGMENT to mplsOutSegmentTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

6.8. mplsXCTable

The mplsXCTable specifies information for associating segments together in order to instruct the LSR to switch between the specified segments. It supports point-to-point, point-to-multi-point and multi-point-to-point connections.

6.9. mplsLabelStackTable

The mplsLabelStackTable specifies the label stack to be pushed onto a packet, beneath the top label. Entries to this table are referred to from mplsXCTable.

6.10. mplsTSpecTable

The mplsTSpecTable contains objects for specifying the traffic parameters of in-segments and out-segments. Entries in this table are referred to from mplsInSegmentTable and mplsOutSegmentTable.

7. Example of LSP Setup

In this section we provide a brief example of using the MIB objects described in Section 8 to set up an LSP. While this example is not meant to illustrate every nuance of the MIB, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB itself.

Suppose that one would like to manually create a best-effort, unidirectional LSP. Assume that the LSP enters the LSR via MPLS interface A with ifIndex 12 and exits the LSR via MPLS interface B with ifIndex 13. Let us assume that we do not wish to have a label stack beneath the top label on the outgoing labeled packets. The following example illustrates which rows and corresponding objects might be created to accomplish this.

First, the TSpec entries must be set-up for both segments.

In mplsTSpecTable for the incoming direction:

```
{
   mplsTSpecIndex           = 5
   mplsTSpecMaxRate         = 100000,
   mplsTSpecMeanRate        = 100000,
   mplsTSpecMaxBurstSize    = 2000,
   mplsTSpecRowStatus       = createAndGo(4)
}
```

In mplsTSpecTable for the outgoing direction:

```
{
   mplsTSpecIndex           = 6
   mplsTSpecMaxRate         = 100000,
   mplsTSpecMeanRate        = 100000,
   mplsTSpecMaxBurstSize    = 2000,
   mplsTSpecRowStatus       = createAndGo(4)
}
```

Note that if we were setting up a bi-directional LSP, the segments in the reverse direction can share the TSpec entries (and hence resources) with the segments in the forward direction.

We must next create the appropriate in-segment and out-segment entries with suitable traffic parameters by pointing to the appropriate TSpec entries that we have just created.

In mplsInSegmentTable:

```
{
      mplsInSegmentIfIndex     = 12, -- incoming interface
      mplsInSegmentLabel       = 21, -- incoming label
      mplsInSegmentNPop        = 1,
      mplsInSegmentTSpecIndex  = 5,
      mplsInSegmentRowStatus   = createAndGo(4)
   }
```

In mplsOutSegmentTable:

```
   {
      mplsOutSegmentIndex           = 1,
      mplsOutSegmentIfIndex         = 13, -- outgoing interface
      mplsOutSegmentPushTopLabel    = true(1),
      mplsOutSegmentTopLabel        = 22, -- outgoing label
      mplsOutSegmentTSpecIndex      = 6,
      mplsOutSegmentRowStatus       = createAndGo(4)
   }
```

Next, a cross-connect entry is created thereby associating the newly created segments together.

In mplsXCTable:

```
   {
      mplsXCIndex             = 2,
      mplsXCLspId             = "1.2.3.4-2",
      mplsInSegmentIfIndex    = 12,
      mplsInSegmentLabel      = 21,
      mplsOutSegmentIndex     = 1,
      mplsXCCOS               = 0,
      mplsXCIsPersistent      = false (1),
      mplsLabelStackIndex     = 0, -- only a single outgoing label
      mplsXCRowStatus         = createAndGo(4)
   }
```

Note that the mplsInSegmentXCIndex and mplsOutSegmentXCIndex objects will automatically be populated with the value 2 when these segments are referred to from the corresponding cross-connect entry.

8. Application of the Interface Group to MPLS

The Interfaces Group of MIB II defines generic managed objects for managing interfaces. This memo contains the media-specific extensions to the Interfaces Group for managing MPLS interfaces.

This memo assumes the interpretation of the Interfaces Group to be in accordance with [IFMIB] which states that the interfaces table (ifTable) contains information on the managed resource's interfaces and that each sub-layer below the internetwork layer of a network interface is considered an interface. Thus, the MPLS layer interface is represented as an entry in the ifTable. This entry is concerned with the MPLS layer as a whole, and not with individual LSPs/tunnels which are managed via the MPLS-specific managed objects specified in this memo and [TEMIB]. The inter-relation of entries in the ifTable is defined by Interfaces Stack Group defined in [IFMIB].

8.1. Support of the MPLS Layer by ifTable

Some specific interpretations of ifTable for the MPLS layer follow.

```
Object         Use for the MPLS layer
```

ifIndex	Each MPLS interface is represented by an ifEntry.
ifDescr	Description of the MPLS interface.
ifType	The value that is allocated for MPLS is 166.
ifSpeed	The total bandwidth in bits per second for use by the MPLS layer.
ifPhysAddress	Unused.
ifAdminStatus	See [IFMIB].
ifOperStatus	Assumes the value down(2) if the MPLS layer is down.
ifLastChange	See [IFMIB].
ifInOctets	The number of received octets over the interface, i.e., the number of received, octets received as labeled packets.
ifOutOctets	The number of transmitted octets over the interface, i.e., the number of octets transmitted as labeled packets.
ifInErrors	The number of labeled packets dropped due to uncorrectable errors.
ifInUnknownProtos	The number of received packets discarded during

	packet header validation, including packets with unrecognized label values.
ifOutErrors	See [IFMIB].
ifName	Textual name (unique on this system) of the interface or an octet string of zero length.
ifLinkUpDownTrapEnable	Default is disabled (2).
ifConnectorPresent	Set to false (2).
ifHighSpeed	See [IFMIB].
ifHCInOctets	The 64-bit version of ifInOctets; supported if required by the compliance statements in [IFMIB].
ifHCOutOctets	The 64-bit version of ifOutOctets; supported if required by the compliance statements in [IFMIB].
ifAlias	The non-volatile 'alias' name for the interface as specified by a network manager.

9. MPLS Label Switch Router MIB Definitions

```
MPLS-LSR-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   experimental, Integer32, Unsigned32, Counter32,
   Counter64, Gauge32, IpAddress
      FROM SNMPv2-SMI
   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF
   TEXTUAL-CONVENTION, TruthValue, RowStatus
      FROM SNMPv2-TC
   ifIndex, InterfaceIndex, InterfaceIndexOrZero
      FROM IF-MIB;

mplsLsrMIB MODULE-IDENTITY
   LAST-UPDATED "200002161200Z"  -- 16 February 2000 12:00:00 EST
   ORGANIZATION "Multiprotocol Label Switching (MPLS) Working Group"
   CONTACT-INFO
       "        Cheenu Srinivasan
        Postal: Tachion Networks, Inc.
                2 Meridian Road
                Eatontown, NJ 0772
        Tel:    +1 732 542 7750 x234
        Email:  cheenu@tachion.com

                Arun Viswanathan
        Postal: Force10 Networks
                1440 McCarthy Blvd
                Milpitas, CA 95035
        Tel:    +1-408-571-3516
        Email:  arun@force10networks.com

                Thomas D. Nadeau
        Postal: Cisco Systems, Inc.
                250 Apollo Drive
                Chelmsford, MA 01824
        Tel:    +1-978-244-3051
        Email:  tnadeau@cisco.com"

   DESCRIPTION
       "This MIB contains managed object definitions for the
        Multiprotocol Label Switching (MPLS) Router as
        defined in: Rosen, E., Viswanathan, A., and R.
        Callon, Multiprotocol Label Switching Architecture,
        Internet Draft <draft-ietf-mpls-arch-06.txt>,
        February 2000."

   -- Revision history.
   REVISION
       "199907161200Z"  -- 16 July 1999 12:00:00 EST
   DESCRIPTION
      "Initial draft version."
   REVISION
       "200002161200Z"  -- 16 February 2000 12:00:00 EST
   DESCRIPTION
       "Second draft version."
   REVISION
       "200003061200Z"  -- 6 March 2000 12:00:00 EST
   DESCRIPTION
       "Third draft version."

   ::= { experimental 96 }


-- Textual Conventions.

MplsLSPID ::= TEXTUAL-CONVENTION
   STATUS        current
   DESCRIPTION
       "An identifier that is assigned to each LSP and is
        used to uniquely identify it. This is assigned at
        the head end of the LSP and can be used by all LSRs
        to identify this LSP. This value is piggybacked by
        the signaling protocol when this LSP is signaled
        within the network. This identifier can then be used
        at each LSR to identify which labels are being
        swapped to other labels for this LSP. For IPv4
        addresses this results in a 6-octet long cookie."
   SYNTAX        OCTET STRING (SIZE (0..63))

MplsLsrIANAAddrFamily ::= TEXTUAL-CONVENTION
   STATUS        current
   DESCRIPTION
       "An address family.  These values are defined in RFC
        1700 and are maintained by The IANA.  All values may
        not be relevant in all contexts when used in this
        MIB, but are included for completeness."
   REFERENCE
       "RFC 1700 - Assigned Numbers, Reynolds, J. and J.
        Postel, Oct. 1994"
   SYNTAX        INTEGER {
                   other(0),
                    ipv4(1),
                    ipv6(2),
                    nsap(3),
                    hdlc(4),
                    bbn1822(5),
                    ieee802(6),
                    e163(7),
                    e164(8),
                    f69(9),
                    x121(10),
                    ipx(11),
                    appleTalk(12),
                    decnetIV(13),
                    banyanVines(14),
                    e164WithNsap(15)
                }

-- An MPLS label.
MplsLabel ::= TEXTUAL-CONVENTION
   STATUS        current
   DESCRIPTION
       "Represents an MPLS label.  Note that the contents of
        a label field are interpreted in an interface-type
        specific fashion.  For example, the 20-bit wide
        label carried in the MPLS shim header is contained
        in bits 0-19 and bits 20-31 must be zero.  The frame
        relay label can be either 10, 17 or 23 bits wide
        depending on the size of the DLCI field and bits 10-
        31, 17-31 or 23-31 must be zero, respectively.  For
        an ATM interface, bits 0-15 must be interpreted as
        the VCI, bits 16-23 as the VPI and bits 24-31 must
        be zero.  Note that the permissible label values are
        also a function of the interface type.  For example,
        the value 3 has special semantics in the control
        plane for an MPLS shim header label and is not a
        valid label value in the data path."
   REFERENCE
       "1. MPLS Label Stack Encoding, Rosen et al, draft-
          ietf-mpls-label-encaps-07.txt, March 2000.
        2. Use of Label Switching on Frame Relay Networks,
          Conta et al, draft-ietf-mpls-fr-03.txt, Nov.
          1998."
   SYNTAX Integer32

Ipv6Address ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION
       "IPv6 address."
   SYNTAX      OCTET STRING (SIZE(16))

MplsBitRate ::= TEXTUAL-CONVENTION
   DISPLAY-HINT "d"
   STATUS      current
   DESCRIPTION
       "An estimate of bandwidth in units of 1,000 bits per
        second.  If this object reports a value of 'n' then
        the rate of the object is somewhere in the range of
        'n-500' to 'n+499'. For objects which do not vary in
        bitrate, or for those where no accurate estimation
        can be made, this object should contain the nominal
        bitrate."
   SYNTAX      Integer32 (1..2147483647)

MplsBurstSize ::= TEXTUAL-CONVENTION
   DISPLAY-HINT "d"
   STATUS      current
   DESCRIPTION
       "The number of octets of MPLS data that the stream
        may send back-to-back without concern for policing."
   SYNTAX      Integer32 (1..2147483647)

MplsBufferSize ::= TEXTUAL-CONVENTION
```

```
   DISPLAY-HINT "d"
   STATUS      current
   DESCRIPTION
       "Size of buffer in octets of MPLS data."
   SYNTAX      Integer32 (1..2147483647)


-- Top level components of this MIB.

-- tables, scalars
mplsLsrObjects       OBJECT IDENTIFIER ::= { mplsLsrMIB 1 }
-- traps
mplsLsrNotifications OBJECT IDENTIFIER ::= { mplsLsrMIB 2 }
-- conformance
mplsLsrConformance   OBJECT IDENTIFIER ::= { mplsLsrMIB 3 }


-- MPLS Interface Configuration Table.

mplsInterfaceConfTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsInterfaceConfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table specifies per-interface MPLS capability
        and associated information."
   ::= { mplsLsrObjects 1 }

mplsInterfaceConfEntry OBJECT-TYPE
   SYNTAX        MplsInterfaceConfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
```

```
       DESCRIPTION
       "An entry in this table is created by an LSR for
        every interface capable of supporting MPLS. When the
        global label space is in use, an entry with index 0
        is created in this table. This entry contains
        parameters that apply to all interfaces that
        participate in the global label space. Other
        interfaces defined in this table indicate whether or
        not they participate in the global label space by
        setting the mplsInterfaceIsGlobalLabelSpace variable
        to true. Note that interfaces which have specified
        that they participate in an interface-specific label
        space may also participate in the global label space
        simultaneously. In this case, the interface with
        index 0 should referenced for global label space
        parameters such as the label ranges. It may be
        useful to configure additional interfaces in this
        table for interfaces which participate in the global
        label space so that parameters such as bandwidth and
        buffer resources maybe specified individually.

        Please note that either
        mplsInterfaceIsGlobalLabelSpace or
        mplsInterfaceIsLocalLabelSpace MUST be set to true
        on every interface configured in this table."
   INDEX       { mplsInterfaceConfIndex }
      ::= { mplsInterfaceConfTable 1 }

MplsInterfaceConfEntry ::= SEQUENCE {
      mplsInterfaceConfIndex          InterfaceIndexOrZero,
      mplsInterfaceLabelMinIn         MplsLabel,
      mplsInterfaceLabelMaxIn         MplsLabel,
      mplsInterfaceLabelMinOut        MplsLabel,
      mplsInterfaceLabelMaxOut        MplsLabel,
      mplsInterfaceTotalBandwidth        MplsBitRate,
      mplsInterfaceAvailableBandwidth    MplsBitRate,
      mplsInterfaceTotalBuffer        MplsBufferSize,
      mplsInterfaceAvailableBuffer    MplsBufferSize,
      mplsInterfaceIsGlobalLabelSpace    TruthValue,
      mplsInterfaceIsLocalLabelSpace     TruthValue,
      mplsInterfaceAdminStatus        INTEGER,
      mplsInterfaceOperStatus         INTEGER
   }

mplsInterfaceConfIndex OBJECT-TYPE
   SYNTAX        InterfaceIndexOrZero
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This is a unique index for an entry in the
        MplsInterfaceConfTable. A non-zero index for an
        entry indicates the ifIndex for the corresponding
        interface entry in of the MPLS-layer in the ifTable.
        Note that the global label space may apply to
        several interfaces, and therefore the configuration
        of the global label space interface parameters will
        apply to all of the interfaces that are
        participating in the global label space."
   REFERENCE
       "RFC 2233 - The Interfaces Group MIB using SMIv2,
        McCloghrie, K., and F. Kastenholtz, Nov. 1997"
   ::= { mplsInterfaceConfEntry 1 }

mplsInterfaceLabelMinIn OBJECT-TYPE
```

```
   SYNTAX        MplsLabel
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This is the minimum value of an MPLS label that this
        LSR is willing to receive on this interface. Please
        note that in the case that the
        mplsInterfaceIsLocalLabelSpace is set to true, this
        value indicates the value appropriate for the per-
        interface label range. If the
        mplsInterfaceIsGlobalLabelSpace is true, please
        refer to the interface whose index is 0 for the
        value which applies to the global label space."
   ::= { mplsInterfaceConfEntry 2 }

mplsInterfaceLabelMaxIn OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This is the maximum value of an MPLS label that this
        LSR is willing to receive on this interface. Please
        note that in the case that the
        mplsInterfaceIsLocalLabelSpace is set to true, this
        value indicates the value appropriate for the per-
        interface label range. If the
        mplsInterfaceIsGlobalLabelSpace is true, please
        refer to the interface whose index is 0 for the
        value which applies to the global label space."
   ::= { mplsInterfaceConfEntry 3 }

mplsInterfaceLabelMinOut OBJECT-TYPE
```

```
   SYNTAX        MplsLabel
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This is the minimum value of an MPLS label that this
        LSR is willing to send on this interface. Please
        note that in the case that the
        mplsInterfaceIsLocalLabelSpace is set to true, this
        value indicates the value appropriate for the per-
        interface label range. If the
        mplsInterfaceIsGlobalLabelSpace is true, please
        refer to the interface whose index is 0 for the
        value which applies to the global label space."
   ::= { mplsInterfaceConfEntry 4 }

mplsInterfaceLabelMaxOut OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This is the maximum value of an MPLS label that this
        LSR is willing to send on this interface. Please
        note that in the case that the
        mplsInterfaceIsLocalLabelSpace is set to true, this
        value indicates the value appropriate for the per-
        interface label range. If the
        mplsInterfaceIsGlobalLabelSpace is true, please
        refer to the interface whose index is 0 for the
        value which applies to the global label space."
   ::= { mplsInterfaceConfEntry 5 }

mplsInterfaceTotalBandwidth        OBJECT-TYPE
```

```
   SYNTAX        MplsBitRate
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates the total amount of usable
        bandwidth on this interface and is specified in
        kilobits per second (Kbps/sec). This variable is not
        applicable when applied to the interface with index
        0."
::= { mplsInterfaceConfEntry 6 }

mplsInterfaceAvailableBandwidth       OBJECT-TYPE
   SYNTAX        MplsBitRate
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates the total amount of available
        bandwidth available on this interface and is
        specified in kilobits per second (Kbps/sec). This
        value is calculated as the difference between the
        amount of bandwidth currently in use and that
        specified in mplsInterfaceTotalBandwidth. This
        variable is not applicable when applied to the
        interface with index 0."
::= { mplsInterfaceConfEntry 7 }

mplsInterfaceTotalBuffer        OBJECT-TYPE
   SYNTAX        MplsBufferSize
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates the total amount of buffer
```

```
        space allocated for this interface. This variable is
        not applicable when applied to the interface with
        index 0."
::= { mplsInterfaceConfEntry 8 }

mplsInterfaceAvailableBuffer       OBJECT-TYPE
   SYNTAX        MplsBufferSize
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value reflects the total amount of buffer space
        available on this interface. This variable is not
        applicable when applied to the interface with index
        0."
::= { mplsInterfaceConfEntry 9 }

mplsInterfaceIsGlobalLabelSpace    OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates whether or not this interface
        participates in the global label space."
    DEFVAL      { false }
::= { mplsInterfaceConfEntry 10 }

mplsInterfaceIsLocalLabelSpace    OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates whether or not this interface
```

```
        uses in the local or per-interface label space."
    DEFVAL      { false }
::= { mplsInterfaceConfEntry 11 }

mplsInterfaceAdminStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),     -- enable MPLS on this interface
         down(2),   -- disable MPLS on this interface
         testing(3) -- in some test mode
      }
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "This variable indicates the administrator's intent
        as to whether MPLS should be enabled, disabled, or
        running in some diagnostic testing mode on this
        interface."
   DEFVAL        { down }
   ::= { mplsInterfaceConfEntry 12 }

mplsInterfaceOperStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),          -- ready to pass packets
         down(2),
         testing(3),     -- in some test mode
         unknown(4),     -- status cannot be determined for some
                         -- reason
         dormant(5),
         notPresent(6),  -- some component is missing
         lowerLayerDown(7)
                       -- down due to the state of
                       -- lower layer interfaces
      }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value reflects the actual or operational status
        of MPLS on this interface. The operStatus MUST NOT
        enter the up state unless either
        mplsInterfaceIsGlobalLabelSpace or
        mplsInterfaceIsLocalLabelSpace is set to true."
   ::= { mplsInterfaceConfEntry 13 }

-- End of mplsInterfaceConfTable


-- MPLS Interface Performance Table.

mplsInterfacePerfTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsInterfacePerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table provides MPLS performance information on
        a per-interface basis."
   ::= { mplsLsrObjects 2 }

mplsInterfacePerfEntry OBJECT-TYPE
   SYNTAX        MplsInterfacePerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table is created by the LSR for
        every interface capable of supporting MPLS.  Its is
        an extension to the mplsInterfaceConfEntry table."
   AUGMENTS      { mplsInterfaceConfEntry }
      ::= { mplsInterfacePerfTable 1 }

MplsInterfacePerfEntry ::= SEQUENCE {
      -- incoming direction
      mplsInterfaceInLabelsUsed           Gauge32,
      mplsInterfaceInPackets              Counter32,
      mplsInterfaceInDiscards             Counter32,
      mplsInterfaceFailedLabelLookup      Counter32,

      -- outgoing direction
      mplsInterfaceOutLabelsUsed          Gauge32,
      mplsInterfaceOutPackets             Counter32,
      mplsInterfaceOutDiscards            Counter32,
      mplsInterfaceOutFragments           Counter32
   }

mplsInterfaceInLabelsUsed OBJECT-TYPE
   SYNTAX        Gauge32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates the specific number of labels
        that are in use at this point in time on this
        interface in the incoming direction."
   ::= { mplsInterfacePerfEntry 1 }

mplsInterfaceInPackets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This variable reflects the number of labeled packets
        that have been received on this interface."
   ::= { mplsInterfacePerfEntry 2 }

mplsInterfaceInDiscards OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The number of inbound labeled packets, which were
        chosen to be discarded even though no errors had
        been detected to prevent their being transmitted.
        One possible reason for discarding such a labeled
        packet could be to free up buffer space."
   ::= { mplsInterfacePerfEntry 3 }

mplsInterfaceFailedLabelLookup OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value indicates the number of labeled packets
        that have been received on this interface and were
        discarded because there were no matching entries
        found for them in mplsInSegmentTable."
   ::= { mplsInterfacePerfEntry 4 }

mplsInterfaceOutLabelsUsed OBJECT-TYPE
   SYNTAX        Gauge32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Indicates the number of top-most labels in the
        outgoing label stacks that are in use at this point
        in time on this interface."
   ::= { mplsInterfacePerfEntry 5 }

mplsInterfaceOutPackets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This variable contains the number of labeled packets
        that have been transmitted on this interface."
   ::= { mplsInterfacePerfEntry 6 }

mplsInterfaceOutDiscards OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The number of outbound labeled packets, which were
        chosen to be discarded even though no errors had
        been detected to prevent their being transmitted.
        One possible reason for discarding such a labeled
        packet could be to free up buffer space."
   ::= { mplsInterfacePerfEntry 7 }

mplsInterfaceOutFragments OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
```

```
       "This variable indicates the number of outgoing MPLS
        packets that required fragmentation before
        transmission on this interface."
::= { mplsInterfacePerfEntry 8 }

-- End of mplsInterfacePerfTable


-- In-segment table.

mplsInSegmentTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsInSegmentEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table contains a collection of incoming
        segments to an LSR."
   ::= { mplsLsrObjects 3 }

mplsInSegmentEntry  OBJECT-TYPE
   SYNTAX        MplsInSegmentEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents one incoming
        segment.  An entry can be created by a network
        administrator or an SNMP agent, or an MPLS signaling
        protocol. The creator of the entry is denoted by
        mplsInSegmentOwner. An entry in this table is
        indexed by the ifIndex of the incoming interface and
        the (top) label.  Note that some segments are
        associated with a tunnel, so the traffic parameters
        of these rows are supported as read-only objects and
        their modification can be done only via the tunnel
        table, mplsTunnelTable."
   REFERENCE
       "MPLS Traffic Engineering Management Information Base
        Using SMIv2, Srinivasan, Viswanathan and Nadeau,
        draft-ietf-mpls-te-mib-02.txt, February 2000."
   INDEX         { mplsInSegmentIfIndex, mplsInSegmentLabel }
   ::= { mplsInSegmentTable 1 }

MplsInSegmentEntry ::= SEQUENCE {
      mplsInSegmentIfIndex         InterfaceIndex,
      mplsInSegmentLabel           MplsLabel,
      mplsInSegmentNPop            Integer32,
      mplsInSegmentAddrFamily      MplsLsrIANAAddrFamily,
      mplsInSegmentXCIndex         Integer32,
      mplsInSegmentTSpecIndex      Integer32,
      mplsInSegmentOwner           INTEGER,
      mplsInSegmentAdminStatus     INTEGER,
      mplsInSegmentOperStatus      INTEGER,
      mplsInSegmentRowStatus       RowStatus
   }

mplsInSegmentIfIndex OBJECT-TYPE
   SYNTAX        InterfaceIndexOrZero
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This is a unique index for an entry in the
        MplsInSegmentTable. This value represents the
        interface index for the incoming MPLS interface.  A
        value of zero represents an incoming label from the
        per-platform label space.  In this case, the
        mplsInSegmentLabel is interpreted to be an MPLS-type
        label."
   ::= { mplsInSegmentEntry 1 }

mplsInSegmentLabel OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The incoming label for this segment."
   ::= { mplsInSegmentEntry 2 }

mplsInSegmentNPop OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The number of labels to pop from the incoming
        packet.  Normally only the top label is popped from
        the packet and used for all switching decisions for
        that packet."
   DEFVAL        { 1 }
   ::= { mplsInSegmentEntry 3 }

mplsInSegmentAddrFamily OBJECT-TYPE
   SYNTAX        MplsLsrIANAAddrFamily
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The IANA address family of the incoming packet.  A
        value of zero indicates that the family type is
        either unknown or undefined. This latter case is
        possible for example, when packet streams of
        different types are merged in a multipoint-to-point
        connection."
   REFERENCE
       "RFC 1700 - Assigned Numbers, Reynolds and Postel,
        October 1994."
   DEFVAL        { 0 }
   ::= { mplsInSegmentEntry 4 }

mplsInSegmentXCIndex OBJECT-TYPE
SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The index into mplsXCTable is used to identify which
        cross-connect entry this segment is part of.  Note
        that a value of zero indicates that it is not being
        referred to by any cross-connect entry."
   DEFVAL        { 0 }
   ::= { mplsInSegmentEntry 5 }

mplsInSegmentTSpecIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This variable represents a pointer into the
        mplsTSpecTable and indicates the TSpec which is to
        be assigned to this segment.  A value of zero
        indicates best-effort treatment.  Two or more
        segments can indicate resource sharing by pointing
        to the same entry in mplsTSpecTable."
   DEFVAL        { 0 }
   ::= { mplsInSegmentEntry 6 }

mplsInSegmentOwner OBJECT-TYPE
   SYNTAX        INTEGER {
                snmp(1),
                ldp(2),
                rsvp(3),
                policyAgent(4),
                other(5)
      }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes the entity that created and is responsible
        for managing this segment."
   ::= { mplsInSegmentEntry 7 }

mplsInSegmentAdminStatus OBJECT-TYPE
   SYNTAX        INTEGER {
                up(1),      -- ready to pass packets
                down(2),
                testing(3)  -- in some test mode
      }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value is used to represent the managerÆs
        desired operational status of this segment."
   ::= { mplsInSegmentEntry 8 }
```

```
mplsInSegmentOperStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),          -- ready to pass packets
         down(2),
         testing(3),     -- in some test mode
         unknown(4),     -- status cannot be determined for
                       -- some reason
         dormant(5),
         notPresent(6),  -- some component is missing
         lowerLayerDown(7)
                       -- down due to the state of
                       -- lower layer interfaces
      }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value represents the actual operational status
        of this segment."
   ::= { mplsInSegmentEntry 9 }

mplsInSegmentRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This variable is used to create, modify, and/or
        delete a row in this table."
   ::= { mplsInSegmentEntry 10 }

-- End of mplsInSegmentTable
-- In-segment performance table.

mplsInSegmentPerfTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsInSegmentPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table contains statistical information for
        incoming MPLS segments to an LSR."
   ::= { mplsLsrObjects 4 }

mplsInSegmentPerfEntry  OBJECT-TYPE
   SYNTAX        MplsInSegmentPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table contains statistical
        information about one incoming segment which was
        configured in the mplsInSegmentTable. The counters
        in this entry should behave in a manner similar to
        that of the interface."
   AUGMENTS      { mplsInSegmentEntry }
      ::= { mplsInSegmentPerfTable 1 }

MplsInSegmentPerfEntry ::= SEQUENCE {
      mplsInSegmentOctets             Counter32,
      mplsInSegmentPackets            Counter32,
      mplsInSegmentErrors             Counter32,
      mplsInSegmentDiscards           Counter32,

      -- high capacity counter
      mplsInSegmentHCOctets           Counter64
   }

mplsInSegmentOctets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value represents the total number of octets
        received by this segment."
   ::= { mplsInSegmentPerfEntry 1 }

mplsInSegmentPackets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Total number of packets received by this segment."
   ::= { mplsInSegmentPerfEntry 2 }

mplsInSegmentErrors OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The number of errored packets received on this
        segment."
   ::= { mplsInSegmentPerfEntry 3 }

mplsInSegmentDiscards OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The number of labeled packets received on this in-
        segment, which were chosen to be discarded even
        though no errors had been detected to prevent their
        being transmitted.  One possible reason for
        discarding such a labeled packet could be to free up
        buffer space."
   ::= { mplsInSegmentPerfEntry 4 }

mplsInSegmentHCOctets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The total number of octets received.  This is the 64
        bit version of mplsInSegmentOctets."
   ::= { mplsInSegmentPerfEntry 5 }

-- End of mplsInSegmentPerfTable.


-- Out-segment table.

mplsOutSegmentIndexNext OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This object contains the next appropriate value to
        be used for mplsOutSegmentIndex when creating
        entries in the mplsOutSegmentTable. If the number of
        unassigned entries is exhausted, this object will
        take on the value of 0.  To obtain the
        mplsOutSegmentIndex value for a new entry, the
        manager must first issue a management protocol
        retrieval operation to obtain the current value of
        this object. The agent should modify the value to
        reflect the next unassigned index after each
        retrieval operation. After a manager retrieves a
        value the agent will determine through its local
        policy when this index value will be made available
        for reuse."
::= { mplsLsrObjects 5 }


mplsOutSegmentTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsOutSegmentEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table contains a representation of the outgoing
        segments from an LSR."
   ::= { mplsLsrObjects 6 }

mplsOutSegmentEntry  OBJECT-TYPE
   SYNTAX        MplsOutSegmentEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents one incoming
        segment.  An entry can be created by a network
        administrator or an SNMP agent, or an MPLS signaling
        protocol. The creator of the entry is denoted by
        mplsOutSegmentOwner. An entry in this table is
        indexed by the ifIndex of the incoming interface and
        the (top) label.  Note that since it is possible
        that some segments are associated with a tunnel,
        traffic parameters of these rows are supported as
        read-only objects and their modification can be done
        only via the tunnel table, mplsTunnelTable."
   REFERENCE
       "MPLS Traffic Engineering Management Information Base
        Using SMIv2, Srinivasan, Viswanathan and Nadeau,
        draft-ietf-mpls-te-mib-02.txt, February 2000."

   INDEX         { mplsOutSegmentIndex }
      ::= { mplsOutSegmentTable 1 }

MplsOutSegmentEntry ::= SEQUENCE {
      mplsOutSegmentIndex                        Integer32,
      mplsOutSegmentIfIndex                      InterfaceIndex,
      mplsOutSegmentPushTopLabel                 TruthValue,
      mplsOutSegmentTopLabel                     MplsLabel,
      mplsOutSegmentNextHopIpAddrType            INTEGER,
      mplsOutSegmentNextHopIpv4Addr              IpAddress,
      mplsOutSegmentNextHopIpv6Addr              Ipv6Address,
      mplsOutSegmentXCIndex                      Integer32,
      mplsOutSegmentTSpecIndex                   Unsigned32,
      mplsOutSegmentOwner                        INTEGER,
      mplsOutSegmentAdminStatus                  INTEGER,
      mplsOutSegmentOperStatus                   INTEGER,
      mplsOutSegmentRowStatus                    RowStatus
   }

mplsOutSegmentIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value contains a unique index for this row.
        While a value of 0 is not valid as an index for this
        row it can be supplied as a valid value to index
        mplsXCTable to access entries for which no out-
        segment has been configured."
   ::= { mplsOutSegmentEntry 1 }

mplsOutSegmentIfIndex OBJECT-TYPE
   SYNTAX        InterfaceIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value contains the interface index of the
        outgoing interface."
   ::= { mplsOutSegmentEntry 2 }

mplsOutSegmentPushTopLabel OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value indicates whether or not a top label
        should be pushed onto the outgoing packet's label
        stack.  The value of this variable must be set to
        true if the outgoing interface is ATM, which does
        not support pop-and-go, or if it is a tunnel
        origination. Note that it is considered an error in
        the case that mplsOutSegmentPushTopLabel is set to
        false, but the cross-connect entry which refers to
        this out-segment has a non-zero mplsLabelStackIndex.
        The LSR should ensure that this situation cannot
        happen "
   ::= { mplsOutSegmentEntry 3 }

mplsOutSegmentTopLabel OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If mplsOutSegmentPushTopLabel is true then this is
        the label that should be pushed onto the outgoing
        packet's label stack.  Note that the contents of the
        label field can be interpreted in an outgoing
        interface specific fashion.  For example, the label
        carried in the MPLS shim header is 20 bits wide and
        the top 12 bits must be zero.  The Frame Relay label
        is 24 bits wide and the top 8 bits must be zero.
        For ATM interfaces the lowermost 16 bits are
        interpreted as the VCI, the next 8 bits as the VPI
        and the remaining bits must be zero."
   ::= { mplsOutSegmentEntry 4 }

mplsOutSegmentNextHopIpAddrType OBJECT-TYPE
   SYNTAX        INTEGER { none (1), ipV4 (2), ipV6 (3) }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Indicates whether the next hop address is IPv4 or
        IPv6.  Note that a value of none (1) is valid only
        when the outgoing interface is of type point-to-
        point."
   DEFVAL        { none }
   ::= { mplsOutSegmentEntry 5 }

mplsOutSegmentNextHopIpv4Addr OBJECT-TYPE
   SYNTAX        IpAddress
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "IPv4 Address of the next hop.  Its value is
        significant only when
        mplsOutSegmentNextHopIpAddrType is ipV4 (2),
        otherwise it should return a value of 0."
   ::= { mplsOutSegmentEntry 6 }

mplsOutSegmentNextHopIpv6Addr OBJECT-TYPE
   SYNTAX        Ipv6Address
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "IPv6 address of the next hop.  Its value is
        significant only when
        mplsOutSegmentNextHopIpAddrType is ipV6 (3),
        otherwise it should return a value of 0."
   ::= { mplsOutSegmentEntry 7 }

mplsOutSegmentXCIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Index into mplsXCTable which identifies which cross-
        connect entry this segment is part of.  A value of
        zero indicates that this entry is not referred to by
        any cross-connect entry."
   DEFVAL        { 0 }
   ::= { mplsOutSegmentEntry 8 }

mplsOutSegmentTSpecIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "A pointer into the mplsTSpecTable indicating the
        TSpec to be assigned for this segment.  A value of
        zero indicates best-effort treatment.  Two or more
        segments can indicate resource sharing by pointing
        to the same entry in mplsTSpecTable."
   DEFVAL        { 0 }
   ::= { mplsOutSegmentEntry 9 }


mplsOutSegmentOwner OBJECT-TYPE
   SYNTAX        INTEGER {
         snmp(1),
         ldp(2),
         rsvp(3),
         policyAgent(4),
         other(5)
      }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes the entity which created and is responsible
        for managing this segment."
```

```
::= { mplsOutSegmentEntry 10 }

mplsOutSegmentAdminStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),     -- ready to pass packets
         down(2),
         testing(3) -- in some test mode
      }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The desired operational status of this segment."
   ::= { mplsOutSegmentEntry 11 }

mplsOutSegmentOperStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),          -- ready to pass packets
         down(2),
         testing(3),     -- in some test mode
         unknown(4),     -- status cannot be determined for
                       -- some reason
         dormant(5),
         notPresent(6),  -- some component is missing
         lowerLayerDown(7)
                       -- down due to the state of
                       -- lower layer interfaces
      }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The actual operational status of this segment."
   ::= { mplsOutSegmentEntry 12 }
```

```
mplsOutSegmentRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsOutSegmentEntry 13 }

-- End of mplsOutSegmentTable


-- Out-segment performance table.

mplsOutSegmentPerfTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsOutSegmentPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table contains statistical information about
        incoming segments to an LSR. The counters in this
        entry should behave in a manner similar to that of
        the interface."
   ::= { mplsLsrObjects 7 }

mplsOutSegmentPerfEntry  OBJECT-TYPE
   SYNTAX        MplsOutSegmentPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table contains statistical
        information about one incoming segment configured in
        mplsOutSegmentTable."
   AUGMENTS      { mplsOutSegmentEntry }
      ::= { mplsOutSegmentPerfTable 1 }

MplsOutSegmentPerfEntry ::= SEQUENCE {
      mplsOutSegmentOctets              Counter32,
      mplsOutSegmentPackets             Counter32,
      mplsOutSegmentErrors              Counter32,
      mplsOutSegmentDiscards            Counter32,

      -- HC counter
      mplsOutSegmentHCOctets            Counter64
   }

mplsOutSegmentOctets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value contains the total number of octets sent
        on this segment."
   ::= { mplsOutSegmentPerfEntry 1 }

mplsOutSegmentPackets OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This value contains the total number of packets sent
        on this segment."
   ::= { mplsOutSegmentPerfEntry 2 }
```

```
mplsOutSegmentErrors OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of packets that could not be sent due to
        errors on this segment."
   ::= { mplsOutSegmentPerfEntry 3 }

mplsOutSegmentDiscards OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The number of labeled packets received on this out-
        segment, which were chosen to be discarded even
        though no errors had been detected to prevent their
        being transmitted.  One possible reason for
        discarding such a labeled packet could be to free up
        buffer space."
   ::= { mplsOutSegmentPerfEntry 4 }

mplsOutSegmentHCOctets OBJECT-TYPE
   SYNTAX        Counter64
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Total number of octets sent.  This is the 64 bit
        version of mplsOutSegmentOctets."
   ::= { mplsOutSegmentPerfEntry 5 }

-- End of mplsOutSegmentPerfTable.
-- Cross-connect table.


mplsXCIndexNext OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This object contains an appropriate value to be used
        for mplsXCIndex when creating entries in the
        mplsXCTable. The value 0 indicates that no
        unassigned entries are available. To obtain the
        value of mplsXCIndex for a new entry in the
        mplsXCTable, the manager issues a management
        protocol retrieval operation to obtain the current
        value of mplsXCIndex. After each retrieval
        operation, the agent should modify the value to
        reflect the next unassigned index. After a manager
        retrieves a value the agent will determine through
        its local policy when this index value will be made
        available for reuse."
::= { mplsLsrObjects 8 }

mplsXCTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsXCEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table specifies information for switching
        between LSP segments.  It supports point-to-point,
        point-to-multipoint and multipoint-to-point
        connections.  mplsLabelStackTable specifies the
        label stack information for a cross-connect LSR and
        is referred to from mplsXCTable."
   ::= { mplsLsrObjects 9 }

mplsXCEntry  OBJECT-TYPE
   SYNTAX        MplsXCEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "A row in this table represents one cross-connect
        entry.  The following objects index it:

        - cross-connect index mplsXCIndex that uniquely
          identifies a group of cross-connect entries
        - interface index of the in-segment,
        mplsInSegmentIfIndex
        - incoming label(s), mplsInSegmentLabel
        - out-segment index, mplsOutSegmentIndex

       Originating LSPs:
        These are represented by using the special
        combination of values mplsInSegmentIfIndex=0 and
        mplsInSegmentLabel=0 as indexes.  In this case the
        mplsOutSegmentIndex MUST be non-zero.

       Terminating LSPs:
       These are represented by using the special value
        mplsOutSegmentIndex=0 as index.

       Special labels:
        Entries indexed by reserved MPLS label values 0
        through 15 imply terminating LSPs and MUST have
        mplsOutSegmentIfIndex = 0. Note that situations
        where LSPs are terminated with incoming label equal
        to 0, should have mplsInSegmentIfIndex = 0 as well,
        but can be distinguished from originating LSPs
        because the mplsOutSegmentIfIndex = 0. The
        mplsOutSegmentIfIndex MUST only be set to 0 in
        cases of terminating LSPs.

        An entry can be created by a network administrator
        or by an SNMP agent as instructed by an MPLS
        signaling protocol."
   INDEX         { mplsXCIndex, mplsInSegmentIfIndex,
                mplsInSegmentLabel, mplsOutSegmentIndex }
      ::= { mplsXCTable 1 }

MplsXCEntry ::= SEQUENCE {
      mplsXCIndex              Integer32,
      mplsXCLspId              MplsLSPID,
      mplsXCLabelStackIndex     Integer32,
      mplsXCCOS            Integer32,
      mplsXCIsPersistent       TruthValue,
      mplsXCOwner              INTEGER,
      mplsXCAdminStatus        INTEGER,
      mplsXCOperStatus      INTEGER,
      mplsXCRowStatus          RowStatus
   }

mplsXCIndex OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Primary index for the row identifying a group of
        cross-connect segments."
   ::= { mplsXCEntry 1 }

mplsXCLspId OBJECT-TYPE
   SYNTAX        MplsLSPID
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value identifies the label switched path that
        this cross-connect entry belongs to."
   ::= { mplsXCEntry 2 }

mplsXCLabelStackIndex OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Primary index into mplsLabelStackTable identifying a
        stack of labels to be pushed beneath the top label.
        Note that the top label identified by the out-
        segment ensures that all the components of a
        multipoint-to-point connection have the same
        outgoing label.  A value of 0 indicates that no
        labels are to be stacked beneath the top label."
   ::= { mplsXCEntry 3 }

mplsXCCOS OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value is used to override the incoming COS
        field for a cross-connect. It may also be used as a
        value to assign to outgoing packets for an outgoing
        segment of a tunnel. Note that packet treatment at
        this LSR is determined by the incoming COS value and
        the new COS value only impacts packet treatment at a
        downstream LSR."
   ::= { mplsXCEntry 4 }

mplsXCIsPersistent OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes whether or not this cross-connect entry and
        associated in- and out-segments should be restored
        automatically after failures. This value MUST be set
        to false in cases where this cross-connect entry was
        created by a signaling protocol."
   DEFVAL        { false }
   ::= { mplsXCEntry 5 }

mplsXCOwner OBJECT-TYPE
   SYNTAX        INTEGER {
                snmp(1),
                ldp(2),
                rsvp(3),
                policyAgent(4),
                other(5)
      }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes the entity that created and is responsible
        for managing this cross-connect."
   ::= { mplsXCEntry 6 }

mplsXCAdminStatus OBJECT-TYPE
   SYNTAX        INTEGER {
                up(1),     -- ready to pass packets
                down(2),
                testing(3) -- in some test mode
                }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The desired operational status of this segment."
   ::= { mplsXCEntry 7 }

mplsXCOperStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),          -- ready to pass packets
         down(2),
         testing(3),     -- in some test mode
         unknown(4),     -- status cannot be determined for
                       -- some reason
         dormant(5),
         notPresent(6),  -- some component is missing
         lowerLayerDown(7)
                       -- down due to the state of
                       -- lower layer interfaces
      }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The actual operational status of this cross-
        connect."
   ::= { mplsXCEntry 8 }

mplsXCRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsXCEntry 9 }

-- End of mplsXCTable


-- Label stack table.
mplsMaxLabelStackDepth  OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The maximum stack depth supported by this LSR."
::= { mplsLsrObjects 10 }

mplsLabelStackIndexNext  OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
```

```
       "This object contains an appropriate value to be used
        for mplsLabelStackIndex when creating entries in the
        mplsLabelStackTable.  The value 0 indicates that no
        unassigned entries are available.  To obtain an
        mplsLabelStackIndex value for a new entry, the
        manager issues a management protocol retrieval
        operation to obtain the current value of this
        object. After each retrieval operation, the agent
        should modify the value to reflect the next
        unassigned index. After a manager retrieves a value
        the agent will determine through its local policy
        when this index value will be made available for
        reuse."
::= { mplsLsrObjects 11 }

mplsLabelStackTable OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsLabelStackEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table specifies the label stack to be pushed
        onto a packet, beneath the top label.  Entries into
        this table are referred to from mplsXCTable."
   ::= { mplsLsrObjects 12 }

mplsLabelStackEntry OBJECT-TYPE
   SYNTAX        MplsLabelStackEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents one label which is
        to be pushed onto an outgoing packet, beneath the
        top label.  An entry can be created by a network
        administrator or by an SNMP agent as instructed by
        an MPLS signaling protocol."
   INDEX         { mplsLabelStackIndex, mplsLabelStackLabelIndex }
      ::= { mplsLabelStackTable 1 }

MplsLabelStackEntry ::= SEQUENCE {
      mplsLabelStackIndex             Integer32,
      mplsLabelStackLabelIndex        Integer32,
      mplsLabelStackLabel             MplsLabel,
      mplsLabelStackRowStatus         RowStatus
   }

mplsLabelStackIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Primary index for this row identifying a stack of
        labels to be pushed on an outgoing packet, beneath
        the top label."
   ::= { mplsLabelStackEntry 1 }

mplsLabelStackLabelIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "Secondary index for this row identifying one label
        of the stack. Note that an entry with a smaller
        mplsLabelStackLabelIndex would refer to a label
        higher up the label stack and would be popped at a
        downstream LSR before a label represented by a
        higher mplsLabelStackLabelIndex at a downstream
        LSR."
   ::= { mplsLabelStackEntry 2 }

mplsLabelStackLabel OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The label to pushed."
   ::= { mplsLabelStackEntry 3 }

mplsLabelStackRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsLabelStackEntry 4 }

-- End of mplsLabelStackTable

-- TSpec table.

mplsTSpecIndexNext OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This object contains an appropriate value which will
        be used for mplsTSpecIndex when creating entries in
        the mplsTSpecTable.  The value 0 indicates that no
        unassigned entries are available.  To obtain the
        mplsTSpecIndex value for a new entry, the manager
        issues a management protocol retrieval operation to
        obtain the current value of this object. After each
        retrieval operation, the agent should modify the
        value to reflect the next unassigned index. After a
        manager retrieves a value the agent will determine
        through its local policy when this index value will
        be made available for reuse."
::= { mplsLsrObjects 13 }

mplsTSpecTable OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsTSpecEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table specifies the Traffic Specification
        (TSpec) objects for in and out-segments."
   ::= { mplsLsrObjects 14 }

mplsTSpecEntry OBJECT-TYPE
   SYNTAX        MplsTSpecEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents the TSpec objects
        for one or more in or out segments.  A single entry
        can be pointed to by multiple segments indicating
        resource sharing."
   INDEX         { mplsTSpecIndex }
      ::= { mplsTSpecTable 1 }
```

```
MplsTSpecEntry ::= SEQUENCE {
      mplsTSpecIndex                  Integer32,
      mplsTSpecMaxRate                MplsBitRate,
      mplsTSpecMeanRate               MplsBitRate,
      mplsTSpecMaxBurstSize           MplsBurstSize,
      mplsTSpecRowStatus              RowStatus
   }

mplsTSpecIndex OBJECT-TYPE
   SYNTAX        Integer32 (1..2147483647)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Uniquely identifies this row of the table.  Note
        that zero represents an invalid index."
   ::= { mplsTSpecEntry 1 }

mplsTSpecMaxRate OBJECT-TYPE
   SYNTAX        MplsBitRate
   UNITS         "bits per second"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Maximum rate in bits/second."
   ::= { mplsTSpecEntry 4 }

mplsTSpecMeanRate OBJECT-TYPE
   SYNTAX        MplsBitRate
   UNITS         "bits per second"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Mean rate in bits/second."
   ::= { mplsTSpecEntry 5 }

mplsTSpecMaxBurstSize OBJECT-TYPE
   SYNTAX        MplsBurstSize
   UNITS         "bytes"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Maximum burst size in bytes."
   ::= { mplsTSpecEntry 6 }

mplsTSpecRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsTSpecEntry 7 }

-- End of mplsTSpecTable

-- Notification Configuration

mplsInterfaceTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "If this object is true, then it enables the
        generation of mplsInterfaceUp and mplsInterfaceDown
        traps, otherwise these traps are not emitted."
        DEFVAL  { false }
::= { mplsLsrObjects 15 }

mplsInSegmentTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "If this object is true, then it enables the
        generation of mplsInSegmentUp and mplsInSegmentDown
        traps, otherwise these traps are not emitted."
        DEFVAL { false }
::= { mplsLsrObjects 16 }


mplsOutSegmentTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "If this object is true, then it enables the
        generation of mplsOutSegmentUp and
        mplsOutSegmentDown traps, otherwise these traps are
        not emitted."
        DEFVAL { false }
::= { mplsLsrObjects 17 }

mplsXCTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "If this object is true, then it enables the
        generation of mplsXCUp and mplsXCDown traps,
        otherwise these traps are not emitted."
        DEFVAL { false }
::= { mplsLsrObjects 18 }

-- Interface

mplsInterfaceUp NOTIFICATION-TYPE
   OBJECTS     { mplsInterfaceConfIndex,
                mplsInterfaceAdminStatus, mplsInterfaceOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsInterfaceOperStatus object for one of the
        entries in mplsInterfaceConfTable is about to leave
        the down state and transition into some other state
        (but not into the notPresent state).  This other
        state is indicated by the included value of
        mplsInterfaceOperStatus."
   ::= { mplsLsrNotifications 1 }

mplsInterfaceDown NOTIFICATION-TYPE
   OBJECTS     { mplsInterfaceConfIndex,
                mplsInterfaceAdminStatus, mplsInterfaceOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsInterfaceOperStatus object for one of the
        entries in mplsInterfaceConfTable is about to enter
        the down state from some other state (but not from
        the notPresent state).  This other state is
        indicated by the included value of
        mplsInterfaceOperStatus."
   ::= { mplsLsrNotifications 2 }

-- In-segment.

mplsInSegmentUp NOTIFICATION-TYPE
   OBJECTS     { mplsInSegmentIfIndex, mplsInSegmentLabel,
                 mplsInSegmentAdminStatus, mplsInSegmentOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsInSegmentOperStatus object for one of the
        configured in-segments is about to leave the down
        state and transition into some other state (but not
        into the notPresent state).  This other state is
        indicated by the included value of
        mplsInSegmentOperStatus."
   ::= { mplsLsrNotifications 3 }

mplsInSegmentDown NOTIFICATION-TYPE
   OBJECTS     { mplsInSegmentIfIndex, mplsInSegmentLabel,
                 mplsInSegmentAdminStatus, mplsInSegmentOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsInSegmentOperStatus object for one of the
        configured in-segments is about to enter the down
        state from some other state (but not from the
        notPresent state).  This other state is indicated by
        the included value of mplsInSegmentOperStatus."
   ::= { mplsLsrNotifications 4 }

-- Out-segment.

mplsOutSegmentUp NOTIFICATION-TYPE
   OBJECTS     { mplsOutSegmentIndex, mplsInSegmentAdminStatus,
                mplsInSegmentOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsOutSegmentOperStatus object for one of the
        configured out-segments is about to leave the down
        state and transition into some other state (but not
        into the notPresent state).  This other state is
        indicated by the included value of
        mplsOutSegmentOperStatus."
   ::= { mplsLsrNotifications 5 }

mplsOutSegmentDown NOTIFICATION-TYPE
   OBJECTS     { mplsOutSegmentIndex, mplsInSegmentAdminStatus,
                mplsInSegmentOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsOutSegmentOperStatus object for one of the
        configured out-segments is about to enter the down
        state from some other state (but not from the
        notPresent state).  This other state is indicated by
        the included value of mplsOutSegmentOperStatus."
   ::= { mplsLsrNotifications 6 }

-- Cross-connect.
```

```
mplsXCUp NOTIFICATION-TYPE
   OBJECTS     { mplsXCIndex,
                 mplsInSegmentIfIndex, mplsInSegmentLabel,
                 mplsOutSegmentIndex,
                 mplsXCAdminStatus, mplsXCOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsXCOperStatus object for one of the configured
        cross-connect entries is about to leave the down
        state and transition into some other state (but not
        into the notPresent state).  This other state is
        indicated by the included value of
        mplsXCOperStatus."
   ::= { mplsLsrNotifications 7 }

mplsXCDown NOTIFICATION-TYPE
   OBJECTS     { mplsXCIndex,
                 mplsInSegmentIfIndex, mplsInSegmentLabel,
                 mplsOutSegmentIndex,
                 mplsXCAdminStatus, mplsXCOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsXCOperStatus object for one of the configured
        cross-connect entries is about to enter the down
        state from some other state (but not from the
        notPresent state).  This other state is indicated by
        the included value of mplsXCOperStatus."
   ::= { mplsLsrNotifications 8 }
```

```
-- End of notifications.


-- Module compliance.

mplsLsrGroups
   OBJECT IDENTIFIER ::= { mplsLsrConformance 1 }

mplsLsrCompliances
   OBJECT IDENTIFIER ::= { mplsLsrConformance 2 }

mplsLsrModuleCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
       "Compliance statement for agents that support the
        MPLS LSR MIB."
   MODULE -- this module

      -- The mandatory groups have to be implemented by all LSRs.
      -- However, they may all be supported as read-only objects
      -- in the case where manual configuration is unsupported.

      MANDATORY-GROUPS    { mplsInSegmentGroup, mplsOutSegmentGroup,
                            mplsXCGroup, mplsInterfaceGroup,
                            mplsPerfGroup }

      GROUP mplsHCInterfacePerfGroup
      DESCRIPTION
          "This group is mandatory for high-speed MPLS
           capable interfaces for which the objects
           mplsInterfaceInOctets and mplsInterfaceOutOctets
           wrap around too quickly."
      GROUP mplsHCInSegmentPerfGroup
      DESCRIPTION
          "This group is mandatory for those in-segment
           entries for which the object
           mplsInSegmentOutOctets wraps around too
           quickly."

      GROUP mplsHCOutSegmentPerfGroup
      DESCRIPTION
          "This group is mandatory for those out-segment
           entries for which the object
           mplsOutSegmentOctets wraps around too quickly."

      GROUP mplsTSpecGroup
      DESCRIPTION
          "This group is mandatory for those LSRs that
           support Int-Serv style resource reservation."

      -- Depending on whether the device implements persistent
      -- cross-connects or not one of the following two groups
      -- is mandatory.

      GROUP mplsXCIsPersistentGroup
      DESCRIPTION
          "This group is mandatory for devices which
           support persistent cross-connects.  The
           following constraints apply: mplsXCIsPersistent
           must at least be read-only returning true(2)."

      GROUP mplsXCIsNotPersistentGroup
      DESCRIPTION
```

```
          "This group is mandatory for devices which
           support non-persistent cross-connects.  The
           following constraints apply: mplsXCIsPersistent
           must at least be read-only returning false(1)."


      -- mplsInterfaceConfTable

      OBJECT      mplsInterfaceAdminStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "A value of testing(3) need not be supported."

      OBJECT      mplsInterfaceOperStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Only up(1) and down(2) need to be supported."


      -- mplsInSegmentTable

      OBJECT      mplsInSegmentIfIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsInSegmentLabel
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."
      OBJECT     mplsInSegmentXCIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsInSegmentTSpecIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsInSegmentNPop
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access if not required.  This object
           should be set to 1 if it is read-only."

      OBJECT      mplsInSegmentAddrFamily
      SYNTAX      INTEGER { other(0) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required.  A value of
           other(0) should be supported."

      OBJECT      mplsInSegmentAdminStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "A value of testing(3) need not be supported."

      OBJECT      mplsInSegmentOperStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Only up(1) and down(2) need to be supported."


      -- mplsOutSegmentTable

      OBJECT mplsOutSegmentIndexNext
      MIN-ACCESS    read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentIfIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentPushTopLabel
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentTopLabel
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."
      OBJECT      mplsOutSegmentNextHopIpAddrType
      SYNTAX      INTEGER { none(1), ipV4(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "ipV6(3) need not be supported."

      OBJECT      mplsOutSegmentNextHopIpv4Addr
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentNextHopIpv6Addr
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentXCIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentTSpecIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsOutSegmentAdminStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "A value of testing(3) need not be supported."
      OBJECT      mplsOutSegmentOperStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Only up(1) and down(2) need to be supported."

      OBJECT      mplsOutSegmentRowStatus
      SYNTAX      INTEGER { active(1), notInService(2),
                            createAndGo(4), destroy(6) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "The notReady(3) and createAndWait(5) states need
           not be supported."


      -- mplsXCTable

      OBJECT mplsXCIndexNext
      MIN-ACCESS    read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsXCIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT mplsXCLabelStackIndexNext
      MIN-ACCESS    read-only
      DESCRIPTION
          "Write access is not required."
      OBJECT      mplsXCLabelStackIndex
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsXCCOS
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsXCIsPersistent
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsXCAdminStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "A value of testing(3) need not be supported."

      OBJECT      mplsXCOperStatus
      SYNTAX      INTEGER { up(1), down(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Only up(1) and down(2) need to be supported."

      OBJECT      mplsXCRowStatus
      SYNTAX      INTEGER { active(1), notInService(2),
                            createAndGo(4), destroy(6) }
      MIN-ACCESS  read-only
      DESCRIPTION
```

```
          "The notReady(3) and createAndWait(5) states need
           not be supported."

   ::= { mplsLsrCompliances 1 }


-- Units of conformance.

mplsInterfaceGroup OBJECT-GROUP
   OBJECTS { mplsInterfaceConfIndex,
             mplsInterfaceLabelMinIn, mplsInterfaceLabelMaxIn,
             mplsInterfaceLabelMinOut, mplsInterfaceLabelMaxOut,
             mplsInterfaceInLabelsUsed, mplsInterfaceOutLabelsUsed,
             mplsInterfaceIsGlobalLabelSpace,
             mplsInterfaceIsLocalLabelSpace,
             mplsInterfaceAdminStatus, mplsInterfaceOperStatus,
             mplsInterfaceTrapEnable
          }

   STATUS  current
   DESCRIPTION
          "Collection of objects needed for MPLS interface
           configuration and performance information."
   ::= { mplsLsrGroups 1 }

mplsInSegmentGroup  OBJECT-GROUP
   OBJECTS { mplsInSegmentIfIndex,
             mplsInSegmentLabel,
             mplsInSegmentNPop,
             mplsInSegmentAddrFamily,
             mplsInSegmentXCIndex,
             mplsInSegmentTSpecIndex,
             mplsInSegmentOctets,
             mplsInSegmentDiscards,
             mplsInSegmentOwner,
             mplsInSegmentAdminStatus,
             mplsInSegmentOperStatus,
             mplsInSegmentRowStatus,
             mplsInSegmentTrapEnable
          }
   STATUS  current
   DESCRIPTION
          "Collection of objects needed to implement an in-
           segment."
   ::= { mplsLsrGroups 2 }

mplsOutSegmentGroup  OBJECT-GROUP
   OBJECTS { mplsOutSegmentIndexNext,
             mplsOutSegmentIndex,
             mplsOutSegmentIfIndex,
             mplsOutSegmentPushTopLabel,
             mplsOutSegmentTopLabel,
             mplsOutSegmentNextHopIpAddrType,
             mplsOutSegmentNextHopIpv4Addr,
             mplsOutSegmentNextHopIpv6Addr,
             mplsOutSegmentXCIndex,
             mplsOutSegmentTSpecIndex,
             mplsOutSegmentOwner,
             mplsOutSegmentOctets,
             mplsOutSegmentDiscards,
             mplsOutSegmentAdminStatus,
             mplsOutSegmentOperStatus,
             mplsOutSegmentRowStatus,
             mplsOutSegmentTrapEnable
           }
   STATUS  current
   DESCRIPTION
          "Collection of objects needed to implement an out-
           segment."
   ::= { mplsLsrGroups 3 }

mplsXCGroup  OBJECT-GROUP
   OBJECTS { mplsXCIndexNext,
             mplsXCIndex,
             mplsXCLabelStackIndex,
             mplsXCAdminStatus,
             mplsXCOperStatus,
             mplsXCRowStatus,
             mplsXCTrapEnable
          }
   STATUS  current
   DESCRIPTION
          "Collection of objects needed to implement a
           cross-connect entry."
   ::= { mplsLsrGroups 4 }

mplsPerfGroup OBJECT-GROUP
   OBJECTS { mplsInterfaceInPackets,
             mplsInterfaceInDiscards,
             mplsInterfaceOutPackets,
             mplsInterfaceOutDiscards,
             mplsInSegmentOctets,
             mplsInSegmentPackets,
             mplsInSegmentDiscards,
             mplsOutSegmentOctets,
             mplsOutSegmentPackets,
             mplsOutSegmentDiscards }
   STATUS  current
   DESCRIPTION
          "Collection of objects providing performance
           information
        about an LSR."
   ::= { mplsLsrGroups 5 }

mplsHCInSegmentPerfGroup OBJECT-GROUP
   OBJECTS { mplsInSegmentHCOctets }
   STATUS  current
   DESCRIPTION
          "Object(s) providing performance information
           specific to out-segments for which the object
           mplsInterfaceInOctets wraps around too quickly."
   ::= { mplsLsrGroups 6 }

mplsHCOutSegmentPerfGroup OBJECT-GROUP
   OBJECTS { mplsOutSegmentHCOctets }
   STATUS  current
   DESCRIPTION
          "Object(s) providing performance information
           specific to out-segments for which the object
           mplsInterfaceOutOctets wraps around too
           quickly."
   ::= { mplsLsrGroups 7 }

mplsTSpecGroup OBJECT-GROUP
   OBJECTS { mplsTSpecIndex,
             mplsTSpecMaxRate,
             mplsTSpecMeanRate,
             mplsTSpecMaxBurstSize,
             mplsTSpecRowStatus }
   STATUS  current
   DESCRIPTION
          "Object(s) required for supporting Int-Serv style
           resource reservation."
   ::= { mplsLsrGroups 8 }

mplsXCIsPersistentGroup OBJECT-GROUP
   OBJECTS { mplsXCIsPersistent }
   STATUS  current
   DESCRIPTION
          "Objects needed to support persistent cross-
           connects."
   ::= { mplsLsrGroups 9 }

mplsXCIsNotPersistentGroup OBJECT-GROUP
   OBJECTS { mplsXCIsPersistent }
   STATUS  current
   DESCRIPTION
          "Objects needed to support non-persistent cross-
           connects."
   ::= { mplsLsrGroups 10 }

mplsLsrNotificationGroup NOTIFICATION-GROUP
   NOTIFICATIONS { mplsInterfaceUp,
                  mplsInterfaceDown,
                  mplsInSegmentUp,
                  mplsInSegmentDown,
                  mplsOutSegmentUp,
                  mplsXCUp,
                  mplsXCDown
                  }
   STATUS  current
   DESCRIPTION
          "Set of notifications implemented in this module.
           None is mandatory."
   ::= { mplsLsrGroups 11 }

-- End of MPLS-LSR-MIB
END
```

10.   Security Considerations

It is clear that this MIB is potentially useful for monitoring of MPLS LSRs. This MIB can also be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

At this writing, no security holes have been identified beyond those that SNMP Security [SNMPArch] is itself intended to address. These relate to primarily controlled access to sensitive information and the ability to configure a device - or which might result from operator error, which is beyond the scope of any security architecture.

There are a number of management objects defined in this MIB which have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The use of SNMP Version 3 is recommended over prior versions, for configuration control, as its security model is improved.

SNMPv1 or SNMPv2 are by themselves not a secure environment. Even if the network itself is secure (for example by using IPSec [IPSEC]), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [SNMPv3USM] and the View-based Access Control [SNMPv3VACM] is recommended. It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

There are a number of managed objects in this MIB that may contain information that may be sensitive from a business perspective, in that they represent a customer's interface to the MPLS network. Allowing uncontrolled access to these objects could result in malicious and unwanted disruptions of network traffic or incorrect configurations for these customers. There are no objects that are particularly sensitive in their own right, such as passwords or monetary amounts.

11. Acknowledgments

We wish to thank Ron Bonica, Keith McCloghrie, Dan Tappan, Bala Rajagopalan, Eric Gray, Vasanthi Thirumalai and Adrian Farrel.

12. References

[MPLSArch] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-06.txt>, August 1999.

[MPLSFW] Callon, R., Doolan, P., Feldman, N., Fredette, A., Swallow, G., and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Internet Draft <draft-ietf-mpls-framework-05.txt>, September 1999.

[TEMIB] Srinivasan, C., Viswanathan, A. and Nadeau, T., "MPLS Traffic Engineering Management Information Base Using SMIv2", Internet Draft <draft-ietf-mpls-te-mib-02.txt>, February 2000.

[LDPMIB] Cucchiara, J., Sjostrand, H., and J. Luciani, " Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)", Internet Draft <draft-ietf-mpls-ldp-mib-05.txt>, February 2000.

[LblStk] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Federokow, G., Li, T., and A. Conta, "MPLS Label Stack Encoding", Internet Draft <draft-ietf-mpls-label-encaps-07.txt>, September 1999.

[RSVPTun] Awaduche, D., Berger, L., Der-Haw, G., Li, T., Swallow, G., and V. Srinivasan, "Extensions to RSVP for LSP Tunnels", Internet Draft <draft-mpls-rsvp-

lsp-tunnel-04.txt>, September 1999.

[CRLDP] B. Jamoussi (Editor), "Constraint-Based LSP Setup using LDP", Internet Draft <draft-ietf-mpls-cr-ldp-03.txt>, September 1999.

[Assigned] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1700, October 1994. See also: http://www.isi.edu/in-notes/iana/assignments/smi-numbers

[SNMPArch] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.

[SMIv1] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, May 1990.

[SNMPv1MIBDef]Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, March 1991.

[SNMPv1Traps] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[SMIv2] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[SNMPv2TC] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, SNMP Research, Inc., Cisco Systems, Inc., January 1996.

[SNMPv2Conf] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.

[SNMPv1] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, May 1990.

[SNMPv2c] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[SNMPv2TM] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[SNMPv3MP] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, January 1998.

[SNMPv3USM] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[SNMPv2PO]   Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[SNMPv3App]   Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2273, January 1998.

[SNMPv3VACM]   Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[IPSEC]   Kent, S., and Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[IFMIB]   McCloghrie, K., and F. Kastenholtz, "The Interfaces Group MIB using SMIv2", RFC 2233, Nov. 1997

13. Authors' Addresses

Cheenu Srinivasan
Tachion Networks, Inc.
2 Meridian Road
Eatontown, NJ 07724
Phone: +1-732-542-7750 x234
Email: cheenu@tachion.com

Arun Viswanathan
Force10 Networks

1440 McCarthy Blvd
Milpitas, CA 95035
Phone: +1-408-571-3516
Email: arun@force10networks.com

Thomas D. Nadeau
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Phone: +1-978-244-3051
Email: tnadeau@cisco.com

14. Full Copyright Statement