Network Working Group
Internet Draft
Expires: September 2000

Cheenu Srinivasan
Tachion Networks, Inc.

Arun Viswanathan
Force10 Networks

Thomas D. Nadeau
Cisco Systems, Inc.

MPLS Traffic Engineering Management Information Base Using SMIv2

draft-ietf-mpls-te-mib-03.txt

Status of this Memo

material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

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 Multi-Protocol Label Switching (MPLS) [MPLSArch] [MPLSFW] based traffic engineering.

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] based traffic engineering. This MIB should be used in conjunction with the companion document [LSRMIB] for MPLS based traffic engineering configuration and management.

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 MPLS architecture document [MPLSArch] and MPLS Label Switch Router MIB [LSRMIB]. Some frequently used terms are described next.

An explicitly routed LSP (ERLSP) is referred to as an MPLS tunnel. It consists of one in-segment and/or one out-segment at the ingress/egress LSRs, each segment being associated with one MPLS interface. These are also referred to as tunnel segments. Additionally, at an intermediate LSR, we model a connection as consisting of one or more in-segments and/or one or more out-segments. The binding or interconnection between in-segments and out-segments in performed using a cross-connect. These objects are defined in the MPLS Label Switch Router MIB [LSRMIB].

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 traffic engineering MIB is designed to satisfy the following requirements and constraints.

- The MIB must support the configuration of point-to-point unidirectional tunnels.

- The MIB should be able to support the configuration of point-to-point bi-directional tunnels.

- The MIB should be able to support the configuration of multi-point-to-point unidirectional tunnels.

- MPLS tunnels need not be interfaces, but it should be possible to configure a tunnel as an interface.

- The MIB should be able to support both manually configured MPLS tunnels as well as those set up via any MPLS signaling protocol.

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

5. Outline

Traffic engineering support for MPLS tunnels requires the following configuration.

- Setting up MPLS tunnels along with appropriate configuration parameters.

- Configuring tunnel loose and strict source routed hops.

These actions may need to be accompanied with corresponding actions using [LSRMIB] to establish and configure tunnel segments, if this is done manually. Also, the in-segment and out-segment performance tables, mplsInSegmentPerfTable and mplsOutSegmentPerfTable [LSRMIB], should be used to determine performance of the tunnels and tunnel segments.

5.1. Summary of Traffic Engineering MIB

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

- Tunnel table (mplsTunnelTable) for setting up MPLS tunnels.

- Tunnel hop table (mplsTunnelHopTable) for configuring strict and loose source routed MPLS tunnels hops.

- Resource table (mplsTunnelResourceTable) for setting up the tunnel resources.

- Tunnel Actual Route Table (mplsTunnelARHopTable) for viewing the actual route used by the tunnel.

These tables are described in the subsequent sections.

6. Brief Description of MIB Objects

The objects described in this section support the functionality described in documents [RSVPTun][CR-LDP]. The tables support both manually configured and signaled tunnels. Moreover, they provide the capability to associate two unidirectional tunnels to form a single bi-directional tunnel.

6.1. mplsTunnelTable

The mplsTunnelTable allows new MPLS tunnels to be created between an MPLS LSR and a remote endpoint, and existing tunnels to be reconfigured or removed. Note that we only support point-to-point tunnel segments, although multi-point-to-point and point-to-multi-point connections are supported by an LSR acting as a cross-connect. Each MPLS tunnel can thus have one out-segment originating at an LSR and/or one in-segment terminating at that LSR.

mplsTunnelTable does not define the in and out segments forming the tunnel. Instead, these are defined by creating rows in the in-segment and out-segment tables, defining relationships in the cross-connect table and referring to these rows in the mplsTunnelTable using a cross-connect index,

mplsTunnelXCID. These segment and cross-connect related objects are defined in [LSRMIB].

6.2. mplsTunnelHopTable

mplsTunnelHopTable is used to indicate the hops, strict or loose, for an MPLS tunnel defined in mplsTunnelTable, when it is established via signaling. Each row in this table is indexed primarily by the same index mplsTunnelIndex as the row of the corresponding tunnel in mplsTunnelTable. Each row also has a secondary index, mplsTunnelHopIndex, corresponding to the next hop of this tunnel. The scalar mplsTunnelMaxHops indicates the maximum number of hops that can be specified on each tunnel supported by this LSR.

6.3. mplsTunnelResourceTable

mplsTunnelResourceTable is used to indicate the resources required for a tunnel. Multiple tunnels may share the same resource by pointing to the same entry in this table. Tunnels that do not share resources must point to separate entries in this table.

6.4. mplsTunnelARHopTable

mplsTunnelARHopTable is used to indicate the actual route taken by a tunnel. This table is particularly

useful in cases where tunnel hops are loosely routed. This table will allow the user to see the actual path taken by the tunnel through the network. Support of this tunnel is dependent upon the MPLS signaling protocols providing this information.

7. Example of Tunnel Setup

This section contains an example of which MIB objects should be modified if one would like to create loosely routed, unidirectional traffic engineered tunnel, which spans two hops of a simple network. Note that these objects should be created on the "head-end" LSR.

In mplsTunnelTable:

```
{
  mplsTunnelIndex             = 1,
  mplsTunnelInstance          = 1,
  mplsTunnelName              = "My first tunnel ",
  mplsTunnelDescr             = "Here to there",
  mplsTunnelIsIf              = true (1),
  mplsTunnelIfIndex           = 12,
  mplsTunnelDirection         = in (1),
  mplsTunnelXCIndex           = 15,
  mplsTunnelSignallingProto   = none (1),
  mplsTunnelLocalCookie       = 123.123.123.1,
  mplsTunnelRemoteCookie      = 123.123.123.1,
  mplsTunnelIsMergeable       = false (1),
  mplsTunnelSetupPrio         = 0,
  mplsTunnelHoldingPrio       = 0,
```

```
  mplsTunnelIsPersistent      = false (1),
     mplsTunnelSessionAttributes = 0,
     mplsTunnelOwner             = snmp (1),
     mplsTunnelLocalProtectInUse = false (0),
     mplsTunnelResourceEntryIndex= 5,
     mplsTunnelInstancePriority  = 1,
     mplsTunnelRowStatus         = createAndGo (4)
   }

   In mplsTunnelResourceTable:
   {
     mplsTunnelResourceIndex     = 5,
     mplsTunnelInMaxRate         = 100,
     mplsTunnelInMeanRate        = 100,
     mplsTunnelInMaxBurstSize    = 100,
     mplsTunnelOutMaxRate        = 100,
     mplsTunnelOutMeanRate       = 100,
     mplsTunnelOutMaxBurstSize   = 100,
     mplsTunnelResourceRowStatus = createAndGo (4)
   }
```

The next two instances of mplsTunnelHopEntry are used to denote the hops this tunnel will take across the network. Note that mplsTunnelIndex, mplsTunnelInstance, and mplsTunnelHopIndex are used to index these objects. For the purposes of this example, all entries in the MplsTunnelHopTable will use the index of 1, which was defined for the mpslTunnelEntry above.

The following denotes the beginning of the network, or the first hop. We have used the fictitious LSR

identified by "123.123.125.1" as our example head-end router.

```
In mplsTunnelHopTable:
{
  mplsTunnelHopIndex              = 1,
  mplsTunnelHopAddrType           = 1,
  mplsTunnelHopIpv4Addr           = 123.123.125.1,
  mplsTunnelHopIpv4PrefixLen      = 9,
  mplsTunnelHopStrictOrLoose      = loose (2),
  mplsTunnelHopRowStatus          = createAndGo (4)
}
```

The following denotes the end of the network, or the last hop in our example. We have used the fictitious LSR identified by "123.123.126.1" as our end router.

```
In mplsTunnelHopTable:
{
  mplsTunnelHopIndex              = 2,
  mplsTunnelHopAddrType           = 1,
  mplsTunnelHopIpv4Addr           = 123.123.126.1,
  mplsTunnelHopIpv4PrefixLen      = 9,
  mplsTunnelHopStrictOrLoose      = loose (2),
  mplsTunnelHopOwner              = snmp (1),
  mplsTunnelHopRowStatus          = createAndGo (4)
}
```

7.1. 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 <TBD>.
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.

8. MPLS Traffic Engineering MIB Definitions

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

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   experimental, Integer32, 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
   MplsBitRate, MplsBurstSize
      FROM MPLS-LSR-MIB;

mplsTeMIB MODULE-IDENTITY
   LAST-UPDATED
      "200003031200Z"  -- 3 Mar 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 module contains managed object
        definitions for MPLS Traffic Engineering (TE) as
        defined in: Extensions to RSVP for LSP Tunnels,
        Awduche et al, Internet Draft <draft-mpls-rsvp-
        lsp-tunnel-02.txt>, March 1999; Constraint-Based
        LSP Setup using LDP, B. Jamoussi, Internet Draft
        <draft-ietf-mpls-cr-ldp-01.txt>, Feb. 1999;
        Requirements for Traffic Engineering Over MPLS,
        Awduche, D., J. Malcolm, J., Agogbua, J.,
        O'Dell, M., J. McManus, <rfc2702.txt>, September
        1999."

   -- Revision history.
REVISION
       "199907161200Z"  -- 16 July 1999 12:00:00 EST
```

```
   DESCRIPTION
       "Initial draft version."
   REVISION
       "200003101700Z"  -- 10 March 2000 17:00:00 EST
   DESCRIPTION
       "Initial version, published as RFC xxxx."

   ::= { experimental 95 }


-- Textual Conventions.

-- 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 label carried in the MPLS shim header is 20
        bits wide and the top 12 bits must be zero.  The
        frame relay label can be either 10, 17 or 23
        bits wide depending on the size of the DLCI
        field size and the top 22, 15, or 9 bits must be
        zero, respectively.  For an ATM interface, the
        lowermost 16 bits are interpreted as the VCI,
        the next 8 bits as the VPI and the remaining
        bits must be zero.  Also note 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-04.txt, April 1999.
        2. Use of Label Switching on Frame Relay
        Networks, Conta et al, draft-ietf-mpls-fr-
        03.txt, Nov. 1998."
    SYNTAX Integer32

MplsTunnelIndex ::= TEXTUAL-CONVENTION
    STATUS        current
    DESCRIPTION
        "Primary index into mplsTunnelTable."
    SYNTAX        INTEGER (0..65535)

MplsTunnelInstance ::= TEXTUAL-CONVENTION
    STATUS        current
    DESCRIPTION
        "Instance of the tunnel; secondary index into
        mplsTunnelTable."
    SYNTAX        INTEGER (0..65535)

MplsTunnelCookie ::= TEXTUAL-CONVENTION
    STATUS        current
    DESCRIPTION
        "A globally unique identifier that is assigned to
        each ERLSP.  This is assigned at the head end of
        the ERLSP and can be used by all LSRs to
        identify this ERLSP.  At the head end this
        cookie is maintained in the tunnel table as
        mplsTunnelLocalCookie.  For signaled tunnels
        this cookie is piggybacked by the signaling
```

```
        protocol to the remote end where the cookie is
        stored in the remote LSR's tunnel table as
        mplsTunnelRemoteCookie for the tunnel. For
        creating bi-directional tunnels the cookie is
        used to associate the two unidirectional ERLSPs
        as belonging to the same tunnel.

        It is recommended that the cookie value be
        assigned by concatenating the head-end LSR's IP
        address with the tunnel index.  For IPv4
        addresses this results in a 6-octet long
        cookie."
   SYNTAX        OCTET STRING (SIZE (0..255))

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


-- Top level components of this MIB.

-- tables, scalars
mplsTeObjects       OBJECT IDENTIFIER ::= { mplsTeMIB 1 }
-- traps
mplsTeNotifications OBJECT IDENTIFIER ::= { mplsTeMIB 2 }
-- conformance
mplsTeConformance   OBJECT IDENTIFIER ::= { mplsTeMIB 3 }
```

```
-- MPLS tunnel table.

mplsTunnelIndexNext OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This object contains the next appropriate value
        to be used for mplsTunnelIndex when creating
        entries in the mplsTunnelTable. If the number of
        unassigned entries is exhausted, this object
        will take on the value of 0.  To obtain the
        mplsTunnelIndex 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."
::= { mplsTeObjects 1 }

mplsTunnelTable OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsTunnelEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "The mplsTunnelTable allows new MPLS tunnels to
        be created between an LSR and a remote endpoint,
```

```
        and existing tunnels to be reconfigured or
        removed.  Note that only point-to-point tunnel
        segments are supported, although multi-point-to-
        point and point-to-multi-point connections are
        supported by an LSR acting as a cross-connect.
        Each MPLS tunnel can thus have one out-segment
        originating at this LSR and/or one in-segment
        terminating at this LSR."
   ::= { mplsTeObjects 2 }

mplsTunnelEntry OBJECT-TYPE
   SYNTAX        MplsTunnelEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents an MPLS
        tunnel.  An entry can be created by a network
        administrator or by an SNMP agent as instructed
        by an MPLS signaling protocol. Whenever a new
        entry is created with mplsTunnelIsIf set to
        true(1), then a corresponding entry is created
        in ifTable as well (see RFC 2233). The ifType of
        this entry is mplsTunnel(150)."
   REFERENCE
       "1. RFC 2233 - The Interfaces Group MIB using
        SMIv2, McCloghrie, K., and F. Kastenholtz, Nov.
        1997
        2. RFC 1700 - Assigned Numbers, Reynolds, J. and
        J. Postel, Oct. 1994"
   INDEX         { mplsTunnelIndex, mplsTunnelInstance }
      ::= { mplsTunnelTable 1 }
```

```
MplsTunnelEntry ::= SEQUENCE {
      mplsTunnelIndex             MplsTunnelIndex,
      mplsTunnelInstance          MplsTunnelInstance,
      mplsTunnelName              DisplayString,
      mplsTunnelDescr             DisplayString,
      mplsTunnelIsIf              TruthValue,
      mplsTunnelIfIndex           InterfaceIndexOrZero,
      mplsTunnelDirection         INTEGER,
      mplsTunnelXCIndex           Integer32,
      mplsTunnelSignallingProto    INTEGER,
      mplsTunnelLocalCookie        MplsTunnelCookie,
      mplsTunnelRemoteCookie       MplsTunnelCookie,
      mplsTunnelSetupPrio         INTEGER,
      mplsTunnelHoldingPrio        INTEGER,
      mplsTunnelSessionAttributes     INTEGER,
      mplsTunnelOwner             INTEGER,
      mplsTunnelResourceEntryIndex INTEGER,
      mplsTunnelInstancePriority   Integer32,
      mplsTunnelAdminStatus        INTEGER,
      mplsTunnelOperStatus         INTEGER,
      mplsTunnelRowStatus         RowStatus
   }

mplsTunnelIndex OBJECT-TYPE
   SYNTAX        MplsTunnelIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Uniquely identifies this row."
   ::= { mplsTunnelEntry 1 }
```

```
mplsTunnelInstance OBJECT-TYPE
   SYNTAX        MplsTunnelIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Uniquely identifies an instance of a tunnel. It
        is useful to identify multiple instances of
        tunnels for the purposes of backup and parallel
        tunnels."
   ::= { mplsTunnelEntry 2 }

mplsTunnelName OBJECT-TYPE
   SYNTAX        DisplayString
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The canonical name assigned to the tunnel. This
        name can be used to refer to the tunnel on the
        LSR s console port.  If mplsTunnelIsIf is set to
        true then the ifName of the interface
        corresponding to this tunnel should have a value
        equal to mplsTunnelName.  Also see the
        description of ifName in RFC 2233."
   REFERENCE
       "RFC 2233 - The Interfaces Group MIB using SMIv2,
        McCloghrie, K., and F. Kastenholtz, Nov. 1997"
   ::= { mplsTunnelEntry 3 }

mplsTunnelDescr OBJECT-TYPE
   SYNTAX        DisplayString
   MAX-ACCESS    read-create
   STATUS        current
```

```
   DESCRIPTION
       "A textual string containing information about
        the tunnel.  If there is no description this
        object contains a zero length string."
   ::= { mplsTunnelEntry 4 }

mplsTunnelIsIf OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes whether or not this tunnel corresponds
        to an interface represented in the interfaces
        group table. Note that if this variable is set
        to true then the ifName of the interface
        corresponding to this tunnel should have a value
        equal to mplsTunnelName.  Also see the
        description of ifName in RFC 2233."
   REFERENCE
       "RFC 2233 - The Interfaces Group MIB using SMIv2,
        McCloghrie, K., and F. Kastenholtz, Nov. 1997"
   DEFVAL        { false }
   ::= { mplsTunnelEntry 5 }

mplsTunnelIfIndex OBJECT-TYPE
   SYNTAX        InterfaceIndexOrZero
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If mplsTunnelIsIf is set to true, then this
        value contains the LSR-assigned ifIndex which
        corresponds to an entry in the interfaces table.
```

```
        Otherwise this variable should contain the value
        of zero indicating that a valid ifIndex was not
        assigned to this tunnel interface."
       REFERENCE
      "RFC 2233 - The Interfaces Group MIB using SMIv2,
       McCloghrie, K., and F. Kastenholtz, Nov. 1997"
   ::= { mplsTunnelEntry 6 }

mplsTunnelDirection OBJECT-TYPE
   SYNTAX        INTEGER { in(1), out(2), in-out(3) }
   MAX-ACCESS    read-create
   STATUS        current
       DESCRIPTION
       "Indicates whether this tunnel is unidirectional-
        incoming, unidirectional-outgoing, or bi-
        directional."
   ::= { mplsTunnelEntry 7 }

mplsTunnelXCIndex OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This variable represents an index into the
        mplsXCTable. This table identifies the segments
        that compose this tunnel, their characteristics,
        and relationships to each other."
   REFERENCE
       "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Label Switch Router Management Information
        Base Using SMIv2, Internet Draft <draft-ietf-
        mpls-lsr-mib-01.txt>, March 2000."
```

```
   DEFVAL        { 0 }
   ::= { mplsTunnelEntry 8 }

mplsTunnelSignallingProto OBJECT-TYPE
   SYNTAX        INTEGER { none(1), ldp(2), rsvp(3),
other(4) }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The signaling protocol, if any, which was used
        to setup this tunnel."
   DEFVAL        { none }
   ::= { mplsTunnelEntry 9 }

mplsTunnelLocalCookie OBJECT-TYPE
   SYNTAX        MplsTunnelCookie
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The local cookie assigned to the outgoing
        direction of this tunnel at this LSR."
   ::= { mplsTunnelEntry 10 }

mplsTunnelRemoteCookie OBJECT-TYPE
   SYNTAX        MplsTunnelCookie
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "The remote cookie assigned to the incoming
        direction of tunnel by the remote (head-end)
        LSR."
   ::= { mplsTunnelEntry 11 }
```

```
mplsTunnelSetupPrio OBJECT-TYPE
   SYNTAX        INTEGER (0..7)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Indicates the setup priority of this tunnel."
   REFERENCE
       "1. Extensions to RSVP for LSP Tunnels, Awduche
        et al, Internet Draft <draft-mpls-rsvp-lsp-
        tunnel-02.txt>, March 1999.,
        2. Constraint-Based LSP Setup using LDP,
        Jamoussi, Internet Draft <draft-ietf-mpls-cr-ldp-
        01.txt>, Feb. 1999."
   ::= { mplsTunnelEntry 12 }

mplsTunnelHoldingPrio OBJECT-TYPE
   SYNTAX        INTEGER (0..7)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Indicates the holding priority for this tunnel."
   REFERENCE
       "1. Extensions to RSVP for LSP Tunnels, Awduche
        et al, Internet Draft <draft-mpls-rsvp-lsp-
        tunnel-02.txt>, March 1999.
        2. Constraint-Based LSP Setup using LDP, B.
        Jamoussi, Internet Draft <draft-ietf-mpls-cr-ldp-
        01.txt>, Feb. 1999."
   ::= { mplsTunnelEntry 13 }

mplsTunnelSessionAttributes OBJECT-TYPE
```

```
   SYNTAX  BITS {
            ingressMayReroute (0),
            mergingPermitted (1),
            isPersistent (2),
            localProtectionAvailable (3),
            isPinned (4)
          }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This bitmask indicates optional session values
        for this tunnel. The following describes these
        bitfields:

       fastReroute: This flag indicates that the tunnel
        ingress node may choose to reroute this tunnel
        without tearing it down.

       mergingPermitted: This flag permits transit
        routers to merge this session with other RSVP
        sessions for the purpose of reducing resource
        overhead on downstream transit routers, thereby
        providing better network scalability.

       isPersistent: Indicates whether this tunnel
        should be restored automatically after a failure
        occurs.

       localProtectionAvailable: This flag permits
        transit routers to use a local repair mechanism
        which may result in violation of the explicit
        routing of this tunnel. When a fault is
```

```
        det4ected on an adjacent downstream link or
        node, a transit router can reroute traffic for
        fast service restoration.

       isPinned: This flag Indicates whether the loose-
        routed hops of this tunnel are to be pinned."
   REFERENCE
       "Extensions to RSVP for LSP Tunnels, Awduche et
        al, Internet Draft <draft-mpls-rsvp-lsp-tunnel-
        02.txt>, March 1999."
   ::= { mplsTunnelEntry 14 }

mplsTunnelOwner OBJECT-TYPE
   SYNTAX  INTEGER {
            snmp (1),
            ldp (2),
            rsvp (3),
            policyAgent (4),
            other (5)
         }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Indicates which protocol created and is
        responsible for managing this tunnel."
   ::= { mplsTunnelEntry 15 }

mplsTunnelResourceEntryIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
```

```
       "Indicates which entry from the
        mplsTunnelResourceTable is currently in use by
        this tunnel."
::= { mplsTunnelEntry 16 }

mplsTunnelInstancePriority OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value indicates which priority, in
        descending order with 0 indicating the lowest
        priority, within a group of tunnel instances. A
        group of tunnel instances is defined as a set of
        tunnels with the same mplsTunnelIndex in this
        table, but with a different mplsTunnelInstance.
        Tunnel group priorities are used to denote the
        priority at which particular a tunnel instance
        will supercede another. Instances of tunnels
        containing the same mplsTunnelInstancePriority
        will be used for load sharing."
::= { mplsTunnelEntry 17 }

mplsTunnelAdminStatus 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
```

```
       "Indicates the desired operational status of this
        tunnel."
   ::= { mplsTunnelEntry 18 }

mplsTunnelOperStatus OBJECT-TYPE
   SYNTAX        INTEGER {
         up(1),            -- ready to pass packets
         down(2),
         testing(3),       -- in some test mode
         unknown(4),       -- status cannot be determined
         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
       "Indicates the actual operational status of this
        tunnel, which is typically but not limited to, a
        function of the state of individual segments of
        this tunnel."
   ::= { mplsTunnelEntry 19 }

mplsTunnelRowStatus 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."
```

```
   ::= { mplsTunnelEntry 20 }

-- End of mplsTunnelTable

-- Begin of mplsTunnelResourceTable

mplsTunnelResourceIndexNext OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "This object contains the next appropriate value
        to be used for mplsTunnelResourceIndex when
        creating entries in the mplsTunnelResourceTable.
        If the number of unassigned entries is
        exhausted, this object will take on the value of
        0.  To obtain the mplsTunnelResourceIndex 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."
::= { mplsTeObjects 3 }

mplsTunnelResourceTable OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsTunnelResourceEntry
   MAX-ACCESS    not-accessible
   STATUS        current
```

```
   DESCRIPTION
       "The mplsTunnelResourceTable allows a manager to
        specify which resources are desired for an MPLS
        tunnel.  This table also allows several tunnels
        to point to a single entry in this table,
        implying that these tunnels should share
        resources."
   ::= { mplsTeObjects 4 }

mplsTunnelResourceEntry OBJECT-TYPE
   SYNTAX        MplsTunnelResourceEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents a set of
        resources for an MPLS tunnel.  An entry can be
        created by a network administrator or by an SNMP
        agent as instructed by any MPLS signaling
        protocol."
   INDEX         { mplsTunnelResourceIndex }
      ::= { mplsTunnelResourceTable 1 }

MplsTunnelResourceEntry ::= SEQUENCE {
      mplsTunnelResourceIndex         Unsigned32,
      mplsTunnelResourceInMaxRate        MplsBitRate,
      mplsTunnelResourceInMeanRate    MplsBitRate,
      mplsTunnelResourceInMaxBurstSize MplsBurstSize,
      mplsTunnelResourceOutMaxRate    MplsBitRate,
      mplsTunnelResourceOutMeanRate   MplsBitRate,
      mplsTunnelResourceOutMaxBurstSize MplsBurstSize,
      mplsTunnelResourceRowStatus        RowStatus
```

```
   }

mplsTunnelResourceIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Uniquely identifies this row."
::= { mplsTunnelResourceEntry 1 }

mplsTunnelResourceInMaxRate OBJECT-TYPE
   SYNTAX        MplsBitRate
   UNITS         "bits per second"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The maximum incoming rate in bits/second.  Note
        that setting mplsTunnelInMaxRate,
        mplsTunnelInMeanRate, and
        mplsTunnelInMaxBurstSize to 0 indicates best-
        effort treatment.
        This object is copied to an instance of
        mplsTSpecMaxRate in mplsTSpecTable the index of
        which is copied into the corresponding
        mplsInSegmentTSpecIndex."
   REFERENCE
       "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Label Switch Router Management Information
        Base Using SMIv2, Internet Draft <draft-ietf-
        mpls-lsr-mib-01.txt>, March 2000."
   DEFVAL { 0 }
   ::= { mplsTunnelResourceEntry 2 }
```

```
mplsTunnelResourceInMeanRate OBJECT-TYPE
   SYNTAX        MplsBitRate
   UNITS         "bits per second"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This object is copied into an instance of
        mplsTSpecMeanRate in the mplsTSpecTable. The
        index of this table is then copied into the
        corresponding mplsInSegmentTSpecIndex.

        When resource allocation is performed as
        requested by this TSpec object, it is copied
        into an entry in mplsTSpecTable [LSRMIB]:
        mplsTunnelInMeanRate to mplsTSpecMeanRate. The
        mplsTSpecDirection of this entry is set to
         in (1).  The mplsTSpecIndex value of this entry
        is copied to mplsInSegmentTSpecIndex of the
        corresponding in-segment entry."
   REFERENCE
       "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Label Switch Router Management Information
        Base Using SMIv2, Internet Draft <draft-ietf-
        mpls-lsr-mib-01.txt>, Feb. 2000."
   DEFVAL  { 0 }
   ::= { mplsTunnelResourceEntry 3 }

mplsTunnelResourceInMaxBurstSize OBJECT-TYPE
   SYNTAX        MplsBurstSize
   UNITS         "bytes"
   MAX-ACCESS    read-create
```

```
   STATUS        current
   DESCRIPTION
       "The maximum burst size in bytes.  This object is
        copied to mplsInSegmentMaxBurstSize of the
        corresponding in-segment.

        When resource allocation is performed as
        requested by this TSpec object, it is copied
        into an entry in mplsTSpecTable [LSRMIB]:
        mplsTunnelInMaxBurstSize to
        mplsTSpecMaxBurstSize. The mplsTSpecDirection of
        this entry is set to  in  (1).  The
        mplsTSpecIndex value of this entry is copied to
        mplsInSegmentTSpecIndex of the corresponding in-
        segment entry."
   REFERENCE
       "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Label Switch Router Management Information
        Base Using SMIv2, Internet Draft <draft-ietf-
        mpls-lsr-mib-01.txt>, March 2000."
   DEFVAL  { 0 }
   ::= { mplsTunnelResourceEntry 4 }

mplsTunnelResourceOutMaxRate OBJECT-TYPE
   SYNTAX        MplsBitRate
   UNITS         "bits per second"
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The maximum outgoing rate in bits/second.  Note
        that setting mplsTunnelOutMaxRate to 0 indicates
        best-effort treatment.  This object is copied to
```

```
        mplsOutSegmentMaxRate of the corresponding out-
        segment.

        When resource allocation is performed as
        requested by the following outgoing TSpec
        object, it is copied into an entry in
        mplsTSpecTable [LSRMIB] as follows:
        mplsTunnelOutMaxRate to mplsTSpecMaxRate. Also
        note that the mplsTSpecDirection of this entry
        is set to out (2).  The mplsTSpecIndex value of
        this entry is copied to mplsOutSegmentTSpecIndex
        of the corresponding out-segment entry."
    REFERENCE
        "Srinivasan, C., Viswanathan, A., and Nadeau, T.,
         MPLS Label Switch Router Management Information
         Base Using SMIv2, Internet Draft <draft-ietf-
         mpls-lsr-mib-01.txt>, March 2000."
    DEFVAL  { 0 }
    ::= { mplsTunnelResourceEntry 5 }

mplsTunnelResourceOutMeanRate OBJECT-TYPE
    SYNTAX        MplsBitRate
    UNITS         "bits per second"
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The mean outgoing rate in bits/second. Note that
         setting mplsTunnelOutMeanRate to 0 indicates
         best-effort treatment. This object is copied to
         mplsOutSegmentMeanRate of the corresponding out-
         segment.
```

```
        When resource allocation is performed as
        requested by the following outgoing TSpec
        object, it is copied into an entry in
        mplsTSpecTable [LSRMIB] as follows:
        mplsTunnelOutMeanRate to mplsTSpecMeanRate. Also
        note that the mplsTSpecDirection of this entry
        is set to out (2).  The mplsTSpecIndex value of
        this entry is copied to mplsOutSegmentTSpecIndex
        of the corresponding out-segment entry."
    REFERENCE
        "Srinivasan, C., Viswanathan, A., and T. Nadeau,
         MPLS Label Switch Router Management Information
         Base Using SMIv2, Internet Draft <draft-ietf-
         mpls-lsr-mib-01.txt>, March 2000."
    DEFVAL  { 0 }
    ::= { mplsTunnelResourceEntry 6 }

mplsTunnelResourceOutMaxBurstSize OBJECT-TYPE
    SYNTAX        MplsBurstSize
    UNITS         "bytes"
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The maximum burst size in bytes.  This object is
         copied to mplsOutSegmentMaxBurstSize of the
         corresponding out-segment. Note that setting
         mplsTunnelOutMaxBurstSize to 0 indicates best-
         effort treatment.

         When resource allocation is performed as
         requested by the following outgoing TSpec
         object, it is copied into an entry in
```

```
        mplsTSpecTable [LSRMIB] as follows:
        mplsTunnelOutMaxBurstSize to
        mplsTSpecMaxBurstSize. Also note that the
        mplsTSpecDirection of this entry is set to out
        (2).  The mplsTSpecIndex value of this entry is
        copied to mplsOutSegmentTSpecIndex of the
        corresponding out-segment entry."
    REFERENCE
        "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Label Switch Router Management Information
        Base Using SMIv2, Internet Draft <draft-ietf-
        mpls-lsr-mib-01.txt>, March 2000."
    DEFVAL  { 0 }
::= { mplsTunnelResourceEntry 7 }

mplsTunnelResourceRowStatus 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."
::= { mplsTunnelResourceEntry 8 }

-- End mplsTunnelResourceTable

-- Maximum number of tunnel hops supported.

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

```
   DESCRIPTION
       "The maximum number of hops that can be specified
        for a tunnel on this device."
   ::= { mplsTeObjects 5 }


-- Tunnel hop table.

mplsTunnelHopTable  OBJECT-TYPE
   SYNTAX          SEQUENCE OF MplsTunnelHopEntry
   MAX-ACCESS      not-accessible
   STATUS          current
   DESCRIPTION
       "The mplsTunnelHopTable is used to indicate the
        hops, strict or loose, for an MPLS tunnel
        defined in mplsTunnelTable, when it is
        established via signaling, for the outgoing
        direction of the tunnel.  Each row in this table
        is indexed primarily by the same index,
        mplsTunnelIndex, as the row of the corresponding
        tunnel in mplsTunnelTable.  Each row also has a
        secondary index mplsTunnelHopIndex corresponding
        to the next hop that this row corresponds to.
        The first row in the table is the first hop
        after the origination point of the tunnel.  In
        case we want to specify a particular interface
        on the originating LSR of an outgoing tunnel by
        which we want packets to exit the LSR, we
        specify this as the first hop for this tunnel in
        mplsTunnelHopTable."
   ::= { mplsTeObjects 6 }
```

```
mplsTunnelHopEntry  OBJECT-TYPE
   SYNTAX          MplsTunnelHopEntry
   MAX-ACCESS      not-accessible
   STATUS          current
   DESCRIPTION
       "An entry in this table represents a tunnel hop.
        An entry is created by a network administrator
        for signaled ERLSP set up by an MPLS signaling
        protocol."
   INDEX { mplsTunnelIndex, mplsTunnelInstance,
mplsTunnelHopIndex }
      ::= { mplsTunnelHopTable 1 }

MplsTunnelHopEntry ::= SEQUENCE {
      mplsTunnelHopIndex              Integer32,
      mplsTunnelHopAddrType           INTEGER,
      mplsTunnelHopIpv4Addr           IpAddress,
      mplsTunnelHopIpv4PrefixLen      INTEGER,
      mplsTunnelHopIpv6Addr           Ipv6Address,
      mplsTunnelHopIpv6PrefixLen      INTEGER,
      mplsTunnelHopAsNumber           INTEGER,
      mplsTunnelHopStrictOrLoose      INTEGER,
      mplsTunnelHopRowStatus          RowStatus
   }

mplsTunnelHopIndex OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Secondary index into this table identifying the
        particular hop."
```

```
   ::= { mplsTunnelHopEntry 1 }

mplsTunnelHopAddrType OBJECT-TYPE
   SYNTAX        INTEGER { ipV4(1), ipV6(2), asNumber(3)
}
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes the address type of this tunnel hop."
   DEFVAL        { ipV4 }
   ::= { mplsTunnelHopEntry 2 }

mplsTunnelHopIpv4Addr OBJECT-TYPE
   SYNTAX        IpAddress
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If mplsTunnelHopAddrType is set to ipV4(1), then
        this value will contain the IPv4 address of this
        hop. This object is otherwise insignificant and
        should contain a value of 0."
   ::= { mplsTunnelHopEntry 3 }

mplsTunnelHopIpv4PrefixLen OBJECT-TYPE
   SYNTAX        INTEGER (1..32)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If mplsTunnelHopAddrType is ipV4(1), then the
        prefix length for this hop's IPv4 address is
        contained herein. This object is otherwise
        insignificant and should contain a value of 0."
```

```
   ::= { mplsTunnelHopEntry 4 }

mplsTunnelHopIpv6Addr OBJECT-TYPE
   SYNTAX        Ipv6Address
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If the mplsTunnelHopAddrType is set to ipV6(2),
        then this variable contains the IPv6 address of
        this hop.  This object is otherwise
        insignificant and should contain a value of 0."
   ::= { mplsTunnelHopEntry 5 }

mplsTunnelHopIpv6PrefixLen OBJECT-TYPE
   SYNTAX        INTEGER (1..128)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If mplsTunnelHopAddrType is set to ipV6(2), this
        value will contain the prefix length for this
        hop's IPv6 address. This object is otherwise
        insignificant and should contain a value of 0."
   ::= { mplsTunnelHopEntry 6 }

mplsTunnelHopAsNumber OBJECT-TYPE
   SYNTAX        INTEGER (0..65535)
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If mplsTunnelHopAddrType is set to asNumber(3),
        then this value will contain the AS number of
        this hop. This object is otherwise insignificant
```

```
        and should contain a value of 0 to indicate this
        fact."
   ::= { mplsTunnelHopEntry 7 }

mplsTunnelHopStrictOrLoose OBJECT-TYPE
   SYNTAX        INTEGER { strict(1), loose(2) }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Denotes whether this is tunnel hop is routed in
        a strict or loose fashion."
   ::= { mplsTunnelHopEntry 8 }

mplsTunnelHopRowStatus 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."
   ::= { mplsTunnelHopEntry 9 }

-- End of mplsTunnelHopTable

-- Tunnel Actual Route Hop table.

mplsTunnelARHopTable  OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsTunnelARHopEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "The mplsTunnelARHopTable is used to indicate the
```

```
        (A)ctual (R)oute hops, for an MPLS tunnel
        defined in mplsTunnelTable, as set up by the
        specified MPLS signaling protocol, for the
        outgoing direction of the tunnel.  Each row in
        this table is indexed primarily by the same
        indices, mplsTunnelIndex and mplsTunnelInstance,
        as the row of the corresponding tunnel in
        mplsTunnelTable.  Each row also has a third
        index mplsTunnelARHopIndex, corresponding to the
        next hop that this row corresponds to.  The
        first row in the table is the first hop after
        the origination point of the tunnel.

        Please note that since the information necessary
        to build entries within this table are not
        provided by some MPLS signaling protocols; hence
        implementation of this table is optional. The
        information in this table is available from the
        MPLS signaling protocol after the path has been
        set-up."
   ::= { mplsTeObjects 7 }

mplsTunnelARHopEntry  OBJECT-TYPE
   SYNTAX        MplsTunnelARHopEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table represents an currently
        active tunnel hop. An entry is created in this
        table by the MPLS signaling protocol when the
        active path for this tunnel is known."
   INDEX { mplsTunnelIndex, mplsTunnelInstance,
```

```
mplsTunnelARHopIndex }
      ::= { mplsTunnelARHopTable 1 }

MplsTunnelARHopEntry ::= SEQUENCE {
      mplsTunnelARHopIndex             Integer32,
      mplsTunnelARHopAddrType          INTEGER,
      mplsTunnelARHopIpv4Addr          IpAddress,
      mplsTunnelARHopIpv4PrefixLen     INTEGER,
      mplsTunnelARHopIpv6Addr          Ipv6Address,
      mplsTunnelARHopIpv6PrefixLen     INTEGER,
      mplsTunnelARHopAsNumber          INTEGER
   }

mplsTunnelARHopIndex OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Secondary index into this table identifying the
        particular hop."
   ::= { mplsTunnelARHopEntry 1 }

mplsTunnelARHopAddrType OBJECT-TYPE
   SYNTAX        INTEGER { ipV4(1), ipV6(2), asNumber(3)
}
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Denotes the address type of this tunnel hop."
   DEFVAL        { ipV4 }
   ::= { mplsTunnelARHopEntry 2 }
```

```
mplsTunnelARHopIpv4Addr OBJECT-TYPE
   SYNTAX        IpAddress
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If mplsTunnelARHopAddrType is set to ipV4(1),
        then this value will contain the IPv4 address of
        this hop. This object is otherwise insignificant
        and should contain a value of 0."
   ::= { mplsTunnelARHopEntry 3 }

mplsTunnelARHopIpv4PrefixLen OBJECT-TYPE
   SYNTAX        INTEGER (1..32)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If mplsTunnelARHopAddrType is ipV4(1), then the
        prefix length for this hop's IPv4 address is
        contained herein. This object is otherwise
        insignificant and should contain a value of 0."
   ::= { mplsTunnelARHopEntry 4 }

mplsTunnelARHopIpv6Addr OBJECT-TYPE
   SYNTAX        Ipv6Address
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If the mplsTunnelARHopAddrType is set to
        ipV6(2), then this variable contains the IPv6
        address of this hop.  This object is otherwise
        insignificant and should contain a value of 0."
   ::= { mplsTunnelARHopEntry 5 }
```

```
mplsTunnelARHopIpv6PrefixLen OBJECT-TYPE
   SYNTAX        INTEGER (1..128)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If mplsTunnelARHopAddrType is set to ipV6(2),
        this value will contain the prefix length for
        this hop's IPv6 address. This object is
        otherwise insignificant and should contain a
        value of 0."
   ::= { mplsTunnelARHopEntry 6 }

mplsTunnelARHopAsNumber OBJECT-TYPE
   SYNTAX        INTEGER (0..65535)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "If mplsTunnelARHopAddrType is set to
        asNumber(3), then this value will contain the AS
        number of this hop. This object is otherwise
        insignificant and should contain a value of 0 to
        indicate this fact."
   ::= { mplsTunnelARHopEntry 7 }

-- End of mplsTunnelARHopTable


-- Notifications.

mplsTunnelUp NOTIFICATION-TYPE
   OBJECTS     { mplsTunnelIndex, mplsTunnelAdminStatus,
```

```
                mplsTunnelOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsTunnelOperStatus object for one of the
        configured tunnels 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
        mplsTunnelOperStatus."
   ::= { mplsTeNotifications 1 }

mplsTunnelDown NOTIFICATION-TYPE
   OBJECTS     { mplsTunnelIndex, mplsTunnelAdminStatus,
                 mplsTunnelOperStatus }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsTunnelOperStatus object for one of the
        configured tunnels 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
        mplsTunnelOperStatus."
   ::= { mplsTeNotifications 2 }

-- End of notifications.


-- Module compliance.

mplsTeGroups
```

```
   OBJECT IDENTIFIER ::= { mplsTeConformance 1 }

mplsTeCompliances
   OBJECT IDENTIFIER ::= { mplsTeConformance 2 }

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

      -- The mandatory group has to be implemented by all
      -- LSRs that originate/terminate ESLSPs/tunnels.
      -- In addition, depending on the type of tunnels
      -- supported, other groups become mandatory as
      -- explained below.

      MANDATORY-GROUPS    { mplsTunnelGroup }

      GROUP mplsTunnelManualGroup
      DESCRIPTION
          "This group is mandatory for devices which
           support manual configuration of tunnels, in
           addition to mplsTunnelGroup.  The following
           constraints apply: mplsTunnelSignallingProto
           should be at least read-only with a value of
           none(1)."

      GROUP mplsTunnelSignaledGroup
      DESCRIPTION
```

```
          "This group is mandatory for devices which
           support signaled tunnel set up, in addition
           to mplsTunnelGroup.  The following
           constraints apply: mplsTunnelSignallingProto
           should be at least read-only returning a
           value of ldp(2), or rsvp(3)."

      GROUP mplsTunnelIsNotIntfcGroup
      DESCRIPTION
          "This group is mandatory for devices which
           support tunnels that are not interfaces, in
           addition to mplsTunnelGroup.  The following
           constraints apply: mplsTunnelIsIf must at
           least be read-only returning false(1)."

      GROUP mplsTunnelIsIntfcGroup
      DESCRIPTION
          "This group is mandatory for devices which
           support tunnels that are interfaces, in
           addition to mplsTunnelGroup.  The following
           constraints apply: mplsTunnelIsIf must at
           least be read-only returning true(2)."


      -- mplsTunnelTable

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

      OBJECT      mplsTunnelInstance
```

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

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

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

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

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

      OBJECT      mplsTunnelDirection
      SYNTAX      INTEGER { out(2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "The values in(1) and in-out(3) need not be
           supported."

      OBJECT      mplsTunnelXCIndex
```

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

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

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

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

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

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

      OBJECT mplsTunnelSessionAttributes
      MIN-ACCESS  read-only
      DESCRIPTION
```

```
          "Write access is not required."

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

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

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

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

      OBJECT      mplsTunnelAdminStatus
      SYNTAX      INTEGER { up (1), down (2) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Only up and down states must be supported.
           Write access is not required."

      OBJECT      mplsTunnelOperStatus
      SYNTAX      INTEGER { up (1), down (2) }
      MIN-ACCESS  read-only
```

```
      DESCRIPTION
          "Only up and down states must be supported.
           Write access is not required."

      OBJECT      mplsTunnelRowStatus
      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. Write access is not
           required."


      -- mplsTunnelHopTable

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

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

      OBJECT      mplsTunnelHopIpv4Addr
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsTunnelHopIpv4PrefixLen
```

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

      OBJECT      mplsTunnelHopIpv6Addr
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsTunnelHopIpv6PrefixLen
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required."

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

      OBJECT      mplsTunnelHopStrictOrLoose
      SYNTAX      INTEGER { strict(1) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "loose(2) need not be supported. Write access
           is not required."

      OBJECT      mplsTunnelHopRowStatus
      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. Write access is not
            required."

   -- mplsTunnelResourceTable

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

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

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

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

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

      OBJECT      mplsTunnelResourceOutMaxBurstSize
      MIN-ACCESS  read-only
```

```
      DESCRIPTION
          "Write access is not required."

      OBJECT      mplsTunnelResourceRowStatus
      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. Write access is not
           required."


   ::= { mplsTeCompliances 1 }


-- Units of conformance.

mplsTunnelGroup OBJECT-GROUP
   OBJECTS { mplsTunnelIndexNext, mplsTunnelIndex,
   mplsTunnelInstance,
           mplsTunnelName, mplsTunnelDirection,
   mplsTunnelXCIndex,
           mplsTunnelIfIndex, mplsTunnelAdminStatus,
           mplsTunnelOperStatus, mplsTunnelRowStatus }
   STATUS  current
   DESCRIPTION
       "Necessary, but not sufficient, set of objects to
        implement tunnels.  In addition, depending on
        the type of the tunnels supported (for example,
        manually configured or signaled, persistent or
        non-persistent, etc.), the following other
```

```
        groups defined below are mandatory:
        mplsTunnelManualGroup and/or
        mplsTunnelSignaledGroup,
        mplsTunnelIsNotIntfcGroup and/or
        mplsTunnelIsIntfcGroup."
   ::= { mplsTeGroups 1 }

mplsTunnelManualGroup  OBJECT-GROUP
   OBJECTS { mplsTunnelSignallingProto }
   STATUS  current
   DESCRIPTION
       "Object(s) needed to implement manually
        configured tunnels."
   ::= { mplsTeGroups 2 }

mplsTunnelSignaledGroup OBJECT-GROUP
   OBJECTS { mplsTunnelSignallingProto,
             mplsTunnelLocalCookie,
             mplsTunnelRemoteCookie,
             mplsTunnelHopIndex, mplsTunnelHopAddrType,
             mplsTunnelHopIpv4Addr,
             mplsTunnelHopIpv4PrefixLen,
             mplsTunnelHopIpv6Addr,
             mplsTunnelHopIpv6PrefixLen,
             mplsTunnelHopStrictOrLoose,
             mplsTunnelHopRowStatus,
             mplsTunnelResourceEntryIndex }
   STATUS  current
   DESCRIPTION
       "Object needed to implement signaled tunnels."
   ::= { mplsTeGroups 3 }
```

```
mplsTunnelIsIntfcGroup OBJECT-GROUP
   OBJECTS { mplsTunnelIsIf }
   STATUS  current
   DESCRIPTION
       "Objects needed to implement tunnels that are
        interfaces."
   ::= { mplsTeGroups 4 }

mplsTunnelIsNotIntfcGroup OBJECT-GROUP
   OBJECTS { mplsTunnelIsIf }
   STATUS  current
   DESCRIPTION
       "Objects needed to implement tunnels that are not
        interfaces."
   ::= { mplsTeGroups 5 }

mplsTeNotificationGroup NOTIFICATION-GROUP
   NOTIFICATIONS { mplsTunnelUp, mplsTunnelDown }
   STATUS  current
   DESCRIPTION
       "Set of notifications implemented in this module.
        None is mandatory."
   ::= { mplsTeGroups 6 }

-- End of MPLS-TE-MIB
END
```

9. Security Considerations

There are a number of management objects defined in this MIB that 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.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is 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.

10. Acknowledgments

We wish to thank Eric Gray, Patrick Kerharo, and Pramod Koppol for their comments on this draft.

11. References

[MPLSArch] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-05.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.

[LSRMIB] Srinivasan, C., Viswanathan, A. and T. Nadeau, "MPLS Label Switch Router Management Information Base Using SMIv2", Internet Draft <draft-ietf-mpls-lsr-mib-01.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-04.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.

[CR-LDP] 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

12. 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.
300 Apollo Drive
Chelmsford, MA 01824
Phone: +1-978-244-3051
Email: tnadeau@cisco.com

13. Full Copyright Statement