Abstract

This memo defines an portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects to configure and/or monitor a Multiprotocol Label Switching (MPLS) Label Switching Router (LSR).
1. Introduction

This memo defines an 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 Multiprotocol Label Switching (MPLS) [RFC3031] Label Switching Router (LSR).

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119, reference [RFC2119].
2. Terminology

This document uses terminology from the document describing the MPLS architecture [RFC3031]. 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 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.

- **in-segment**: This is analogous to an MPLS label.
- **out-segment**: This is analogous to an MPLS label.
- **cross-connect**: This describes the conceptual connection between a set of in-segments and out-segments. Note that either set may be 0; that is, a cross-connect may connect only out-segments together with no in-segments in the case where an LSP is originating on an LSR.

3. The SNMP Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. 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.

4.1. Summary of LSR MIB Module

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

- The interface table (mplsInterfaceTable), which is used for revealing 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.

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.

5. Brief Description of MIB Module Objects

Sections 5.1-5.2 describe objects pertaining to MPLS-capable interfaces of an LSR. The objects described in Sections 5.3-5.8, were derived from the Incoming Label Map (ILM) and Next Hop Label Forwarding Entry (NHLFE) as specified in the MPLS architecture document [RFC3031]. It is appropriate to note that the in-segment, out-segment, and cross-connect tables were modeled after similar tables found in [RFC2515].

5.1. mplsInterfaceTable

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.

5.2. mplsInterfacePerfTable

This table contains objects to measure the MPLS performance of MPLS capable interfaces and is an AUGMENT to mplsInterfaceTable.
5.3. mplsInSegmentTable

This table contains a description of the incoming MPLS segments to an LSR and their associated parameters. This index for this table is mplsInSegmentIndex. The index structure of this table is specifically designed to handle many different MPLS implementations that manage their labels both in a distributed and centralized manner.

The table is designed to handle existing MPLS labels as well as future label strategies that may require labels longer than the ones defined in RFC3031. In these cases, the object mplsInSegmentLabelPtr may be used indicate the first accessible object in a separate table that can be used to represent the label because it is too long to be represented in a single 32-bit value (mplsInSegmentLabel).

5.4. 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.

5.5. mplsOutSegmentTable

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

5.6. 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.

5.7. 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-multipoint and multipoint-to-point connections.
The operational status object indicates the packet forwarding state of a cross-connect entry. For example, when the operational status objects is ‘down’ it indicates that the specified cross-connect entry will not forward packets. Likewise, when it is set to ‘up’ it indicates that packets will be forwarded.

The administrative status object indicates the forwarding state desired by the operator.

5.8. 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.

5.9 mplsInSegmentMapTable

The mplsInSegmentMapTable specifies the mapping from the mplsInSegmentIndex to the corresponding mplsInSegmentInterface and mplsInSegmentLabel objects. The purpose of this table is to provide the manager with an alternative means by which to locate in-segments. For instance, this table can be useful when tracing LSPs from LSR to LSR by first following the in-segment to out-segment, retrieving the outgoing label and out-going interface, and then proceeding to interrogate this table at the next-hop LSR to continue the trace.

6. Use of 32-bit and 64-bit Counters

64-bit counters are provided in this MIB module for high speed interfaces where the use of 32-bit counters might be impractical. The requirements on the use of 32-bit and 64-bit counters (copied verbatim from [RFC2863]) are as follows.

For interfaces that operate at 20,000,000 (20 million) bits per second or less, 32-bit byte and packet counters MUST be supported. For interfaces that operate faster than 20,000,000 bits/second, and slower than 650,000,000 bits/second, 32-bit packet counters MUST be supported and 64-bit octet counters MUST be supported. For interfaces that operate at 650,000,000 bits/second or faster, 64-bit packet counters AND 64-bit octet counters MUST be supported.

7. Example of LSP Setup

In this section we provide a brief example of setting up an LSP using this MIB module’s objects. While this example is not meant to illustrate every nuance of the MIB module, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB module 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 impose any additional 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. Those objects relevant to illustrating the relationships amongst different tables are shown here. Other objects may be needed before conceptual row activation can happen.

The RowStatus values shown in this section are those to be used in the set request, typically createAndGo(4) which is used to create the conceptual row and have its status immediately set to active. Note that the proper use of createAndGo(4) requires that all columns that do not have a DEFVAL to be specified in order for the SET to succeed. In the example below we have not specify all such columns for the sake of keeping the example short. Please keep in mind that all such fields must be send during a real SET operation. A subsequent retrieval operation on the conceptual row will return a different value, such as active(1). Please see [RFC2579] for a detailed discussion on the use of RowStatus.

We first create a cross-connect entry that associates the desired segments together.

In mplsXCTable:

```
{  
  mplsXCIndex = 0x02,  
  mplsXCInSegmentIndex = 0x00000015,  
  mplsXCOutSegmentIndex = 0x01,  
  mplsXCLspId = 0x0102 -- unique ID  
  mplsXLabelStackIndex = 0x00, -- only a single  
                          -- outgoing label  
  mplsXCRowStatus = createAndGo(4)  
}
```

Next, we create the appropriate in-segment and out-segment entries based on the cross-connect. Note that some agents may wish to automatically create the in and out-segments based on the cross-connect creation.

In mplsInSegmentTable:

```
{  
  mplsInSegmentIndex = 0x00000015  
  mplsInSegmentLabel = 21, -- incoming label  
}  
```
mplsInSegmentNPop = 1,
mplsInSegmentInterface = 12, -- incoming interface

-- RowPointer MUST point to the first accessible column.
mplsInSegmentLabelPtr = 0.0,
mplsInSegmentTrafficParamPtr = 0.0,
mplsInSegmentRowStatus = createAndGo(4)
}

In mplsOutSegmentTable:
{
    mplsOutSegmentIndex = 0x01,
mplsOutSegmentInterface = 13, -- outgoing interface
    mplsOutSegmentPushTopLabel = true(1),
mplsOutSegmentTopLabel = 22, -- outgoing label

    -- RowPointer MUST point to the first accessible column.
    mplsOutSegmentTrafficParamPtr = 0.0,
mplsOutSegmentLabelPtr = 0.0,
mplsOutSegmentRowStatus = createAndGo(4)
}

Note that the mplsInSegmentXCIndex and mplsOutSegmentXCIndex objects will automatically be populated with the string 0x02 when these segments are referred to from the corresponding cross-connect entry.

8. Application of the Interface Group to MPLS

RFC2863 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 [RFC2863] 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 interface is represented as an entry in the ifTable. The inter-relation of entries in the ifTable is defined by Interfaces Stack Group defined in [RFC2863].
When using MPLS interfaces, the interface stack table might appear as follows:

```
+----------------------------------------+
|   MPLS interface; ifType = mpls(166)   |
+----------------------------------------+
|            Underlying Layer            |
+----------------------------------------+
```

In the above diagram, "Underlying Layer" refers to the ifIndex of any interface type for which MPLS interworking has been defined. Examples include ATM, Frame Relay, Ethernet, etc.

### 8.1. Support of the MPLS Layer by ifTable

Some specific interpretations of ifTable for the MPLS layer follow.

<table>
<thead>
<tr>
<th>Object</th>
<th>Use for the MPLS layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifIndex</td>
<td>Each MPLS interface is represented by an ifEntry.</td>
</tr>
<tr>
<td>ifDescr</td>
<td>Description of the MPLS interface.</td>
</tr>
<tr>
<td>ifType</td>
<td>The value that is allocated for MPLS is 166.</td>
</tr>
<tr>
<td>ifSpeed</td>
<td>The total bandwidth in bits per second for use by the MPLS layer.</td>
</tr>
<tr>
<td></td>
<td>ifPhysAddress Unused.</td>
</tr>
<tr>
<td>ifAdminStatus</td>
<td>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. Also see [RFC2863].</td>
</tr>
<tr>
<td>ifOperStatus</td>
<td>This value reflects the actual operational status of MPLS on this interface.</td>
</tr>
<tr>
<td></td>
<td>ifLastChange See [RFC2863].</td>
</tr>
<tr>
<td>ifInOctets</td>
<td>The number of received octets over the interface, i.e., the number of received, octets received as labeled packets.</td>
</tr>
<tr>
<td>ifOutOctets</td>
<td>The number of transmitted octets over the interface, i.e., the number of octets transmitted as labeled packets.</td>
</tr>
</tbody>
</table>
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 [RFC2863].

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 [RFC2863].

ifHCInOctets  The 64-bit version of ifInOctets; supported if required by the compliance statements in [RFC2863].

ifHCOutOctets The 64-bit version of ifOutOctets; supported if required by the compliance statements in [RFC2863].

ifAlias       The non-volatile ‘alias’ name for the interface as specified by a network manager.

ifCounterDiscontinuityTime
See [RFC2863].

9.  The Use of RowPointer

RowPointer is a textual convention used to identify a conceptual row in a MIB Table by pointing to the first accessible object in that row. In this MIB module, the trafficParamPtr object from either the mplsInSegmentTable or mplsOutSegmentTable SHOULD indicate the first accessible column in an entry in the MplsTunnelResourceEntry in the MPLS-TE-STD-MIB [RFC3812] to indicate the traffic parameter settings for this segment, if it represents an LSP used for a TE tunnel.

The trafficParamPtr object may optionally point at an externally defined traffic parameter specification table. A value of zeroDotZero indicates best-effort treatment. By having the same value of this object, two or more segments can indicate resource sharing of such things as LSP queue space, etc.
10. MPLS Label Switching Router MIB Module Definitions

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

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
  Integer32, Counter32, Unsigned32, Counter64, Gauge32,
  zeroDotZero
  FROM SNMPv2-SMI                                    -- [RFC2578]
  MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
  FROM SNMPv2-CONF                                   -- [RFC2580]
  TruthValue, RowStatus, StorageType, RowPointer,
  TimeStamp, TEXTUAL-CONVENTION
  FROM SNMPv2-TC                                     -- [RFC2579]
  InterfaceIndexOrZero, ifGeneralInformationGroup,
  ifCounterDiscontinuityGroup
  FROM IF-MIB                                        -- [RFC2863]
  mplsStdMIB, MplsLSPID, MplsLabel, MplsBitRate,
  MplsOwner
  FROM MPLS-TC-STD-MIB                               -- [RFC3811]
  AddressFamilyNumbers
  FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB               -- [IANAFamily]
  InetAddress,_INET_ADDRESS_TYPE
  FROM INET-ADDRESS-MIB                              -- [RFC3291]
 ;

mplsLsrStdMIB MODULE-IDENTITY
LAST-UPDATED "200406030000Z"  -- June 3, 2004
ORGANIZATION "Multiprotocol Label Switching (MPLS) Working Group"
CONTACT-INFO
"        Cheenu Srinivasan
        Bloomberg L.P.
        Email:  cheenu@bloomberg.net

        Arun Viswanathan
        Force10 Networks, Inc.
        Email:  arunv@force10networks.com

        Thomas D. Nadeau
        Cisco Systems, Inc.
        Email:  tnadeau@cisco.com

Comments about this document should be emailed
directly to the MPLS working group mailing list at
mpls@uu.net."

DESCRIPTION
"This MIB module contains managed object definitions for
the Multiprotocol Label Switching (MPLS) Router as

Copyright (C) The Internet Society (2004). The initial version of this MIB module was published in RFC 3812. For full legal notices see the RFC itself or see: http://www.ietf.org/copyrights/ianamib.html

-- Revision history.
REVISION
"200406030000Z" -- June 3, 2004
DESCRIPTION
"Initial revision, published as part of RFC 3813."

::= { mplsStdMIB 2 }

-- TEXTUAL-CONVENTIONS

MplsIndexType ::= TEXTUAL-CONVENTION
STATUS              current
DESCRIPTION
"This is an octet string that can be used as a table index in cases where a large addressable space is required such as on an LSR where many applications may be provisioning labels.

Note that the string containing the single octet with the value 0x00 is a reserved value used to represent special cases. When this TEXTUAL-CONVENTION is used as the SYNTAX of an object, the DESCRIPTION clause MUST specify if this special value is valid and if so what the special meaning is.

In systems that provide write access to the MPLS-LSR-STD MIB, mplsIndexType SHOULD be used as a simple multi-digit integer encoded as an octet string.
No further overloading of the meaning of an index SHOULD be made.

In systems that do not offer write access to the MPLS-LSR-STD MIB, the mplsIndexType may contain implicit formatting that is specific to the implementation to convey additional information such as interface index, physical card or device, or application id. The interpretation of this additional formatting is implementation dependent and not covered in this document. Such formatting MUST
NOT impact the basic functionality of read-only access to the MPLS-LSR-STD MIB by management applications that are not aware of the formatting rules.

SYNTAX OCTET STRING (SIZE(1..24))

MplsIndexNextType ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
"When a MIB module is used for configuration, an object with this SYNTAX always contains a legal value (a non-zero-length string) for an index that is not currently used in the relevant table. The Command Generator (Network Management Application) reads this variable and uses the (non-zero-length string) value read when creating a new row with an SNMP SET.

When the SET is performed, the Command Responder (agent) must determine whether the value is indeed still unused; Two Network Management Applications may attempt to create a row (configuration entry) simultaneously and use the same value. If it is currently unused, the SET succeeds and the Command Responder (agent) changes the value of this object, according to an implementation-specific algorithm. If the value is in use, however, the SET fails. The Network Management Application must then re-read this variable to obtain a new usable value.

Note that the string containing the single octet with the value 0x00 is a reserved value used to represent the special case where no additional indexes can be provisioned, or in systems that do not offer write access, objects defined using this TEXTUAL-CONVENTION MUST return the string containing the single octet with the value 0x00."

SYNTAX OCTET STRING (SIZE(1..24))

-- Top level components of this MIB module.

-- Notifications
mplsLsrNotifications OBJECT IDENTIFIER ::= { mplsLsrStdMIB 0 }

-- Tables, Scalars
mplsLsrObjects OBJECT IDENTIFIER ::= { mplsLsrStdMIB 1 }

-- Conformance
mplsLsrConformance OBJECT IDENTIFIER ::= { mplsLsrStdMIB 2 }

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

mplsInterfaceEntry OBJECT-TYPE
SYNTAX        MplsInterfaceEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION   "A conceptual row in this table is created
               automatically by an LSR for every interface capable
               of supporting MPLS and which is configured to do so.
               A conceptual row in this table will exist if and only if
               a corresponding entry in ifTable exists with ifType =
               mpls(166). If this associated entry in ifTable is
               operationally disabled (thus removing MPLS
               capabilities on that interface), the corresponding
               entry in this table MUST be deleted shortly thereafter.
               An conceptual row with index 0 is created if the LSR
               supports per-platform labels. This conceptual row
               represents the per-platform label space and contains
               parameters that apply to all interfaces that participate
               in the per-platform label space. Other conceptual rows
               in this table represent MPLS interfaces that may
               participate in either the per-platform or per-
               interface label spaces, or both. Implementations
               that either only support per-platform labels,
               or have only them configured, may choose to return
               just the mplsInterfaceEntry of 0 and not return
               the other rows. This will greatly reduce the number
               of objects returned. Further information about label
               space participation of an interface is provided in
               the DESCRIPTION clause of
               mplsInterfaceLabelParticipationType."
INDEX { mplsInterfaceLabelParticipationType }
::= { mplsInterfaceTable 1 }

MplsInterfaceEntry ::= SEQUENCE {
  mplsInterfaceIndex                  InterfaceIndexOrZero,
  mplsInterfaceLabelMinIn             MplsLabel,
  mplsInterfaceLabelMaxIn             MplsLabel,
  mplsInterfaceLabelMinOut            MplsLabel,
  mplsInterfaceLabelMaxOut            MplsLabel,
  mplsInterfaceTotalBandwidth         MplsBitRate,

Srinivasan, et al. Standards Track [Page 14]
mplsInterfaceAvailableBandwidth MplsBitRate,
mplsInterfaceLabelParticipationType BITS
}

mplsInterfaceIndex OBJECT-TYPE
SYNTAX InterfaceIndexOrZero
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This is a unique index for an entry in the
MplsInterfaceTable. A non-zero index for an
entry indicates the ifIndex for the corresponding
interface entry of the MPLS-layer in the ifTable.
The entry with index 0 represents the per-platform
label space and contains parameters that apply to all
interfaces that participate in the per-platform label
space. Other entries defined in this table represent
additional MPLS interfaces that may participate in either
the per-platform or per-interface label spaces, or both."
REFERENCE "RFC 2863 - The Interfaces Group MIB, McCloghrie, K.,
and F. Kastenholtz, June 2000"
 ::= { mplsInterfaceEntry 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."
 ::= { mplsInterfaceEntry 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."
 ::= { mplsInterfaceEntry 3 }

mplsInterfaceLabelMinOut OBJECT-TYPE
SYNTAX MplsLabel
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This is the minimum value of an MPLS label that this
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."
::= { mplsInterfaceEntry 4 }

mplsInterfaceTotalBandwidth OBJECT-TYPE
SYNTAX        MplsBitRate
UNITS         "kilobits per second"
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). This variable is not
applicable when applied to the interface with index
0. When this value cannot be measured, this value
should contain the nominal bandwidth."
::= { mplsInterfaceEntry 5 }

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). 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. When this value cannot be measured, this value
should contain the nominal bandwidth."
::= { mplsInterfaceEntry 6 }

mplsInterfaceLabelParticipationType OBJECT-TYPE
SYNTAX  BITS {
             perPlatform (0),
             perInterface (1)
         }
MAX-ACCESS    read-only
If the value of the mplsInterfaceIndex for this entry is zero, then this entry corresponds to the per-platform label space for all interfaces configured to use that label space. In this case the perPlatform(0) bit MUST be set; the perInterface(1) bit is meaningless and MUST be ignored.

The remainder of this description applies to entries with a non-zero value of mplsInterfaceIndex.

If the perInterface(1) bit is set then the value of mplsInterfaceLabelMinIn, mplsInterfaceLabelMaxIn, mplsInterfaceLabelMinOut, and mplsInterfaceLabelMaxOut for this entry reflect the label ranges for this interface.

If only the perPlatform(0) bit is set, then the value of mplsInterfaceLabelMinIn, mplsInterfaceLabelMaxIn, mplsInterfaceLabelMinOut, and mplsInterfaceLabelMaxOut for this entry MUST be identical to the instance of these objects with index 0. These objects may only vary from the entry with index 0 if both the perPlatform(0) and perInterface(1) bits are set.

In all cases, at a minimum one of the perPlatform(0) or perInterface(1) bits MUST be set to indicate that at least one label space is in use by this interface. In all cases, agents MUST ensure that label ranges are specified consistently and MUST return an inconsistentValue error when they do not.

REFERENCE

 ::= { mplsInterfaceEntry 8 }

-- End of mplsInterfaceTable

-- MPLS Interface Performance Table.

mplsInterfacePerfTable OBJECT-TYPE
SYNTAX SEQUENCE OF MplsInterfacePerfEntry
MAX-ACCESS not-accessible
STATUS current
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. It is an extension to the mplsInterfaceEntry table. Note that the discontinuity behavior of entries in this table MUST be based on the corresponding ifEntry's ifDiscontinuityTime."
AUGMENTS { mplsInterfaceEntry }
 ::= { mplsInterfacePerfTable 1 }

MplsInterfacePerfEntry ::= SEQUENCE {
  -- incoming direction
  mplsInterfacePerfInLabelsInUse Gauge32,
  mplsInterfacePerfInLabelLookupFailures Counter32,

  -- outgoing direction
  mplsInterfacePerfOutLabelsInUse Gauge32,
  mplsInterfacePerfOutFragmentedPkts Counter32
}

mplsInterfacePerfInLabelsInUse OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object counts the number of labels that are in use at this point in time on this interface in the incoming direction. If the interface participates in only the per-platform label space, then the value of the instance of this object MUST be identical to the value of the instance with index 0. If the interface participates in the per-interface label space, then the instance of this object MUST represent the number of per-interface labels that are in use on this interface."
 ::= { mplsInterfacePerfEntry 1 }

mplsInterfacePerfInLabelLookupFailures OBJECT-TYPE
SYNTAX Counter32

MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
 "This object counts the number of labeled packets
 that have been received on this interface and which
 were discarded because there was no matching cross-
 connect entry. This object MUST count on a per-
 interface basis regardless of which label space the
 interface participates in."
 ::= { mplsInterfacePerfEntry 2 }

mplsInterfacePerfOutLabelsInUse OBJECT-TYPE
SYNTAX        Gauge32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
 "This object counts the number of top-most labels in
 the outgoing label stacks that are in use at this
 point in time on this interface. This object MUST
 count on a per-interface basis regardless of which
 label space the interface participates in."
 ::= { mplsInterfacePerfEntry 3 }

mplsInterfacePerfOutFragmentedPkts OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
 "This object counts the number of outgoing MPLS
 packets that required fragmentation before
 transmission on this interface. This object MUST
 count on a per-interface basis regardless of which
 label space the interface participates in."
 ::= { mplsInterfacePerfEntry 4 }

-- mplsInterfacePerf Table end.

mplsInSegmentIndexNext OBJECT-TYPE
SYNTAX        MplsIndexNextType
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
 "This object contains the next available value to
 be used for mplsInSegmentIndex when creating entries
 in the mplsInSegmentTable. The special value of a
 string containing the single octet 0x00 indicates
 that no new entries can be created in this table.
 Agents not allowing managers to create entries
in this table MUST set this object to this special value.
::= { mplsLsrObjects 3 }

-- in-segment table.
mplsInSegmentTable  OBJECT-TYPE
SYNTAX        SEQUENCE OF MplsInSegmentEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION  "This table contains a description of the incoming MPLS segments (labels) to an LSR and their associated parameters. The index for this table is mplsInSegmentIndex. The index structure of this table is specifically designed to handle many different MPLS implementations that manage their labels both in a distributed and centralized manner. The table is also designed to handle existing MPLS labels as defined in RFC3031 as well as longer ones that may be necessary in the future.

In cases where the label cannot fit into the mplsInSegmentLabel object, the mplsInSegmentLabelPtr will indicate this by being set to the first accessible column in the appropriate extension table’s row. In this case an additional table MUST be provided and MUST be indexed by at least the indexes used by this table. In all other cases when the label is represented within the mplsInSegmentLabel object, the mplsInSegmentLabelPtr MUST be set to 0.0. Due to the fact that MPLS labels may not exceed 24 bits, the mplsInSegmentLabelPtr object is only a provision for future-proofing the MIB module. Thus, the definition of any extension tables is beyond the scope of this MIB module."
::= { mplsLsrObjects 4 }

mplsInSegmentEntry  OBJECT-TYPE
SYNTAX        MplsInSegmentEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION  "An entry in this table represents one incoming segment as is represented in an LSR’s LFIB. 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."
The value of mplsInSegmentRowStatus cannot be active(1) unless the ifTable entry corresponding to mplsInSegmentInterface exists. An entry in this table must match any incoming packets, and indicates an instance of mplsXCEntry based on which forwarding and/or switching actions are taken.

INDEX { mplsInSegmentIndex }
::= { mplsInSegmentTable 1 }

MplsInSegmentEntry ::= SEQUENCE {
    mplsInSegmentIndex                MplsIndexType,
    mplsInSegmentInterface            InterfaceIndexOrZero,
    mplsInSegmentLabel                MplsLabel,
    mplsInSegmentLabelPtr             RowPointer,
    mplsInSegmentNPop                 Integer32,
    mplsInSegmentAddrFamily           AddressFamilyNumbers,
    mplsInSegmentXCIndex              MplsIndexType,
    mplsInSegmentOwner                MplsOwner ,
    mplsInSegmentTrafficParamPtr      RowPointer,
    mplsInSegmentRowStatus            RowStatus,
    mplsInSegmentStorageType          StorageType
}

mplsInSegmentIndex OBJECT-TYPE
SYNTAX        MplsIndexType
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"The index for this in-segment. The string containing the single octet 0x00 MUST not be used as an index."
::= { mplsInSegmentEntry 1 }

mplsInSegmentInterface OBJECT-TYPE
SYNTAX        InterfaceIndexOrZero
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"This object represents the interface index for the incoming MPLS interface. A value of zero represents all interfaces participating in the per-platform label space. This may only be used in cases where the incoming interface and label are associated with the same mplsXCEntry. Specifically, given a label and any incoming interface pair from the per-platform label space, the outgoing label/interface mapping remains the same. If this is not the case, then individual entries MUST exist that
can then be mapped to unique mplsXCEntries."
:= { mplsInSegmentEntry 2 }

mplsInSegmentLabel OBJECT-TYPE
SYNTAX        MplsLabel
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"If the corresponding instance of mplsInSegmentLabelPtr is
zeroDotZero then this object MUST contain the incoming label
associated with this in-segment. If not this object SHOULD
be zero and MUST be ignored."
:= { mplsInSegmentEntry 3 }

mplsInSegmentLabelPtr OBJECT-TYPE
SYNTAX        RowPointer
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"If the label for this segment cannot be represented
fully within the mplsInSegmentLabel object,
this object MUST point to the first accessible
column of a conceptual row in an external table containing
the label. In this case, the mplsInSegmentTopLabel
object SHOULD be set to 0 and ignored. This object MUST
be set to zeroDotZero otherwise."
DEFVAL { zeroDotZero }
:= { mplsInSegmentEntry 4 }

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. This is indicated by setting this
object to the default value of 1. If an LSR supports
popping of more than one label, this object MUST
be set to that number. This object cannot be modified
if mplsInSegmentRowStatus is active(1)."
DEFVAL        { 1 }
:= { mplsInSegmentEntry 5 }

mplsInSegmentAddrFamily OBJECT-TYPE
SYNTAX        AddressFamilyNumbers
MAX-ACCESS    read-create
STATUS current
DESCRIPTION
"The IANA address family [IANAFamily] of packets received on this segment, which is used at an egress LSR to deliver them to the appropriate layer 3 entity. A value of other(0) indicates that the family type is either unknown or undefined; this SHOULD NOT be used at an egress LSR. This object cannot be modified if mplsInSegmentRowStatus is active(1)."
REFERENCE
"Internet Assigned Numbers Authority (IANA), ADDRESS FAMILY NUMBERS, (http://www.iana.org/assignments/address-family-numbers), for MIB see: http://www.iana.org/assignments/ianaaddressfamilynumbers-mib"

DEFVAL { other }
::= { mplsInSegmentEntry 6 }

mplsInSegmentXCIndex OBJECT-TYPE
SYNTAX MplsIndexType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Index into mplsXCTable which identifies which cross-connect entry this segment is part of. The string containing the single octet 0x00 indicates that this entry is not referred to by any cross-connect entry. When a cross-connect entry is created which this in-segment is a part of, this object is automatically updated to reflect the value of mplsXCIndex of that cross-connect entry."
::= { mplsInSegmentEntry 7 }

mplsInSegmentOwner OBJECT-TYPE
SYNTAX MplsOwner
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Denotes the entity that created and is responsible for managing this segment."
::= { mplsInSegmentEntry 8 }

mplsInSegmentTrafficParamPtr OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This variable represents a pointer to the traffic parameter specification for this in-segment. This value may point at an entry in the mplsTunnelResourceTable in the MPLS-TE-STD-MIB (RFC3812) to indicate which traffic parameter settings for this segment if it represents an LSP used for a TE tunnel.

This value may optionally point at an externally defined traffic parameter specification table. A value of zeroDotZero indicates best-effort treatment. By having the same value of this object, two or more segments can indicate resource sharing of such things as LSP queue space, etc.

This object cannot be modified if mplsInSegmentRowStatus is active(1). For entries in this table that are preserved after a re-boot, the agent MUST ensure that their integrity be preserved, or this object should be set to 0.0 if it cannot.

DEFVAL { zeroDotZero }
::= { 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. When a row in this table has a row in the active(1) state, no objects in this row can be modified except the mplsInSegmentRowStatus and mplsInSegmentStorageType."
::= { mplsInSegmentEntry 10 }

mplsInSegmentStorageType OBJECT-TYPE
SYNTAX        StorageType
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"This variable indicates the storage type for this object. The agent MUST ensure that this object’s value remains consistent with the associated mplsXCEntry. Conceptual rows having the value 'permanent’ need not allow write-access to any columnar objects in the row."

REFERENCE
"See RFC2579."
DEFVAL { volatile }
::= { mplsInSegmentEntry 11 }

-- 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 5 }

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 is configured in the mplsInSegmentTable. The counters in this entry should behave in a manner similar to that of the interface. mplsInSegmentPerfDiscontinuityTime indicates the time of the last discontinuity in all of these objects."
AUGMENTS  { mplsInSegmentEntry }
::= { mplsInSegmentPerfTable 1 }

MplsInSegmentPerfEntry ::= SEQUENCE {
      mplsInSegmentPerfOctets          Counter32,
      mplsInSegmentPerfPackets         Counter32,
      mplsInSegmentPerfErrors          Counter32,
      mplsInSegmentPerfDiscards        Counter32,

      -- high capacity counter
      mplsInSegmentPerfHCOctets        Counter64,

      mplsInSegmentPerfDiscontinuityTime TimeStamp
    }

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

"This value represents the total number of octets received by this segment. It MUST be equal to the least significant 32 bits of mplsInSegmentPerfHCOctets if mplsInSegmentPerfHCOctets is supported according to the rules spelled out in RFC2863."

::= { mplsInSegmentPerfEntry 1 }

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

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

mplsInSegmentPerfDiscards 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 }

mplsInSegmentPerfHCOctets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The total number of octets received. This is the 64 bit version of mplsInSegmentPerfOctets, if mplsInSegmentPerfHCOctets is supported according to the rules spelled out in RFC2863."
::= { mplsInSegmentPerfEntry 5 }
mplsInSegmentPerfDiscontinuityTime OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value of sysUpTime on the most recent occasion
at which any one or more of this segment's Counter32
or Counter64 suffered a discontinuity. If no such
discontinuities have occurred since the last re-
initialization of the local management subsystem,
then this object contains a zero value."
::= { mplsInSegmentPerfEntry 6 }

-- End of mplsInSegmentPerfTable.

-- out-segment table.

mplsOutSegmentIndexNext OBJECT-TYPE
SYNTAX        MplsIndexNextType
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
"This object contains the next available value to
be used for mplsOutSegmentIndex when creating entries
in the mplsOutSegmentTable. The special value of a
string containing the single octet 0x00
indicates that no new entries can be created in this
table. Agents not allowing managers to create entries
in this table MUST set this object to this special
value."
::= { mplsLsrObjects 6 }

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 7 }

mplsOutSegmentEntry  OBJECT-TYPE
SYNTAX        MplsOutSegmentEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"An entry in this table represents one outgoing
segment. An entry can be created by a network administrator, an SNMP agent, or an MPLS signaling protocol. The object mplsOutSegmentOwner indicates the creator of this entry. The value of mplsOutSegmentRowStatus cannot be active(1) unless the ifTable entry corresponding to mplsOutSegmentInterface exists.

Note that the indexing of this table uses a single, arbitrary index (mplsOutSegmentIndex) to indicate which out-segment (i.e.: label) is being switched to from which in-segment (i.e: label) or in-segments. This is necessary because it is possible to have an equal-cost multi-path situation where two identical out-going labels are assigned to the same cross-connect (i.e.: they go to two different neighboring LSRs); thus, requiring two out-segments. In order to preserve the uniqueness of the references by the mplsXCTable, an arbitrary integer must be used as the index for this table."

INDEX { mplsOutSegmentIndex }
 ::= { mplsOutSegmentTable 1 }

MplsOutSegmentEntry ::= SEQUENCE {
    mplsOutSegmentIndex                MplsIndexType,
    mplsOutSegmentInterface            InterfaceIndexOrZero,
    mplsOutSegmentPushTopLabel         TruthValue,
    mplsOutSegmentTopLabel             MplsLabel,
    mplsOutSegmentTopLabelPtr          RowPointer,
    mplsOutSegmentNextHopAddrType      InetAddressType,
    mplsOutSegmentNextHopAddr          InetAddress,
    mplsOutSegmentXCIndex              MplsIndexType,
    mplsOutSegmentOwner                MplsOwner,
    mplsOutSegmentTrafficParamPtr      RowPointer,
    mplsOutSegmentRowStatus            RowStatus,
    mplsOutSegmentStorageType          StorageType
}

mplsOutSegmentIndex OBJECT-TYPE
 SYNTAX        MplsIndexType
 MAX-ACCESS    not-accessible
 STATUS        current
 DESCRIPTION
  "This value contains a unique index for this row. While a value of a string containing the single octet 0x00 is not valid as an index for entries in this table, it can be supplied as a valid value to index the mplsXCTable to represent entries for"
which no out-segment has been configured or exists."
::= { mplsOutSegmentEntry 1 }

mplsOutSegmentInterface OBJECT-TYPE
SYNTAX InterfaceIndexOrZero
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This value must contain the interface index of the outgoing interface. This object cannot be modified if mplsOutSegmentRowStatus is active(1). The mplsOutSegmentRowStatus cannot be set to active(1) until this object is set to a value corresponding to a valid ifEntry."
::= { 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(1) if the outgoing interface does not support pop-and-go (and no label stack remains). For example, on ATM interface, or if the segment represents 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 MUST ensure that this situation does not happen. This object cannot be modified if mplsOutSegmentRowStatus is active(1)."
DEFVAL { true }
::= { mplsOutSegmentEntry 3 }

mplsOutSegmentTopLabel OBJECT-TYPE
SYNTAX MplsLabel
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"If mplsOutSegmentPushTopLabel is true then this represents the label that should be pushed onto the top of the outgoing packet’s label stack. Otherwise this value SHOULD be set to 0 by the management station and MUST be ignored by the agent. This
mplsOutSegmentEntry

mplsOutSegmentRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the status of the row.  If the
entry is not currently active, the mplsOutSegmentEntry
cannot be modified.

DEFVAL { inactive }
::= { mplsOutSegmentEntry 1 }

mplsOutSegmentLabel OBJECT-TYPE
SYNTAX Label
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"If the label for this segment cannot be represented
fully within the mplsOutSegmentLabel object, this object
MUST point to the first accessible column of a
conceptual row in an external table containing the
label.  In this case, the mplsOutSegmentRowStatus
object SHOULD be set to active(1) and ignored.  This object
MUST be set to unknown otherwise."
DEFVAL { 0 }
::= { mplsOutSegmentEntry 2 }

mplsOutSegmentTopLabelPtr OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"If the label for this segment cannot be represented
fully within the mplsOutSegmentLabel object, this
object MUST point to the first accessible column of a
conceptual row in an external table containing the
label.  In this case, the mplsOutSegmentTopLabel
object SHOULD be set to 0 and ignored.  This object
MUST be set to zeroDotZero otherwise."
DEFVAL { zeroDotZero }
::= { mplsOutSegmentEntry 3 }

mplsOutSegmentNextHopAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Indicates the next hop Internet address type.
Only values unknown(0), ipv4(1) or ipv6(2)
have to be supported.

A value of unknown(0) is allowed only when
the outgoing interface is of type point-to-point.
If any other unsupported values are attempted in a set
operation, the agent MUST return an inconsistentValue
error."
REFERENCE
"See RFC3291."
::= { mplsOutSegmentEntry 4 }

mplsOutSegmentNextHopAddr OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The internet address of the next hop. The type of
this address is determined by the value of the
mplsOutSegmentNextHopAddrType object.

This object cannot be modified if
mplsOutSegmentRowStatus is active(1)."
::= { mplsOutSegmentEntry 7 }

mplsOutSegmentXCIndex OBJECT-TYPE
SYNTAX MplsIndexType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Index into mplsXCTable which identifies which cross-connect entry this segment is part of. A value of the string containing the single octet 0x00 indicates that this entry is not referred to by any cross-connect entry. When a cross-connect entry is created which this out-segment is a part of, this object MUST be updated by the agent to reflect the value of mplsXCIndex of that cross-connect entry."
 ::= { mplsOutSegmentEntry 8 }

mplsOutSegmentOwner OBJECT-TYPE
SYNTAX MplsOwner
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Denotes the entity which created and is responsible for managing this segment."
 ::= { mplsOutSegmentEntry 9 }

mplsOutSegmentTrafficParamPtr OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This variable represents a pointer to the traffic parameter specification for this out-segment. This value may point at an entry in the MplsTunnelResourceEntry in the MPLS-TE-STD-MIB (RFC3812)

RFC Editor: Please fill in RFC number.

to indicate which traffic parameter settings for this segment if it represents an LSP used for a TE tunnel.

This value may optionally point at an externally defined traffic parameter specification table. A value of zeroDotZero indicates best-effort treatment. By having the same value of this object, two or more segments can indicate resource sharing
of such things as LSP queue space, etc.

This object cannot be modified if mplsOutSegmentRowStatus is active(1).
For entries in this table that are preserved after a re-boot, the agent MUST ensure that their integrity be preserved, or this object should be set to 0.0 if it cannot.

DEFVAL { zeroDotZero }
 ::= { mplsOutSegmentEntry 10 }

mplsOutSegmentRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"For creating, modifying, and deleting this row. When a row in this table has a row in the active(1) state, no objects in this row can be modified except the mplsOutSegmentRowStatus or mplsOutSegmentStorageType."

::= { mplsOutSegmentEntry 11 }

mplsOutSegmentStorageType OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This variable indicates the storage type for this object. The agent MUST ensure that this object’s value remains consistent with the associated mplsXCEntry. Conceptual rows having the value ‘permanent’ need not allow write-access to any columnar objects in the row."

DEFVAL { volatile }
 ::= { mplsOutSegmentEntry 12 }

-- 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

Srinivasan, et al. Standards Track [Page 32]
outgoing segments from an LSR. The counters in this
entry should behave in a manner similar to that of
the interface."
 ::= { mplsLsrObjects 8 }

mplsOutSegmentPerfEntry OBJECT-TYPE
SYNTAX        MplsOutSegmentPerfEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
 "An entry in this table contains statistical
information about one outgoing segment configured in
mplsOutSegmentTable. The object
mplsOutSegmentPerfDiscontinuityTime indicates the
time of the last discontinuity in these objects."
AUGMENTS      { mplsOutSegmentEntry }
 ::= { mplsOutSegmentPerfTable 1 }

MplsOutSegmentPerfEntry ::= SEQUENCE {
    mplsOutSegmentPerfOctets             Counter32,
    mplsOutSegmentPerfPackets            Counter32,
    mplsOutSegmentPerfErrors             Counter32,
    mplsOutSegmentPerfDiscards           Counter32,
    -- HC counter
    mplsOutSegmentPerfHCOctets           Counter64,
    mplsOutSegmentPerfDiscontinuityTime  TimeStamp
}

mplsOutSegmentPerfOctets OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
 "This value contains the total number of octets sent
on this segment. It MUST be equal to the least
significant 32 bits of mplsOutSegmentPerfHCOctets
if mplsOutSegmentPerfHCOctets is supported according to
the rules spelled out in RFC2863." :
 ::= { mplsOutSegmentPerfEntry 1 }

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

mplsOutSegmentPerfErrors 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 }

mplsOutSegmentPerfDiscards OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of labeled packets attempted to be transmitted 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 }

mplsOutSegmentPerfHCOctets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Total number of octets sent. This is the 64 bit version of mplsOutSegmentPerfOctets, if mplsOutSegmentPerfHCOctets is supported according to the rules spelled out in RFC2863."
::= { mplsOutSegmentPerfEntry 5 }

mplsOutSegmentPerfDiscontinuityTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value of sysUpTime on the most recent occasion at which any one or more of this segment’s Counter32 or Counter64 suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."
::= { mplsOutSegmentPerfEntry 6 }
-- End of mplsOutSegmentPerfTable.

-- Cross-connect table.

mplsXCIndexNext OBJECT-TYPE
SYNTAX      MplsIndexNextType
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "This object contains the next available value to be used for mplsXCIndex when creating entries in the mplsXCTable. A special value of the zero length string indicates that no more new entries can be created in the relevant table. Agents not allowing managers to create entries in this table MUST set this value to the zero length string."
 ::= { mplsLsrObjects 9 }

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 10 }

mplsXCEntry OBJECT-TYPE
SYNTAX      MplsXCEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
 "A row in this table represents one cross-connect entry. It is indexed by the following objects:

 - cross-connect index mplsXCIndex that uniquely identifies a group of cross-connect entries

 - in-segment index, mplsXCInSegmentIndex

 - out-segment index, mplsXCOutSegmentIndex
LSPs originating at this LSR:
These are represented by using the special value of mplsXCInSegmentIndex set to the string containing a single octet 0x00. In this case the mplsXCOutSegmentIndex MUST not be the string containing a single octet 0x00.

LSPs terminating at this LSR:
These are represented by using the special value mplsXCOutSegmentIndex set to the string containing a single octet 0x00.

Special labels:
Entries indexed by the strings containing the reserved MPLS label values as a single octet 0x00 through 0x0f (inclusive) imply LSPs terminating at this LSR. Note that situations where LSPs are terminated with incoming label equal to the string containing a single octet 0x00 can be distinguished from LSPs originating at this LSR because the mplsXCOutSegmentIndex equals the string containing the single octet 0x00.

An entry can be created by a network administrator or by an SNMP agent as instructed by an MPLS signaling protocol.

INDEX { mplsXCIndex, mplsXCInSegmentIndex, mplsXCOutSegmentIndex }
 ::= { mplsXCTable 1 }

MplsXCEntry ::= SEQUENCE {
  mplsXCIndex                  MplsIndexType,
  mplsXCInSegmentIndex         MplsIndexType,
  mplsXCOutSegmentIndex        MplsIndexType,
  mplsXCLabelStackIndex        MplsIndexType,
  mplsXCOwner                  MplsOwner,
  mplsXCRowStatus              RowStatus,
  mplsXCStorageType            StorageType,
  mplsXCAdminStatus            INTEGER,
  mplsXCOperStatus             INTEGER
}

mplsXCIndex OBJECT-TYPE
SYNTAX        MplsIndexType
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"Primary index for the conceptual row identifying a
 group of cross-connect segments. The string
 containing a single octet 0x00 is an invalid index."
::= { mplsXCEntry 1 }

mplsXCInSegmentIndex OBJECT-TYPE
SYNTAX        MplsIndexType
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"Incoming label index. If this object is set to the string containing
 a single octet 0x00, this indicates a special case outlined in the table’s description above.
 In this case no corresponding mplsInSegmentEntry shall exist."
::= { mplsXCEntry 2 }

mplsXCOutSegmentIndex OBJECT-TYPE
SYNTAX        MplsIndexType
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"Index of out-segment for LSPs not terminating on
 this LSR if not set to the string containing the single octet 0x00. If the segment identified by this entry is terminating, then this object MUST be set to the string containing a single octet 0x00 to indicate that no corresponding mplsOutSegmentEntry shall exist."
::= { mplsXCEntry 3 }

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. This object cannot be modified if mplsXCRowStatus is active(1) except for this object."
::= { mplsXCEntry 4 }

mplsXCLabelStackIndex OBJECT-TYPE
SYNTAX        MplsIndexType
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 the string containing the
single octet 0x00 indicates that no labels are to
be stacked beneath the top label.
This object cannot be modified if mplsXCRowStatus is
active(1)."
 ::= { mplsXEntry 5 }

mplsXOwner OBJECT-TYPE
SYNTAX    MplsOwner
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
"Denotes the entity that created and is responsible
for managing this cross-connect."
 ::= { mplsXEntry 6 }

mplsXCRowStatus OBJECT-TYPE
SYNTAX        RowStatus
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"For creating, modifying, and deleting this row.
When a row in this table has a row in the active(1)
state, no objects in this row except this object
and the mplsXCStorageType can be modified."
 ::= { mplsXEntry 7 }

mplsXCStorageType OBJECT-TYPE
SYNTAX        StorageType
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"This variable indicates the storage type for this
object. The agent MUST ensure that the associated in
and out segments also have the same StorageType value
and are restored consistently upon system restart.
This value SHOULD be set to permanent(4) if created
as a result of a static LSP configuration.

Conceptual rows having the value ‘permanent’
need not allow write-access to any columnar
objects in the row."
DEFVAL { volatile } ::= { mplsXCEntry 8 }

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."
DEFVAL { up }
 ::= { mplsXCEntry 9 }

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 10 }

-- End of mplsXCTable

-- Label stack table.

mplsMaxLabelStackDepth OBJECT-TYPE
SYNTAX   Unsigned32 (1..2147483647)
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "The maximum stack depth supported by this LSR."
 ::= { mplsLsrObjects 11 }
mplsLabelStackIndexNext  OBJECT-TYPE
SYNTAX        MplsIndexNextType
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "This object contains the next available value to
be used for mplsLabelStackIndex when creating entries
in the mplsLabelStackTable. The special string
containing the single octet 0x00
indicates that no more new entries can be created
in the relevant table. Agents not allowing managers
to create entries in this table MUST set this value
to the string containing the single octet 0x00."
 ::= { mplsLsrObjects 12 }

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 13 }

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             MplsIndexType,
   mplsLabelStackLabelIndex        Unsigned32,
   mplsLabelStackLabel             MplsLabel,
   mplsLabelStackLabelPtr          RowPointer,
   mplsLabelStackRowStatus         RowStatus,
   mplsLabelStackStorageType       StorageType
}

mplsLabelStackIndex OBJECT-TYPE
SYNTAX     MplsIndexType
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"Primary index for this row identifying a stack of
labels to be pushed on an outgoing packet, beneath
the top label. An index containing the string with
a single octet 0x00 MUST not be used."
::= { mplsLabelStackEntry 1 }

mplsLabelStackLabelIndex OBJECT-TYPE
SYNTAX     Unsigned32 (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 }

mplsLabelStackLabelPtr OBJECT-TYPE
SYNTAX     RowPointer
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"If the label for this segment cannot be represented
fully within the mplsLabelStackLabel object,
this object MUST point to the first accessible
column of a conceptual row in an external table containing
the label. In this case, the mplsLabelStackLabel
object SHOULD be set to 0 and ignored. This object
MUST be set to zeroDotZero otherwise."
DEFVAL { zeroDotZero }
::= { mplsLabelStackEntry 4 }

mplsLabelStackRowStatus OBJECT-TYPE
SYNTAX          RowStatus  
MAX-ACCESS      read-create 
STATUS          current 
DESCRIPTION     "For creating, modifying, and deleting this row. 
When a row in this table has a row in the active(1) 
state, no objects in this row except this object 
and the mplsLabelStackStorageType can be modified."
 ::= { mplsLabelStackEntry 5 }

mplsLabelStackStorageType OBJECT-TYPE  
SYNTAX          StorageType  
MAX-ACCESS      read-create 
STATUS          current 
DESCRIPTION     "This variable indicates the storage type for this 
object. This object cannot be modified if 
mplsLabelStackRowStatus is active(1). 
No objects are required to be writable for 
rows in this table with this object set to 
permanent(4). 

The agent MUST ensure that all related entries 
in this table retain the same value for this 
object. Agents MUST ensure that the storage type 
for all entries related to a particular mplsXCEntry 
retain the same value for this object as the 
mplsXCEntry’s StorageType."
DEFVAL { volatile }
 ::= { mplsLabelStackEntry 6 }

-- End of mplsLabelStackTable

-- Begin mplsInSegmentMapTable

mplsInSegmentMapTable OBJECT-TYPE  
SYNTAX          SEQUENCE OF MplsInSegmentMapEntry 
MAX-ACCESS      not-accessible 
STATUS          current 
DESCRIPTION     "This table specifies the mapping from the 
mplsInSegmentIndex to the corresponding 
mplsInSegmentInterface and mplsInSegmentLabel 
objects. The purpose of this table is to 
provide the manager with an alternative 
means by which to locate in-segments."
 ::= { mplsLsrObjects 14 }
mplsInSegmentMapEntry OBJECT-TYPE
SYNTAX        MplsInSegmentMapEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"An entry in this table represents one interface
and incoming label pair.

In cases where the label cannot fit into the
mplsInSegmentLabel object, the mplsInSegmentLabelPtr
will indicate this by being set to the first accessible
column in the appropriate extension table’s row,
and the mplsInSegmentLabel SHOULD be set to 0.
In all other cases when the label is
represented within the mplsInSegmentLabel object, the
mplsInSegmentLabelPtr MUST be 0.0.

Implementors need to be aware that if the value of
the mplsInSegmentMapLabelPtrIndex (an OID) has more
that 111 sub-identifiers, then OIDs of column
instances in this table will have more than 128
sub-identifiers and cannot be accessed using SNMPv1,
SNMPv2c, or SNMPv3."
INDEX { mplsInSegmentMapInterface,
    mplsInSegmentMapLabel,
    mplsInSegmentMapLabelPtrIndex }
::= { mplsInSegmentMapTable 1 }

MplsInSegmentMapEntry ::= SEQUENCE {
    mplsInSegmentMapInterface      InterfaceIndexOrZero,
    mplsInSegmentMapLabel          MplsLabel,
    mplsInSegmentMapLabelPtrIndex  RowPointer,
    mplsInSegmentMapIndex          MplsIndexType
}

mplsInSegmentMapInterface OBJECT-TYPE
SYNTAX        InterfaceIndexOrZero
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"This index contains the same value as the
mplsInSegmentIndex in the mplsInSegmentTable."
::= { mplsInSegmentMapEntry 1 }

mplsInSegmentMapLabel OBJECT-TYPE
SYNTAX        MplsLabel
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
"This index contains the same value as the
mplsInSegmentLabel in the mplsInSegmentTable."
::= { mplsInSegmentMapEntry 2 }

mplsInSegmentMapLabelPtrIndex OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This index contains the same value as the
mplsInSegmentLabelPtr.

If the label for the InSegment cannot be represented
fully within the mplsInSegmentLabel object,
this index MUST point to the first accessible
column of a conceptual row in an external table containing
the label. In this case, the mplsInSegmentTopLabel
object SHOULD be set to 0 and ignored. This object MUST
be set to zeroDotZero otherwise."
::= { mplsInSegmentMapEntry 3 }

mplsInSegmentMapIndex OBJECT-TYPE
SYNTAX MplsIndexType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The mplsInSegmentIndex that corresponds
to the mplsInSegmentInterface and
mplsInSegmentLabel, or the mplsInSegmentInterface
and mplsInSegmentLabelPtr, if applicable.
The string containing the single octet 0x00
MUST not be returned."
::= { mplsInSegmentMapEntry 4 }

-- End mplsInSegmentMapTable

-- Notification Configuration

mplsXCNotificationsEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"If this object is set to true(1), then it enables
the emission of mplsXCUp and mplsXCDown
notifications; otherwise these notifications are not
emitted."
REFERENCE
   "See also RFC3413 for explanation that
   notifications are under the ultimate control of the
   MIB module in this document."
DEFVAL { false }
::= { mplsLsrObjects 15 }

-- Cross-connect.

mplsXCUp NOTIFICATION-TYPE
OBJECTS     { mplsXCOperStatus,  -- start of range
               mplsXCOperStatus   -- end of range
}
STATUS      current
DESCRIPTION
   "This notification is generated when the
   mplsXCOperStatus object for one or more contiguous
   entries in mplsXCTable are about to enter the up(1)
   state from some other state. The included values of
   mplsXCOperStatus MUST both be set equal to this
   new state (i.e: up(1)). The two instances of
   mplsXCOperStatus in this notification indicate the range
   of indexes that are affected. Note that all the indexes
   of the two ends of the range can be derived from the
   instance identifiers of these two objects. For
   cases where a contiguous range of cross-connects
   have transitioned into the up(1) state at roughly
   the same time, the device SHOULD issue a single
   notification for each range of contiguous indexes in
   an effort to minimize the emission of a large number
   of notifications. If a notification has to be
   issued for just a single cross-connect entry, then
   the instance identifier (and values) of the two
   mplsXCOperStatus objects MUST be the identical."
::= { mplsLsrNotifications 1 }

mplsXCDown NOTIFICATION-TYPE
OBJECTS     { mplsXCOperStatus,  -- start of range
               mplsXCOperStatus   -- end of range
}
STATUS      current
DESCRIPTION
   "This notification is generated when the
   mplsXCOperStatus object for one or more contiguous
   entries in mplsXCTable are about to enter the
down(2) state from some other state. The included values
of mplsXCOperStatus MUST both be set equal to this
down(2) state. The two instances of mplsXCOperStatus
in this notification indicate the range of indexes
that are affected. Note that all the indexes of the
two ends of the range can be derived from the
instance identifiers of these two objects. For
cases where a contiguous range of cross-connects
have transitioned into the down(2) state at roughly
the same time, the device SHOULD issue a single
notification for each range of contiguous indexes in
an effort to minimize the emission of a large number
of notifications. If a notification has to be
issued for just a single cross-connect entry, then
the instance identifier (and values) of the two
mplsXCOperStatus objects MUST be identical.

::= { mplsLsrNotifications 2 }

-- End of notifications.

-- Module compliance.

mplsLsrGroups
   OBJECT IDENTIFIER ::= { mplsLsrConformance 1 }

mplsLsrCompliances
   OBJECT IDENTIFIER ::= { mplsLsrConformance 2 }

-- Compliance requirement for fully compliant implementations.

mplsLsrModuleFullCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "Compliance statement for agents that provide full
support for MPLS-LSR-STD-MIB. Such devices can
then be monitored and also be configured using
this MIB module."

MODULE IF-MIB -- The Interfaces Group MIB, RFC 2863.
MANDATORY-GROUPS {
   ifGeneralInformationGroup,
   ifCounterDiscontinuityGroup
}

MODULE -- This module.
MANDATORY-GROUPS {
   mplsInterfaceGroup,
   mplsInSegmentGroup,
   mplsOutSegmentGroup,

Srinivasan, et al. Standards Track [Page 46]
mplsXCGroup,
mplsPerfGroup
}

GROUP    mplsLabelStackGroup
DESCRIPTION "This group is only mandatory for LSRs that wish to
support the modification of LSP label stacks.
"

GROUP    mplsHCInSegmentPerfGroup
DESCRIPTION "This group is mandatory for those in-segment entries
for which the object mplsInSegmentOutOctets wraps
around too quickly based on the criteria specified in
RFC 2863 for high-capacity counters.
"

GROUP    mplsHCOutSegmentPerfGroup
DESCRIPTION "This group is mandatory for those out-segment entries
for which the object mplsOutSegmentPerfOctets wraps
around too quickly based on the criteria specified in
RFC 2863 for high-capacity counters.
"

GROUP    mplsLsrNotificationGroup
DESCRIPTION "This group is only mandatory for those implementations
which can efficiently implement the notifications
contained in this group."

OBJECT    mplsInSegmentRowStatus
SYNTAX    RowStatus { active(1), notInService(2) }
WRITE-SYNTAX RowStatus { active(1), notInService(2),
createAndGo(4), destroy(6) }
DESCRIPTION "Support for createAndWait and notReady is
not required."

OBJECT    mplsOutSegmentNextHopAddrType
SYNTAX    InetAddressType { unknown(0), ipv4(1), ipv6(2) }
DESCRIPTION "Only unknown(0), ipv4(1) and ipv6(2) support
is required."

OBJECT    mplsOutSegmentNextHopAddr
SYNTAX    InetAddress (SIZE(0|4|16))
DESCRIPTION "An implementation is only required to support
unknown(0), ipv4(1) and ipv6(2) sizes."

OBJECT    mplsOutSegmentRowStatus
SYNTAX    RowStatus { active(1), notInService(2) }
WRITE-SYNTAX RowStatus { active(1), notInService(2),
createAndGo(4), destroy(6) }

DESCRIPTION "Support for createAndWait and notReady is not
required."

OBJECT       mplsLabelStackRowStatus
SYNTAX       RowStatus { active(1), notInService(2) }
WRITE-SYNTAX RowStatus { active(1), notInService(2),
createAndGo(4), destroy(6) }

DESCRIPTION "Support for createAndWait and notReady is not
required."

OBJECT       mplsXCRowStatus
SYNTAX       RowStatus { active(1), notInService(2) }
WRITE-SYNTAX RowStatus { active(1), notInService(2),
createAndGo(4), destroy(6) }

DESCRIPTION "Support for createAndWait and notReady is not
required."

::= { mplsLsrCompliances 1 }

-- Compliance requirement for read-only implementations.

mplsLsrModuleReadOnlyCompliance MODULE-COMPLIANCE
STATUS       current
DESCRIPTION "Compliance requirement for implementations that only
provide read-only support for MPLS-LSR-STD-MIB. Such
devices can then be monitored but cannot be configured
using this MIB module.
"

MODULE IF-MIB -- The interfaces Group MIB, RFC 2863
MANDATORY-GROUPS {
  ifGeneralInformationGroup,
  ifCounterDiscontinuityGroup
}
GROUP mplsLabelStackGroup
DESCRIPTION "This group is only mandatory for LSRs that wish to support the modification of LSP label stacks."

GROUP mplsHCInSegmentPerfGroup
DESCRIPTION "This group is mandatory for those in-segment entries for which the object mplsInSegmentOutOctets wraps around too quickly based on the criteria specified in RFC 2863 for high-capacity counters."

GROUP mplsHCOutSegmentPerfGroup
DESCRIPTION "This group is mandatory for those out-segment entries for which the object mplsOutSegmentPerfOctets wraps around too quickly based on the criteria specified in RFC 2863 for high-capacity counters."

GROUP mplsLsrNotificationGroup
DESCRIPTION "This group is only mandatory for those implementations which can efficiently implement the notifications contained in this group."

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

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

OBJECT mplsInSegmentNPop
SYNTAX Integer32 (1..1)
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. This object SHOULD be set to 1 if it is read-only."

OBJECT mplsInSegmentAddrFamily
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. A value of other(0) should be supported because there may be cases where the agent may not know about or support any address types."
OBJECT mplsInSegmentRowStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

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

-- mplsOutSegmentTable
OBJECT mplsOutSegmentInterface
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 mplsOutSegmentTopLabelPtr
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT mplsOutSegmentNextHopAddrType
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2) }
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. Only unknown(0), ipv4(1) and ipv6(2) support is required."

OBJECT mplsOutSegmentNextHopAddr
SYNTAX InetAddress (SIZE(0|4|16))
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. An implementation is only required to support unknown(0), ipv4(1) and ipv6(2) sizes."

OBJECT mplsOutSegmentRowStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

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

OBJECT       mplsXCAdminStatus
MIN-ACCESS   read-only
DESCRIPTION "Read only support is required."

OBJECT       mplsXCRowStatus
SYNTAX       RowStatus { active(1) }
MIN-ACCESS   read-only
DESCRIPTION "Write access is not required."

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

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

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

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

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

::= { mplsLsrCompliances 2 }

-- Units of conformance.

mplsInterfaceGroup OBJECT-GROUP
OBJECTS {
  mplsInterfaceLabelMinIn,
  mplsInterfaceLabelMaxIn,
  mplsInterfaceLabelMinOut,
  mplsInterfaceLabelMaxOut,
  mplsInterfaceTotalBandwidth,
  mplsInterfaceAvailableBandwidth,
  mplsInterfaceLabelParticipationType
}
STATUS  current
DESCRIPTION
  "Collection of objects needed for MPLS interface
  and interface performance information."
 ::= { mplsLsrGroups 1 }

mplsInSegmentGroup  OBJECT-GROUP
OBJECTS {
  mplsInSegmentIndexNext,
  mplsInSegmentInterface,
  mplsInSegmentLabel,
  mplsInSegmentLabelPtr,
  mplsInSegmentNPop,
  mplsInSegmentAddrFamily,
  mplsInSegmentXCIndex,
  mplsInSegmentOwner,
  mplsInSegmentRowStatus,
  mplsInSegmentStorageType,
  mplsInSegmentTrafficParamPtr,
  mplsInSegmentMapIndex
}
STATUS  current
DESCRIPTION
  "Collection of objects needed to implement an in-
  segment."
 ::= { mplsLsrGroups 2 }

mplsOutSegmentGroup  OBJECT-GROUP
OBJECTS {
  mplsOutSegmentIndexNext,
  mplsOutSegmentInterface,
  mplsOutSegmentPushTopLabel,
  mplsOutSegmentTopLabel,
  mplsOutSegmentTopLabelPtr,
  mplsOutSegmentNextHopAddrType,
  mplsOutSegmentNextHopAddr,
  mplsOutSegmentXCIndex,
  mplsOutSegmentOwner,
  mplsOutSegmentPerfOctets,
  mplsOutSegmentPerfDiscards,
  mplsOutSegmentPerfErrors,
  mplsOutSegmentRowStatus,
  mplsOutSegmentStorageType,
  mplsOutSegmentTrafficParamPtr
}
STATUS  current
DESCRIPTION
  "Collection of objects needed to implement an out-
segment."
 ::= { mplsLsrGroups 3 }

mplsXCGroup OBJECT-GROUP
OBJECTS {
 mplsXCIIndexNext,
 mplsXCLspId,
 mplsXCLabelStackIndex,
 mplsXCOwner,
 mplsXCStorageType,
 mplsXCAdminStatus,
 mplsXCOperStatus,
 mplsXCRowStatus,
 mplsXCNotificationsEnable
}
STATUS current
DESCRIPTION
 "Collection of objects needed to implement a
cross-connect entry."
 ::= { mplsLsrGroups 4 }

mplsPerfGroup OBJECT-GROUP
OBJECTS {
 mplsInSegmentPerfOctets,
 mplsInSegmentPerfPackets,
 mplsInSegmentPerfErrors,
 mplsInSegmentPerfDiscards,
 mplsInSegmentPerfDiscontinuityTime,
 mplsOutSegmentPerfOctets,
 mplsOutSegmentPerfPackets,
 mplsOutSegmentPerfDiscards,
 mplsOutSegmentPerfDiscontinuityTime,
 mplsInterfacePerfInLabelsInUse,
 mplsInterfacePerfInLabelLookupFailures,
 mplsInterfacePerfOutFragmentedPkts,
 mplsInterfacePerfOutLabelsInUse
}
STATUS current
DESCRIPTION
 "Collection of objects providing performance
 information
 about an LSR."
 ::= { mplsLsrGroups 5 }

mplsHCInSegmentPerfGroup OBJECT-GROUP
OBJECTS { mplsInSegmentPerfHCOctets }
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 { mplsOutSegmentPerfHCOctets }
STATUS current

DESCRIPTION
"Object(s) providing performance information specific to out-segments for which the object mplsInterfaceOutOctets wraps around too quickly."

::= { mplsLsrGroups 7 }

mplsLabelStackGroup OBJECT-GROUP
OBJECTS {
  mplsLabelStackLabel,
  mplsLabelStackLabelPtr,
  mplsLabelStackRowStatus,
  mplsLabelStackStorageType,
  mplsMaxLabelStackDepth,
  mplsLabelStackIndexNext
}
STATUS current

DESCRIPTION
"Objects needed to support label stacking."

::= { mplsLsrGroups 8 }

mplsLsrNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS {
  mplsXCUp,
  mplsXCDown
}
STATUS current

DESCRIPTION
"Set of notifications implemented in this module."

::= { mplsLsrGroups 9 }

END
11. Security Considerations

It is clear that this MIB module 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.

There are a number of management objects defined in this MIB module with 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. These are the tables and objects and their sensitivity/vulnerability:

- the mplsLsrInSegmentTable, mplsLsrOutSegmentTable, mplsXCTable, mplsOutSegmentPerfTable, mplsInterfacePerfTable, and mplsInSegmentPerfTable collectively contain objects to provision MPLS interfaces, LSPs and their associated parameters on an Label Switching Router (LSR). Unauthorized access to objects in these tables, could result in disruption of traffic on the network. This is especially true if an LSP has been established. The use of stronger mechanisms such as SNMPv3 security should be considered where possible. Specifically, SNMPv3 VACM and USM MUST be used with any v3 agent which implements this MIB module. Administrators should consider whether read access to these objects should be allowed, since read access may be undesirable under certain circumstances.

Some of the readable objects in this MIB module "i.e., objects with a MAX-ACCESS other than not-accessible" may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- the mplsLsrInSegmentTable, mplsLsrOutSegmentTable, mplsXCTable, mplsOutSegmentPerfTable, mplsInterfacePerfTable, and mplsInSegmentPerfTable collectively show the LSP network topology and its performance characteristics. If an Administrator does not want to reveal this information, then these tables should be considered sensitive/vulnerable.
SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure "for example by using IPSec", even then, 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 module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework "see [RFC3410], section 8", including full support for the SNMPv3 cryptographic mechanisms "for authentication and privacy".

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module, 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.

12. Acknowledgments

We wish to thank Ron Bonica, Adrian Farrel, Eric Gray, Tim Mancour, Keith McCloghrie, Bala Rajagopalan, Dan Tappan, Vasanthi Thirumalai, Joseph Benoit, Mike Piecuch, Joan Cucchiara. A special thanks to Bert Wijnen and Mike MacFaden for really getting the MIB module into shape.

13. IANA Considerations

As described in [MPLSMGMT] and as requested in the MPLS-TC-STD-MIB [RFC3811], MPLS related standards track MIB modules should be rooted under the mplsStdMIB subtree. There are 4 MPLS MIB Modules contained in this document, each of the following "IANA Considerations" subsections requests IANA for a new assignment under the mplsStdMIB subtree. New assignments can only be made via a Standards Action as specified in [RFC2434].

13.1. IANA Considerations for MPLS-LSR-STD-MIB

The IANA has assigned { mplsStdMIB 2 } to the MPLS-LSR-STD-MIB module specified in this document.
14. References

14.1. Normative References


IANA Family] Internet Assigned Numbers Authority (IANA), ADDRESS FAMILY NUMBERS, (http://www.iana.org/assignments/address-family-numbers), for MIB see: http://www.iana.org/assignments/ianaaddressfamilynumbers-mib

14.2. Informative References


15. Authors’ Addresses

Cheenu Srinivasan
Bloomberg L.P.
499 Park Ave., New York, NY 10022
Phone: +1-212-893-3682
EMail: cheenu@bloomberg.net

Arun Viswanathan
Force10 Networks, Inc.
1440 McCarthy Blvd
Milpitas, CA 95035
Phone: +1-408-571-3516
EMail: arunv@force10networks.com

Thomas D. Nadeau
Cisco Systems, Inc.
300 Beaver Brook Road
Boxboro, MA 01719
Phone: +1-978-936-1470
EMail: tnadeau@cisco.com
16. Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.