Network Working Group
INTERNET-DRAFT
Updates: RFC3812, RFC4802
Intended Status: Standards Track
Created: April 17, 2009
Expires: October 17, 2009

A. Farrel (Editor)
Old Dog Consulting

S. Yasukawa
NTT

T. Nadeau
BT

Point-to-Multipoint Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) module

draft-ietf-mpls-p2mp-te-mib-09.txt

Status of this Memo

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for point-to-multipoint (P2MP) Multiprotocol Label Switching (MPLS) based traffic engineering (TE).

The MIB module defined in this document is applicable to P2MP MPLS-TE by extensions to the MPLS-TE MIB module defined in RFC 3812. It is equally applicable to P2MP Generalized MPLS (GMPLS) in association with the GMPLS TE MIB module defined in RFC 4802.

Table of Contents

0. Changes Since Previous Revision

[This section to be removed before publication as an RFC.]

- Fix example in Step 5 of Section 5.1.
- Typos.

1. Introduction

This memo defines a 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 point-to-multipoint (P2MP) Multiprotocol Label Switching (MPLS) traffic engineering (TE).

MPLS is defined in [RFC3031] and a signaling protocol for point-to-point (P2P) MPLS-TE (TE extensions to the Resource Reservation Protocol - RSVP-TE) is defined in [RFC3209]. RSVP-TE is extended for use in a P2MP environment by [RFC4875] following the requirements set out in [RFC4461].

[RFC3812] provides a MIB module for modeling and controlling P2P MPLS-TE in conjunction with Textual Conventions defined in [RFC3811]. In addition, [RFC3813] defines a MIB module for modeling and controlling an MPLS Label Switching Router (LSR) that may support MPLS-TE. An overview of MPLS MIB modules can be found in [RFC4221].

This document defines a MIB module for managing and controlling P2MP MPLS-TE. It builds on the objects and tables defined in [RFC3812] so that P2MP MPLS-TE management is an extension of P2P MPLS-TE management.

In addition, this document provides a description of how to use the LSR MIB module [RFC3813] to model and control an LSR that supports P2MP MPLS-TE.

The MIB module defined in this document and the usage of the LSR MIB module are equally applicable to Generalized MPLS (GMPLS) in association with the GMPLS TE MIB module defined in [RFC4802] and the GMPLS LSR MIB module defined in [RFC4803].

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 [RFC2119].

2. The Internet-Standard 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].

3. Feature List

The feature list for this MIB module is built on the feature list for the P2P MPLS-TE MIB module [RFC3812]. The features in the list below are marked with a star (*) if they are new features for this MIB module and with a circle (o) if they are satisfied by [RFC3812].

* The MIB module supports configuration of point-to-multipoint unidirectional tunnels.

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

o The MIB module supports tunnel establishment via an MPLS signaling protocol wherein the tunnel parameters are specified using this MIB module at the head end of the LSP, and end-to-end tunnel LSP establishment is accomplished via signaling. The MIB module also supports manually configured tunnels, i.e., those for which label associations at each hop of the tunnel LSP are provisioned by the administrator via the LSR MIB [RFC3813].

o The MIB module supports persistent, as well as non-persistent tunnels.

4. Outline

Point-to-point MPLS-TE utilizes MPLS tunnels running from source to destination. Each tunnel is supported by one or more label switched paths (LSPs). This is described in the signaling specification [RFC3209] and the document that describes the MPLS-TE MIB module [RFC3812].

P2MP MPLS-TE requires that MPLS tunnels are established from a single source point (the root) to one or more destination points (the leaves). P2MP MPLS-TE tunnels are also supported by one or more P2MP LSPs.

This document models a P2MP LSP as a set of source-to-leaf (S2L) sub-LSPs. That is, the path (or route) from the root to each leaf is separately specified for each leaf as a series of loose or strict hops. The combination (overlay) of the set of S2L LSPs results in the P2MP TE LSP. See [RFC4875] for a more detailed description of S2L LSPs and how they are combined to form a P2MP TE LSP.

Other configuration parameters associated with a P2MP MPLS-TE LSP describe the forwarding behavior of each LSR along the path of the LSP. It should be noted that, according to [RFC4461], these configuration parameters are invariant across the branches of a P2MP LSP toward different leaves. Thus, they can be configured for whole P2MP LSP, and do not need to be configured per leaf.

The setup of P2MP tunnels can be achieved as:

- Management actions only, by using [RFC3813] to configure cross-connections or forwarding state at the LSRs along the path of the tunnel. See Section 6 for more details.

- Control plane actions (i.e., signaling using [RFC4875]) under the direction of a management process, by using [RFC3812] and the MIB module defined in this document.

- Control plane actions ([RFC4875]) under the direction of some other management process, and monitored using [RFC3812] and the MIB module defined in this document.

Note that [RFC4802] defines a MIB module that can be used to manage and model Generalized MPLS (GMPLS) LSPs - it is a series of MIB objects and tables some of which extend tables in MPLS-TE-STD-MIB [RFC3812]. [RFC4461] and [RFC4875] are clear that they apply to MPLS-TE [RFC3031] and GMPLS [RFC3945]. This document describes a MIB module that can be used for both MPLS-TE and GMPLS P2MP LSPs, and all objects apply to both MPLS and GMPLS. Note, however, that GMPLS defines support for unidirectional and bidirectional LSP, while P2MP LSPs can only be unidirectional. Thus, the gmplsTunnelDirection object of GMPLS-TE-STD-MIB defined in [RFC4802] MUST be set to forward(0) when the LSP is a P2MP LSP.

The following sections describe the components of the P2MP MPLS-TE MIB module. The subsequent section provides an explanation and example of how the P2MP MPLS-TE MIB module can be used to configure and manage a P2MP tunnel when used in combination with the MPLS-TE MIB module defined in [RFC3812]. A further section describes how P2MP tunnels can be managed solely through the LSR MIB module defined in [RFC3813], and gives an example.

4.1. Summary of the P2MP MPLS Traffic Engineering MIB Module

The MIB module consists of the following objects and tables:

- The P2MP Tunnel table (mplsTeP2mpTunnelTable) sparse augments the MPLS-TE Tunnel table (mplsTunnelTable) and is used to set up and monitor P2MP MPLS-TE tunnels.

- The P2MP Tunnel Destination table (mplsTeP2mpTunnelDestTable) lists the destinations (leaves) of each P2MP MPLS-TE tunnel, provides the status of the tunnel to each destination, and supplies pointers into the configured hop table, actual route hop table, and computed hop table (mplsTunnelHopTable, mplsTunnelARHopTable, and mplsTunnelCHopTable) for the routes to each of the destinations.

- A small collection of scalars (mplsTeP2mpTunnelConfigured, mplsTeP2mpTunnelActive, and mplsTeP2mpTunnelTotalMaxHops) give information about the P2MP behavior of the LSR.

These tables and scalars are described in the following sections after a description of how the MPLS-TE-STD-MIB module [RFC3812] is used as a basis for MIB management and modeling of P2MP MPLS-TE.

4.2. Use of MPLS-TE-STD-MIB

The MIB module defined in this document builds on the objects and tables of MPLS-TE-STD-MIB defined in [RFC3812]. That is, most of the basic properties of the MPLS tunnel are modeled and managed by objects in MPLS-TE-STD-MIB, and new objects are only defined within this document where additional features or different behavior are required.

When an MPLS-TE tunnel is a P2MP tunnel, certain objects in the mplsTunnelTable have new meanings just as the signaling objects in RSVP-TE [RFC3209] have different meanings when the signaling messages are used to establish P2MP LSPs [RFC4875].

As indicated in the next section, the presence of a conceptual row in the mplsTeP2mpTunnelTable of the MIB module defined in this document shows that a tunnel defined in the corresponding conceptual row of the mplsTunnelTable of MPLS-TE-STD-MIB is a P2MP tunnel. Under those circumstances the following scalars and objects from the appropriate conceptual rows in MPLS-TE-STD-MIB MUST be interpreted as follows. The text below is supplementary to the Description clauses in [RFC3812].

mplsTunnelMaxHops

This object continues to refer to the maximum number of hops that can be configured to a single destination for a tunnel on this device. Thus, for a P2MP tunnel, this refers to the maximum number of hops that can be configured on this device to any individual destination of the tunnel.

A new object, mplsTeP2mpTunnelTotalMaxHops, is defined in this MIB module to supply the total number of hops across all destinations of a P2MP tunnel. mplsTeP2mpTunnelTotalMaxHops would normally be set larger than or equal to mplsTunnelMaxHops.

mplsTunnelEgressLSRId

This object continues to map to the field in the RSVP-TE Session Object that occupies the space used by the IPv4 Tunnel Endpoint Address [RFC3209], but for a P2MP tunnel, this object does not identify an address of the egress of the tunnel. Instead it contains the P2MP ID value that identifies the identifier of the set of destinations for the P2MP tunnel and is carried in the P2MP Session Object [RFC4875]. The Description clause for this object can be read as follows.

> "Identity of the egress LSR associated with this tunnel instance.
>
> When an entry in the mplsTeP2mpTunnelTable is present corresponding to this entry in the mplsTunnelTable, this object contains the P2MP ID that identifies the set of destinations of this tunnel and that is signaled in the P2MP ID field of the P2MP Session Object if the MPLS signaling protocol for this tunnel indicated by mplsTunnelSignallingProto in MPLS-TE-STD-MIB is rsvp(2)."

The destinations of the P2MP tunnel are found in the new mplsTeP2mpTunnelDestTable.

mplsTunnelXCPointer

If the tunnel is a P2MP tunnel as indicated by the presence of an entry in the mplsTeP2mpTunnelTable corresponding to this tunnel, this object is not used. This is because there is not necessarily a single entry in the mplsXCTable corresponding to the cross-connect for the LSP. Instead, the mplsXCTable entries for the LSP are found using the mplsXCIndex set to mplsTeP2mpTunnelP2mpXcIndex from the mplsTeP2mpTunnelTable; all mplsXCTable entries for the same P2MP LSP share a common value of mplsXCIndex.

If this object is present for a P2MP tunnel, it SHOULD contain the value 0.0.

mplsTunnelHopTableIndex

If the tunnel is a P2MP tunnel as indicated by the presence of an entry in the mplsTeP2mpTunnelTable corresponding to this tunnel,

this object is not used. This is because the destinations and paths to those destinations are found in the mplsTeP2mpTunnelDestTable.

If this object is present for a P2MP tunnel, it SHOULD contain the value 0.

mplsTunnelPathInUse

If the tunnel is a P2MP tunnel as indicated by the presence of an entry in the mplsTeP2mpTunnelTable corresponding to this tunnel, this object is not used. This is because the destinations and paths to those destinations are found in the mplsTeP2mpTunnelDestTable.

If this object is present for a P2MP tunnel, it SHOULD contain the value 0.

mplsTunnelARHopTableIndex

If the tunnel is a P2MP tunnel as indicated by the presence of an entry in the mplsTeP2mpTunnelTable corresponding to this tunnel, this object is not used. This is because the destinations and paths to those destinations are found in the mplsTeP2mpTunnelDestTable.

If this object is present for a P2MP tunnel, it SHOULD contain the value 0.

mplsTunnelCHopTableIndex

If the tunnel is a P2MP tunnel as indicated by the presence of an entry in the mplsTeP2mpTunnelTable corresponding to this tunnel, this object is not used. This is because the destinations and paths to those destinations are found in the mplsTeP2mpTunnelDestTable.

If this object is present for a P2MP tunnel, it SHOULD contain the value 0.

4.2.1. Backward Compatibility Concerns for MIB Read Operations

A concern may be raised with regard to the changed semantics of the objects listed in Section 4.2 within the MPLS-TE-STD-MIB module. What would happen if an implementation that was not P2MP-aware attempted to read from the MPLS-TE-STD-MIB module and encountered these objects with changed semantics? Would it attempt to handle a P2MP LSP as a P2P LSP, and would this potentially cause damage to the

implementation?

To clarify the situation, each of the objects with modified semantics is set out below. The term 'legacy system' is used to refer to a management station that is not aware of the P2MP-TE-STD-MIB and is not aware of the modified semantics of these objects.

mplsTunnelMaxHops

If examined by a legacy system, this object will be correctly interpreted as it continues to refer to the number of hops to any single destination. A legacy system will look to this object to determine how many hops it may insert into the path of a P2P LSP, and it will get the correct result from this object.

mplsTunnelEgressLSRId

This object reflects the value used in the signaling protocol in the Session Object. Although the precise semantic of the field is different in P2P and P2MP signaling, the use of the field as part of the tuple that identifies the LSP is unchanged.

If a P2MP tunnel is examined by a legacy system, this object will be correctly interpreted as the same size and format, and will be used to identify the LSP. This will not impact the operation of the legacy system.

mplsTunnelXCPointer

If a P2MP tunnel is examined by a legacy system, this object will report 0.0 giving the impression that no cross-connect entry has been set up for the LSP yet. This will not impact the operation of the legacy system.

mplsTunnelHopTableIndex

If a P2MP tunnel is examined by a legacy system, this object will report zero giving the impression that no tunnel hops have been configured. This will not impact the operation of the legacy system.

mplsTunnelPathInUse

If a P2MP tunnel is examined by a legacy system, this object will report zero giving the impression that no path is in use or available. This will not impact the operation of the legacy system.

mplsTunnelARHopTableIndex

If a P2MP tunnel is examined by a legacy system, this object will report zero giving the impression that no tunnel hops have been reported by the signaling protocol. This is a valid scenario and will not impact the operation of the legacy system.

mplsTunnelCHopTableIndex

If a P2MP tunnel is examined by a legacy system, this object will report zero giving the impression that no tunnel hops have been computed. This is a valid scenario and will not impact the operation of the legacy system.

4.2.2. Backward Compatibility Concerns for MIB Write Operations

Although a legacy system may be able to read objects in the MPLS-TE-STD-MIB which have modified semantics and operate correctly, there is also a concern that the legacy system might try to write to these objects, thus modifying the P2MP LSP in an unexpected way.

This section lists the objects with modified semantics and explains how each is safe against write access by a legacy system.

mplsTunnelMaxHops

If set by a legacy system, this object will correctly control the maximum number of hops in an LSP to a single destination as expected by the legacy system.

mplsTunnelEgressLSRId

A legacy system that was used to modify this object for a P2MP tunnel would be successful and would not damage the operation of the P2MP tunnel. All that would happen is that the identity of the tunnel would be changed.

mplsTunnelXCPointer

If this object is set for a P2MP tunnel by a legacy system, the SET will be successful, but the value (i.e. the object) will be ignored by the management agent and the object will not be used.

mplsTunnelHopTableIndex

If this object is set for a P2MP tunnel by a legacy system, the SET will be successful, but the value (i.e. the object) will be ignored by the management agent and the object will not be used.

mplsTunnelPathInUse

If this object is set for a P2MP tunnel by a legacy system, the SET will be successful, but the value (i.e. the object) will be ignored by the management agent and the object will not be used.

mplsTunnelARHopTableIndex

This object is read-only and cannot be set.

mplsTunnelCHopTableIndex

This object is read-only and cannot be set.

4.3. Scalars

There are three scalars defined for this MIB module.

mplsTeP2mpTunnelConfigured provides a read-only counter of the number of P2MP MPLS-TE tunnels that are configured on this LSR through this MIB module.

mplsTeP2mpTunnelActive provides a read-only counter of the number of P2MP MPLS-TE tunnels configured on this LSR through this MIB module that are currently active.

As described in Section 4.2, mplsTeP2mpTunnelTotalMaxHops is a read-only scalar that reports the maximum number of explicit route hops supported by this LSR for any single P2MP LSP configured or monitored through this MIB module. mplsTeP2mpTunnelTotalMaxHops would normally be set larger than or equal to mplsTunnelMaxHops.

4.4. mplsTeP2mpTunnelTable

The mplsTeP2mpTunnelTable extends (through a sparse augmentation) the MPLS Tunnel table (mplsTunnelTable) from MPLS-TE-STD-MIB [RFC3812] to allow P2MP MPLS-TE tunnels to be created, controlled, and monitored at any LSR in the network.

A P2MP MPLS-TE tunnel may be represented in the MIB, by defining it in the mplsTunnelTable and providing objects in this table to indicate that it is a P2MP tunnel and to define P2MP-specific properties of the tunnel.

4.5. mplsTeP2mpTunnelDestTable

P2MP LSPs have multiple destinations and, although the LSP parameters (such as bandwidth) for each destination are the same, the explicit route requested, computed, and signaled is different for each destination. The mplsTeP2mpTunnelDestTable encodes each destination and the information specific to the LSP to that destination.

mplsTeP2mpTunnelDestTable is indexed by a set of parameters that identify the P2MP LSP itself (tunnel index, tunnel instance, ingress LSR ID, egress LSR ID), the P2MP incoming and outgoing sub-groups (sub-group origin, sub-group ID), and the destination ID (i.e., a leaf).

4.6. mplsTeP2mpTunnelBranchPerfTable

Per-tunnel statistics are counted in mplsTunnelPerfTable in MPLS-TE-STD-MIB [RFC3812], but these objects are only partially

useful for a P2MP tunnel. The five objects in that table (mplsTunnelPerfPackets, mplsTunnelPerfHCPackets, mplsTunnelPerfErrors, mplsTunnelPerfBytes, and mplsTunnelPerfHCBytes) continue to be used for tunnels that forward packets, and reflect the counts of data received on the incoming interfaces and forwarded to the downstream interfaces.

However, in a P2MP tunnel, the downstream interfaces (out-segments) may behave differently and so it is appropriate to record the performance on each out-going branch. This is achieved through the mplsTeP2mpTunnelBranchPerfTable which is indexed by the tunnel identifiers and by the same identifier of the branch as is used in mplsTeP2mpTunnelDestTable.

4.7. Relationships Between MIB Tables

This section provides a diagrammatic representation of the relationships between MIB tables defined in this document as part of MPLS-TE-P2MP-STD-MIB, and the tables defined in MPLS-TE-STD-MIB in [RFC3812] and MPLS-LSR-STD-MIB in [RFC3813]. The dependencies between the various pre-existing MPLS-TE and LSR MIB tables can be seen in [RFC4221].

An arrow in the figure shows that the MIB table pointed from contains a reference to the MIB table pointed to.

```
   mplsTunnelPerfTable
      ^
      |
      v
   mplsTunnelTable-------->mplsTeP2mpTunnelTable
      ^    |                 |        |
      |    |                 |        |
      |    |                 v        |
      |    +------->mplsXCTable--+    v
      v                          |  mplsTeP2mpTunnelDestTable
   mplsTunnelResourceTable       |     |  |  |
      ^                          |     |  |  +--->mplsTunnelHopTable
      |                          |     |  |  +--->mplsTunnelCHopTable
   mplsInSegmentTable<-----------+     |  |  +--->mplsTunnelARHopTable
                                 |     |  |
                                 v     |  |
                mplsOutSegmentTable<---+  |
                                          v
                                     mplsTeP2mpTunnelBranchPerfTable
```

Figure 1 : Dependencies Between MIB Tables

5. Using the P2MP MPLS-TE MIB Module

This section describes how to use the P2MP MPLS-TE MIB module defined in this document to manage and model P2MP MPLS-TE LSPs. A subsection gives an example of usage.

A P2MP MPLS-TE LSP is modeled as a single LSP tunnel. That is, there is a single entry in the mplsTunnelTable of the MPLS-TE-STD-MIB defined in [RFC3812] for each instance of a P2MP LSP tunnel. As described in Section 4.2, certain of the objects in an entry in the mplsTunnelTable are not valid or have special meanings when the entry is used for a P2MP LSP tunnel.

When the MIB modules are used to configure a P2MP MPLS-TE LSP, an entry is first created in the mplsTunnelTable, and then corresponding entries are created in the mplsTeP2mpTunnelTable and the mplsTeP2mpTunnelDestTable from the MPLS-TE-P2MP-STD-MIB module defined in this document. The presence of a corresponding entry in the mplsTeP2mpTunnelTable indicates that an entry in the mplsTunnelTable relates to a P2MP not a P2P MPLS-TE LSP. Thus, the mplsTunnelAdminStatus object should not be set to up(1) until the entries in the mplsTeP2mpTunnelTable and the mplsTeP2mpTunnelDestTable have been completed.

5.1. Example Use of the P2MP MPLS-TE MIB Module

This section contains an example of the use of objects in MPLS-TE-STD-MIB and MPLS-TE-P2MP-STD-MIB to create a P2MP MPLS-TE LSP. Note that the objects described should be created on the "head-end" LSR.

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. 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.

Figure 2 shows the simple topology of the prospective LSP from its root at LSR R, through a branch node at LSR B, to its two destinations, LSRs D1 and D2.

```
                C1---D1
               /
              /
     R---A---B
              \
               \
                C2---D2
```

Figure 2 : Topology of a simple P2MP MPLS-TE LSP

Let us assign IP addresses to the LSRs as follows:

R	192.0.2.1
A	192.0.2.9
B	192.0.2.17
C1	192.0.2.33
C2	192.0.2.34
D1	192.0.2.65
D2	192.0.2.66

Step 1 - Define the resource requirements for the LSP

Let us assume that we require a best effort LSP.

In mplsTunnelResourceTable define as follows:

```
{
  mplsTunnelResourceIndex           = 9,
  mplsTunnelResourceMaxRate         = 0,
  mplsTunnelResourceMeanRate        = 0,
  mplsTunnelResourceMaxBurstSize    = 0,
  mplsTunnelResourceMeanBurstSize   = 0,
  mplsTunnelResourceExBurstSize     = 0,
  mplsTunnelResourceExBurstSize     = unspecified(1),
  mplsTunnelResourceWeight          = 0,
  mplsTunnelResourceRowStatus       = createAndGo(4)
}
```

Step 2 - Define the core parameters for the LSP tunnel.

In mplsTunnelTable define as follows:

```
{
  mplsTunnelIndex              = 4,
  mplsTunnelInstance           = 0,
  mplsTunnelIngressLSRId       = "192.0.2.1",
```

```
       -- The tunnel egress LSR ID is used to
       -- hold the P2MP ID for the P2MP LSP tunnel
       mplsTunnelEgressLSRId        = 328,
       mplsTunnelName               = "My first P2MP tunnel",
       mplsTunnelDescr              = "Here to there and there",
       mplsTunnelIsIf               = true(1),
       -- The XC pointer is not used for P2MP LSPs
       mplsTunnelXCPointer          = 0.0,
       -- This table entry is created by configuration not signaling
       mplsTunnelSignallingProto    = rsvp(2),
       mplsTunnelSetupPrio          = 0,
       mplsTunnelHoldingPrio        = 7,
       mplsTunnelSessionAttributes  = 0,
       mplsTunnelLocalProtectInUse  = false(2),
       mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.9,
       mplsTunnelInstancePriority   = 1,
       -- The index to the mplsTunnelHopTable from this table
       -- is not used
       mplsTunnelHopTableIndex      = 0,
       mplsTunnelIncludeAnyAffinity = 0,
       mplsTunnelIncludeAllAffinity = 0,
       mplsTunnelExcludeAnyAffinity = 0,
       mplsTunnelPathInUse          = 0,
       mplsTunnelRole               = head(1),
       -- Tunnel is not ready for admin status up
       mplsAdminStatus              = down(2),
       mplsTunnelRowStatus          = createAndGo(4)
     }
```

Note that any active or signaled instances of the above tunnel would appear with the same primary mplsTunnelIndex, but would have values greater than 0 for mplsTunnelInstance.

Step 3 - Create the P2MP Tunnel

In mplsTeP2mpTunnelTable define as follows:

```
     {
       mplsTeP2mpTunnelP2mpIntegrity      = true(1),
       -- This is the head end of the LSP and not a branch
       mplsTeP2mpTunnelBranchRole         = notBranch(1),
       -- There is no XC set up yet
       mplsTeP2mpTunnelP2mpXcIndex        = 0
       mplsTeP2mpTunnelRowStatus          = createAndGo(4)
     }
```

Step 4 - Create the configured explicit routes for the LSP

Two pieces of explicit path are required. The first runs from R to D1, and the second from B to D2. See [RFC4875] for a discussion of the construction of explicit routes for P2MP MPLS-TE LSPs.

In mplsTunnelHopTable define as follows:

```
{
  mplsTunnelHopListIndex          = 1,
  mplsTunnelPathOptionIndex       = 1,
  mplsTunnelHopIndex              = 1,
  mplsTunnelHopAddrType           = ipv4(1),
  mplsTunnelHopIpAddr             = "192.0.2.9",
  mplsTunnelHopIpPrefixLen        = 32,
  mplsTunnelHopType               = strict(2),
  mplsTunnelHopInclude            = true(1),
  mplsTunnelHopPathOptionName     = "Here to there",
  mplsTunnelHopEntryPathComp      = explicit(2),
  mplsTunnelHopRowStatus          = createAndGo(4)
}

{
  mplsTunnelHopListIndex          = 1,
  mplsTunnelPathOptionIndex       = 1,
  mplsTunnelHopIndex              = 2,
  mplsTunnelHopAddrType           = ipv4(1),
  mplsTunnelHopIpAddr             = "192.0.2.17",
  mplsTunnelHopIpPrefixLen        = 32,
  mplsTunnelHopType               = strict(2),
  mplsTunnelHopInclude            = true(1),
  mplsTunnelHopPathOptionName     = "Here to there",
  mplsTunnelHopEntryPathComp      = explicit(2),
  mplsTunnelHopRowStatus          = createAndGo(4)
}

{
  mplsTunnelHopListIndex          = 1,
  mplsTunnelPathOptionIndex       = 1,
  mplsTunnelHopIndex              = 3,
  mplsTunnelHopAddrType           = ipv4(1),
  mplsTunnelHopIpAddr             = "192.0.2.33",
  mplsTunnelHopIpPrefixLen        = 32,
  mplsTunnelHopType               = strict(2),
  mplsTunnelHopInclude            = true(1),
  mplsTunnelHopPathOptionName     = "Here to there",
  mplsTunnelHopEntryPathComp      = explicit(2),
  mplsTunnelHopRowStatus          = createAndGo(4)
```

```
     }
     {
       mplsTunnelHopListIndex          = 1,
       mplsTunnelPathOptionIndex       = 1,
       mplsTunnelHopIndex              = 4,
       mplsTunnelHopAddrType           = ipv4(1),
       mplsTunnelHopIpAddr             = "192.0.2.65",
       mplsTunnelHopIpPrefixLen        = 32,
       mplsTunnelHopType               = strict(2),
       mplsTunnelHopInclude            = true(1),
       mplsTunnelHopPathOptionName     = "Here to there",
       mplsTunnelHopEntryPathComp      = explicit(2),
       mplsTunnelHopRowStatus          = createAndGo(4)
     }

     {
       mplsTunnelHopListIndex          = 2,
       mplsTunnelPathOptionIndex       = 1,
       mplsTunnelHopIndex              = 1,
       mplsTunnelHopAddrType           = ipv4(1),
       mplsTunnelHopIpAddr             = "192.0.2.17",
       mplsTunnelHopIpPrefixLen        = 32,
       mplsTunnelHopType               = strict(2),
       mplsTunnelHopInclude            = true(1),
       mplsTunnelHopPathOptionName     = "Here to there",
       mplsTunnelHopEntryPathComp      = explicit(2),
       mplsTunnelHopRowStatus          = createAndGo(4)
     }

     {
       mplsTunnelHopListIndex          = 2,
       mplsTunnelPathOptionIndex       = 1,
       mplsTunnelHopIndex              = 2,
       mplsTunnelHopAddrType           = ipv4(1),
       mplsTunnelHopIpAddr             = "192.0.2.34",
       mplsTunnelHopIpPrefixLen        = 32,
       mplsTunnelHopType               = strict(2),
       mplsTunnelHopInclude            = true(1),
       mplsTunnelHopPathOptionName     = "Here to there",
       mplsTunnelHopEntryPathComp      = explicit(2),
       mplsTunnelHopRowStatus          = createAndGo(4)
     }

     {
       mplsTunnelHopListIndex          = 2,
       mplsTunnelPathOptionIndex       = 1,
       mplsTunnelHopIndex              = 3,
       mplsTunnelHopAddrType           = ipv4(1),
       mplsTunnelHopIpAddr             = "192.0.2.66",
```

```
         mplsTunnelHopIpPrefixLen        = 32,
         mplsTunnelHopType               = strict(2),
         mplsTunnelHopInclude            = true(1),
         mplsTunnelHopPathOptionName     = "Here to there",
         mplsTunnelHopEntryPathComp      = explicit(2),
         mplsTunnelHopRowStatus          = createAndGo(4)
       }
```

Step 5 - Create the destinations for the P2MP LSP tunnel

In mplsTeP2mpTunnelDestTable define as follows:

```
       {
         mplsTeP2mpTunnelDestSrcSubGroupOriginType = unknown(0),
         mplsTeP2mpTunnelDestSrcSubGroupOrigin     = "",
         mplsTeP2mpTunnelDestSrcSubGroupID         = 0,
         mplsTeP2mpTunnelDestSubGroupOriginType    = ipv4(1),
         mplsTeP2mpTunnelDestSubGroupOrigin        = "192.0.2.1",
         mplsTeP2mpTunnelDestSubGroupID            = 132,
         mplsTeP2mpTunnelDestDestinationType       = ipv4(1),
         mplsTeP2mpTunnelDestDestination           = "192.0.2.65",
         mplsTeP2mpTunnelDestHopTableIndex         = 1,
         mplsTeP2mpTunnelDestPathInUse             = 1,
         mplsTeP2mpTunnelDestAdminStatus           = up(1),
         mplsTeP2mpTunnelDestRowStatus             = createAndGo(4)
       }

       {
         mplsTeP2mpTunnelDestSrcSubGroupOriginType = unknown(0),
         mplsTeP2mpTunnelDestSrcSubGroupOrigin     = "",
         mplsTeP2mpTunnelDestSrcSubGroupID         = 0,
         mplsTeP2mpTunnelDestSubGroupOriginType    = ipv4(1),
         mplsTeP2mpTunnelDestSubGroupOrigin        = "192.0.2.1",
         mplsTeP2mpTunnelDestSubGroupID            = 132,
         mplsTeP2mpTunnelDestDestinationType       = ipv4(1),
         mplsTeP2mpTunnelDestDestination           = "192.0.2.66",
         mplsTeP2mpTunnelDestHopTableIndex         = 2,
         mplsTeP2mpTunnelDestPathInUse             = 1,
         mplsTeP2mpTunnelDestAdminStatus           = up(1),
         mplsTeP2mpTunnelDestRowStatus             = createAndGo(4)
       }
```

Step 6 - Activate the tunnel

In mplsTunnelTable define as follows:

```
  {
    mplsTunnelIndex              = 4,
    mplsTunnelInstance           = 0,
    mplsTunnelIngressLSRId       = "192.0.2.1",
    mplsTunnelEgressLSRId        = 328,
    -- Activate the tunnel
    mplsAdminStatus              = up(1)
  }
```

5.2. Example Transit LSR Inspection

The MPLS-TE-P2MP-STD-MIB module can be used at the head end of a P2MP LSP to configure, manage, and monitor the LSP. This is described in Section 5.1.

The MIB module may also be used to monitor P2MP LSPs at transit and egress LSRs. Although many objects in the MIB module are writeable, as with MPLS-TE-STD-MIB, those objects are not normally implemented as writeable except at the head end LSRs.

This section provides a simple example of the use of the P2MP MPLS-TE MIB module at a transit LSR where the module is used to inspect the LSPs. The example uses the topology shown in Figure 2 and the LSP set out in Section 5.1. Consider the situation at LSR B in the figure.

LSR B will receive a single Path message from LSR A (or two separate Path messages depending on implementation - see [RFC4875]) and will send a Path message onwards to LSRs C1 and C2. Similarly, LSR B will receive Resv messages from LSRs C1 and C2, and will send a Resv (or two Resv messages according to implementation - see [RFC4875]) to LSR A. Once the LSP has been set up and the signaling protocol has reached a stable state, the tables in the MPLS-TE-STD-MIB and MPLS-TE-P2MP-STD-MIB can be read as follows.

An entry in mplsTunnelTable provides the base information for the P2MP tunnel.

```
  {
    mplsTunnelIndex             = Path.Session.TunnelID
    mplsTunnelInstance          = Path.SenderTemplate.LSP_ID
    mplsTunnelIngressLSRId      = Path.Session.ExtendedTunnelID
    mplsTunnelEgressLSRId       = Path.Session.P2MPID
    mplsTunnelName              = Path.SessionAttribute.SessionName
    mplsTunnelDescr             = absent ("") or autogenerated
```

```
    mplsTunnelIsIf              = false(2)
    mplsTunnelIfIndex           = 0
    mplsTunnelOwner             = rsvpTe(6)
    mplsTunnelRole              = transit(2) or tail(3)
    mplsTunnelXCPointer         = absent or 0.0 Not used for P2MP LSP
    mplsTunnelSignallingProto   = rsvp(2)
    mplsTunnelSetupPrio         = Path.SessionAttribute.SetupPr
    mplsTunnelHoldingPrio       = Path.SessionAttribute.HoldPr
    mplsTunnelSessionAttributes = Path.SessionAttribute.Flags
    mplsTunnelLocalProtectInUse = Resv.RecordRoute.Flags
    mplsTunnelResourcePointer   = points to the traffic parameter
                                  specification for this tunnel
    mplsTunnelPrimaryInstance   = mplsTunnelInstance
    mplsTunnelInstancePriority  = 0
    mplsTunnelHopTableIndex     = 0
    mplsTunnelPathInUse         = 0
    mplsTunnelARHopTableIndex   = 0
    mplsTunnelCHopTableIndex    = 0
    mplsTunnelIncludeAnyAffinity= Path.SessionAttribute.IncludeAny
    mplsTunnelIncludeAllAffinity= Path.SessionAttribute.IncludeAll
    mplsTunnelExcludeAnyAffinity= Path.SessionAttribute.ExcludeAny
    mplsTunnelTotalUpTime       = time since Resv sent
    mplsTunnelInstanceUpTime    = time since Resv sent
    mplsTunnelPrimaryUpTime     = time since Resv sent
    mplsTunnelPathChanges       = 0
    mplsTunnelLastPathChange    = time since Resv sent
    mplsTunnelCreationTime      = time since Resv sent
    mplsTunnelStateTransitions  = 1
    mplsTunnelAdminStatus       = up(1)
    mplsTunnelOperStatus        = up(1)
    mplsTunnelRowStatus         = active(1)
    mplsTunnelStorageType       = volatile(2)
  }
```

An entry in mplsTeP2mpTunnelTable indicates that the tunnel is a P2MP tunnel, and provides the parameters specific to its P2MP nature. The index objects (mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId, and mplsTunnelEgressLSRId) are identical in value to the entry in the mplsTunnelTable.

```
  {
    mplsTeP2mpTunnelP2mpIntegrity   = Path.LSPAttributes.Flags
    mplsTeP2mpTunnelBranchRole      = branch(2)
    mplsTeP2mpTunnelP2mpXcIndex     = index into mplsXCTable
    mplsTeP2mpTunnelRowStatus       = active(1)
    mplsTeP2mpTunnelStorageType     = volatile(2)
  }
```

Two entries are required in the mplsTeP2mpTunnelDestTable. Index values of the mplsTeP2mpTunnelDestSubGroupID object will have been assigned automatically and will not have been generated by reading the mplsTeP2mpTunnelSubGroupIDNext object. Other index values (mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId, and mplsTunnelEgressLSRId) are identical in value to those in the entry in the mplsTunnelTable and the mplsTeP2mpTunnelTable. The remaining index values are determined from the local LSR's address and the destinations of the P2MP tunnel.

```
  {
    mplsTeP2mpTunnelDestSrcSubGroupOriginType = ipv4(1)
    mplsTeP2mpTunnelDestSrcSubGroupOrigin     =
                                      Path.SenderTemplate.SubGpOrigin
    mplsTeP2mpTunnelDestSrcSubGroupID         =
                                          Path.SenderTemplate.SubGpID
    mplsTeP2mpTunnelDestSubGroupOriginType    = ipv4(1)
    mplsTeP2mpTunnelDestSubGroupOrigin        = "192.0.2.17"
    mplsTeP2mpTunnelDestSubGroupID            = 1
    mplsTeP2mpTunnelDestDestinationType       = ipv4(1)
    mplsTeP2mpTunnelDestDestination           = "192.0.2.65"
    mplsTeP2mpTunnelDestBranchOutSegment      =
                                       index into mplsOutSegmentTable
    mplsTeP2mpTunnelDestHopTableIndex         =
                                    index into the mplsTunnelHopTable
    mplsTeP2mpTunnelDestPathInUse             = 1
    mplsTeP2mpTunnelDestCHopTableIndex        =
                                   index into the mplsTunnelCHopTable
    mplsTeP2mpTunnelDestARHopTableIndex       =
                                  index into the mplsTunnelARHopTable
    mplsTeP2mpTunnelDestTotalUpTime           = time since Resv sent
    mplsTeP2mpTunnelDestInstanceUpTime        = time since Resv sent
    mplsTeP2mpTunnelDestPathChanges           = 1
    mplsTeP2mpTunnelDestLastPathChange        = time since Resv sent
    mplsTeP2mpTunnelDestCreationTime          = time since Resv sent
    mplsTeP2mpTunnelDestStateTransitions      = 1
    mplsTeP2mpTunnelDestDiscontinuityTime     = 0
    mplsTeP2mpTunnelDestAdminStatus           = up(1)
    mplsTeP2mpTunnelDestOperStatus            = up(1)
    mplsTeP2mpTunnelDestRowStatus             = active(1)
    mplsTeP2mpTunnelDestStorageType           = volatile(2)
  }
```

```
     {
       mplsTeP2mpTunnelDestSrcSubGroupOriginType = ipv4(1)
       mplsTeP2mpTunnelDestSrcSubGroupOrigin     =
                                         Path.SenderTemplate.SubGpOrigin
       mplsTeP2mpTunnelDestSrcSubGroupID         =
                                             Path.SenderTemplate.SubGpID
       mplsTeP2mpTunnelDestSubGroupOriginType    = ipv4(1)
       mplsTeP2mpTunnelDestSubGroupOrigin        = "192.0.2.17"
       mplsTeP2mpTunnelDestSubGroupID            = 2
       mplsTeP2mpTunnelDestDestinationType       = ipv4(1)
       mplsTeP2mpTunnelDestDestination           = "192.0.2.66"
       mplsTeP2mpTunnelDestBranchOutSegment      =
                                          index into mplsOutSegmentTable
       mplsTeP2mpTunnelDestHopTableIndex         =
                                       index into the mplsTunnelHopTable
       mplsTeP2mpTunnelDestPathInUse             = 1
       mplsTeP2mpTunnelDestCHopTableIndex        =
                                      index into the mplsTunnelCHopTable
       mplsTeP2mpTunnelDestARHopTableIndex       =
                                     index into the mplsTunnelARHopTable
       mplsTeP2mpTunnelDestTotalUpTime           = time since Resv sent
       mplsTeP2mpTunnelDestInstanceUpTime        = time since Resv sent
       mplsTeP2mpTunnelDestPathChanges           = 1
       mplsTeP2mpTunnelDestLastPathChange        = time since Resv sent
       mplsTeP2mpTunnelDestCreationTime          = time since Resv sent
       mplsTeP2mpTunnelDestStateTransitions      = 1
       mplsTeP2mpTunnelDestDiscontinuityTime     = 0
       mplsTeP2mpTunnelDestAdminStatus           = up(1)
       mplsTeP2mpTunnelDestOperStatus            = up(1)
       mplsTeP2mpTunnelDestRowStatus             = active(1)
       mplsTeP2mpTunnelDestStorageType           = volatile(2)
     }
```

A single entry in mplsTunnelResourceTable is automatically created to reflect the reservation request on the upstream segment and both of the downstream branches. The information is gathered from the received Path message. The table entry is pointed to by mplsTunnelResourcePointer.

The index value (mplsTunnelResourceIndex) is automatically generated.

```
     {
       mplsTunnelResourceIndex           = 33
       mplsTunnelResourceMaxRate         = 0
       mplsTunnelResourceMeanRate        = 0
       mplsTunnelResourceMaxBurstSize    = 0
       mplsTunnelResourceMeanBurstSize   = 0
       mplsTunnelResourceExBurstSize     = 0
```

```
      mplsTunnelResourceExBurstSize     = unspecified(1)
      mplsTunnelResourceWeight          = 0
      mplsTunnelResourceRowStatus       = active(1)
      mplsTunnelResourceStorageType     = volatile(2)
    }
```

Finally, entries may also be read from the tunnel hop tables. mplsTunnelHopTable contains the route information received in the incoming Path message. It is separated out to refer to the two separate downstream branches, and the entries are identified by the corresponding values of mplsTeP2mpTunnelDestHopTableIndex. There are four hops in total in our example.

```
    {
      mplsTunnelHopListIndex           = 27
      mplsTunnelPathOptionIndex        = 1
      mplsTunnelHopIndex               = 1
      mplsTunnelHopAddrType            = ipv4(1)
      mplsTunnelHopIpAddr              = "192.0.2.33"
      mplsTunnelHopIpPrefixLen         = 32
      mplsTunnelHopType                = strict(2)
      mplsTunnelHopInclude             = true(1)
      mplsTunnelHopPathOptionName      = ""
      mplsTunnelHopEntryPathComp       = explicit(2)
      mplsTunnelHopRowStatus           = active(1)
      mplsTunnelHopStorageType         = volatile(2)
    }

    {
      mplsTunnelHopListIndex           = 27
      mplsTunnelPathOptionIndex        = 1
      mplsTunnelHopIndex               = 2
      mplsTunnelHopAddrType            = ipv4(1)
      mplsTunnelHopIpAddr              = "192.0.2.65"
      mplsTunnelHopIpPrefixLen         = 32
      mplsTunnelHopType                = strict(2)
      mplsTunnelHopInclude             = true(1)
      mplsTunnelHopPathOptionName      = ""
      mplsTunnelHopEntryPathComp       = explicit(2)
      mplsTunnelHopRowStatus           = active(1)
      mplsTunnelHopStorageType         = volatile(2)
    }

    {
      mplsTunnelHopListIndex           = 33
      mplsTunnelPathOptionIndex        = 1
      mplsTunnelHopIndex               = 1
      mplsTunnelHopAddrType            = ipv4(1)
```

```
       mplsTunnelHopIpAddr             = "192.0.2.34"
       mplsTunnelHopIpPrefixLen        = 32
       mplsTunnelHopType               = strict(2)
       mplsTunnelHopInclude            = true(1)
       mplsTunnelHopPathOptionName     = ""
       mplsTunnelHopEntryPathComp      = explicit(2)
       mplsTunnelHopRowStatus          = active(1)
       mplsTunnelHopStorageType        = volatile(2)
     }

     {
       mplsTunnelHopListIndex          = 33
       mplsTunnelPathOptionIndex       = 1
       mplsTunnelHopIndex              = 2
       mplsTunnelHopAddrType           = ipv4(1)
       mplsTunnelHopIpAddr             = "192.0.2.66"
       mplsTunnelHopIpPrefixLen        = 32
       mplsTunnelHopType               = strict(2)
       mplsTunnelHopInclude            = true(1)
       mplsTunnelHopPathOptionName     = ""
       mplsTunnelHopEntryPathComp      = explicit(2)
       mplsTunnelHopRowStatus          = active(1)
       mplsTunnelHopStorageType        = volatile(2)
     }
```

If the mplsTunnelCHopTable is used (and it might be used to supply information about path expansions) the contents will, for this example, be identical to the entries in the mplsTunnelHopTable since strict explicit routes were used.

The mplsTunnelARHopTable is used to expose the information reported in the Record Route object carried in the Resv message. In this example, there would also be four entries as shown below.

```
     {
       mplsTunnelARHopListIndex        = 12
       mplsTunnelARHopIndex            = 1
       mplsTunnelARHopAddrType         = ipv4(1)
       mplsTunnelARHopIpAddr           = "192.0.2.33"
     }

     {
       mplsTunnelARHopListIndex        = 12
       mplsTunnelARHopIndex            = 2
       mplsTunnelARHopAddrType         = ipv4(1)
       mplsTunnelARHopIpAddr           = "192.0.2.65"
     }
```

```
     {
       mplsTunnelARHopListIndex        = 197
       mplsTunnelARHopIndex            = 1
       mplsTunnelARHopAddrType         = ipv4(1)
       mplsTunnelARHopIpAddr           = "192.0.2.34"
     }

     {
       mplsTunnelARHopListIndex        = 197
       mplsTunnelARHopIndex            = 2
       mplsTunnelARHopAddrType         = ipv4(1)
       mplsTunnelARHopIpAddr           = "192.0.2.66"
     }
```

6. Managing P2MP MPLS-TE LSPs Through the LSR MIB Module

The nature of P2MP tunnels is such that an LSR that is crossed by a tunnel may either be the ingress of that tunnel or have precisely one upstream LSP segment (also known as an in-segment [RFC3812]) for that LSP. On the other hand, any LSR that is crossed by a tunnel may be an egress for that tunnel, have one or more downstream segments (also known as out-segments [RFC3812]) for that tunnel, or be both an egress and have one or more out-segments. Thus, for an LSP at an LSR there may be zero or one in-segments, and zero, one, or more than one out-segments.

In-segments, out-segments, and their relationship through cross-connections are modeled and managed in the MPLS-LSR-STD-MIB module [RFC3813]. The mplsInSegmentTable contains in-segments, and the mplsOutSegmentTable contains out-segments. The mplsXCTable maintains the relationships between in- and out-segments such that any many-to-many relationship is allowed. Each segment points into the mplsXCTable using mplsInSegmentXCIndex and mplsOutSegmentXCIndex.

The mplsXCTable contains a series of entries indexed by the primary mplsXCIndex object and subsidiary indexes mplsXCInSegmentIndex and mplsXCOutSegmentIndex. All cross-connect entries for the same P2MP LSP use the same value of mplsXCIndex, and this value is found in the mplsTeP2mpTunnelP2mpXcIndex object in mplsTeP2mpTunnelTable.

A single P2MP cross-connect has zero or one in-segment. At the ingress LSR, there is no in-segment and mplsXCInSegmentIndex is set to the single octet 0x00. At transit LSRs, there is exactly one in-segment and mplsXCInSegmentIndex is set to the value of mplsInSegmentIndex for the in-segment as it appears in the mplsInSegmentTable.

A single P2MP cross-connect has zero, one, or many out-segments. If

there is no out-segment (the cross-connect is on an egress LSR), there is one entry in the mplsXCTable indexed by mplsXCIndex set to mplsInSegmentXCIndex from the in-segment's entry in mplsInSegmentTable, mplsXCInSegmentIndex set to the value of mplsInSegmentIndex that identifies the in-segment in mplsInSegmentTable, and mplsXCOutSegmentIndex set to the single octet 0x00. This behavior is exactly as described in [RFC3813].

If there is exactly one out-segment (the cross-connect is on a transit LSR) then the behavior is also exactly as described in [RFC3813], and as well as the in-segment objects described in the previous paragraph, mplsXCOutSegmentIndex is set to the value of mplsOutSegmentIndex that identifies the out-segment in mplsOutSegmentTable. Note that mplsInSegmentXCIndex and mplsOutSegmentXCIndex from the relevant table entries will have the same value which will provide the value of mplsXCIndex for the cross-connect. This value is also found in the mplsTeP2mpTunnelP2mpXcIndex object in mplsTeP2mpTunnelTable.

If there is more than one out-segment then there is one entry in mplsXCTable table for each out-segment. The value of mplsXCIndex is consistent across all of these table entries and can be found in the mplsTeP2mpTunnelP2mpXcIndex object in mplsTeP2mpTunnelTable. The in-segment index (mplsXCInSegmentIndex) is also consistent identifying the single in-segment or (on the ingress LSR) containing the single octet 0x00. Each of these mplsXCTable entries contains a different mplsXCOutSegmentIndex value so that the table can easily be walked to find all of the out-segments for the same cross-connect.

Finally, if an LSR is an egress as well as a transit or branch for the P2MP LSP (we call this a bud LSR), mplsXCTable contains the entries described above in combination. That is, one entry will have mplsXCOutSegmentIndex set to the single octet 0x00, and other entries with the same value of mplsXCIndex and mplsXCInSegmentIndex will exist for each out-segment.

6.1. Example Use of the LSR MIB Module

This section demonstrates how the objects in MPLS-LSR-STD-MIB would be set for an example P2MP LSP cross-connect. The information here does not show how and in what order these objects should be set to create the cross-connect, but shows what information would be read if the tables were examined.

Figure 3 shows the LSP at the LSR that is being examined. There are three interfaces to LSR X: 10, 21 and 22. The LSP enters through interface 10 using label 7, and exits through interfaces 21 and 22 using labels 8 and 9 respectively. Let us assume that LSR X is also

an egress for the LSP.

```
                      -------
                     |       |21   Label 8
            Label 7  |       +-------------->
       --->----------+ LSR X |
                   10|       +-------------->
                     |       |22   Label 9
                      -------
```

Figure 3 : A P2MP LSP at a Branch LSR

In mplsInSegmentTable there is a single entry

```
   {
      mplsInSegmentIndex           = 0x00000015,
      mplsInSegmentLabel           = 7,   -- incoming label
      mplsInSegmentNPop            = 1,
      mplsInSegmentInterface       = 10,  -- incoming interface
      mplsInSegmentXCIndex         = 0x37 -- common index into XC table
   }
```

In mplsOutSegmentTable there are two entries.

```
   {
      mplsOutSegmentIndex          = 0x00000432,
      mplsOutSegmentPushTopLabel   = true(1),
      mplsOutSegmentTopLabel       = 8,   -- outgoing label
      mplsOutSegmentInterface      = 21,  -- outgoing interface
      mplsOutSegmentXCIndex        = 0x37 -- common index into XC table
   }

   {
      mplsOutSegmentIndex          = 0x00000017,
      mplsOutSegmentPushTopLabel   = true(1),
      mplsOutSegmentTopLabel       = 9,   -- outgoing label
      mplsOutSegmentInterface      = 22,  -- outgoing interface
      mplsOutSegmentXCIndex        = 0x37 -- common index into XC table
   }
```

In mplsXCTable there are three entries. The first two are for the cross-connections to the out-segments, and the third is for the local egress.

```
   {
      mplsXCIndex                = 0x37,      -- common index
      mplsXCInSegmentIndex       = 0x00000015,-- the in-segment
      mplsXCOutSegmentIndex      = 0x00000432,-- first out-segment
      mplsXCLspId                = 0x0102     -- unique LSP ID
      mplsXCLabelStackIndex      = 0x00,      -- only one outgoing label
   }
```

```
   {
      mplsXCIndex                = 0x37,      -- common index
      mplsXCInSegmentIndex       = 0x00000015,-- the in-segment
      mplsXCOutSegmentIndex      = 0x00000017,-- second out-segment
      mplsXCLspId                = 0x0102     -- unique LSP ID
      mplsXCLabelStackIndex      = 0x00,      -- only one outgoing label
   }

   {
      mplsXCIndex                = 0x37,      -- common index
      mplsXCInSegmentIndex       = 0x00000015,-- the in-segment
      mplsXCOutSegmentIndex      = 0x00,      -- no out-segment
      mplsXCLspId                = 0x0102     -- unique LSP ID
      mplsXCLabelStackIndex      = 0x00,      -- no other outgoing label
   }
```

7. MPLS Traffic Engineering P2MP MIB Definitions

This MIB module uses imports from [RFC2578], [RFC2580], [RFC2579], [RFC3811], [RFC3812], [RFC3813], [RFC3289], and [RFC4001].

```
   MPLS-TE-P2MP-STD-MIB DEFINITIONS ::= BEGIN

   IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
      Unsigned32, Counter32, Counter64, TimeTicks
         FROM SNMPv2-SMI                                -- RFC 2578
      MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
         FROM SNMPv2-CONF                               -- RFC 2580
      TruthValue, RowStatus, StorageType, TimeStamp
         FROM SNMPv2-TC                                 -- RFC 2579
      mplsStdMIB, MplsPathIndexOrZero
         FROM MPLS-TC-STD-MIB                           -- RFC 3811
      MplsIndexType
         FROM MPLS-LSR-STD-MIB                          -- RFC 3813
      mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
      mplsTunnelEgressLSRId
         FROM MPLS-TE-STD-MIB                           -- RFC 3812
      IndexInteger, IndexIntegerNextFree
         FROM DIFFSERV-MIB                              -- RFC 3289
      InetAddress, InetAddressType
         FROM INET-ADDRESS-MIB                          -- RFC 4001
      ;

   mplsTeP2mpStdMIB MODULE-IDENTITY
      LAST-UPDATED "200904170000Z"  -- April 17, 2009
      ORGANIZATION
         "Multiprotocol Label Switching (MPLS) Working Group"
      CONTACT-INFO
           "        Adrian Farrel
                    Old Dog Consulting
            Email:  adrian@olddog.co.uk

                    Seisho Yasukawa
                    NTT Corporation
            Email:  s.yasukawa@hco.ntt.co.jp

                    Thomas D. Nadeau
                    British Telecom
            Email:  tom.nadeau@bt.com

                   Comments about this document should be emailed
                   directly to the MPLS working group mailing list at
                   mpls@lists.ietf.org"
```

```
      DESCRIPTION
           "Copyright (c) 2009 IETF Trust and the persons identified as
            the document authors. All rights reserved.

            This document is subject to BCP 78 and the IETF Trust's
            Legal Provisions Relating to IETF Documents in effect on the
            date of publication of this document
            (http://trustee.ietf.org/license-info). Please review these
            documents carefully, as they describe your rights and
            restrictions with respect to this document.

            The initial version of this MIB module was published in
            RFC XXXX. For full legal notices see the RFC itself or see:
            http://www.ietf.org/copyrights/ianamib.html
-- RFC Editor. Please replace XXXX with the RFC number for this
-- document and remove this note.

            This MIB module contains managed object definitions
            for Point-to-Multipoint (P2MP) MPLS Traffic Engineering (TE)
            defined in:
            1. Signaling Requirements for Point-to-Multipoint
               Traffic-Engineered MPLS Label Switched Paths (LSPs),
               S. Yasukawa, RFC 4461, April 2006.
            2. Extensions to Resource Reservation Protocol - Traffic
               Engineering (RSVP-TE) for Point-to-Multipoint TE Label
               Switched Paths (LSPs), Aggarwal, R., Papadimitriou, D.,
               and Yasukawa, S., RFC 4875, May 2007."

      -- Revision history.

      REVISION
         "200904170000Z"  -- April 17, 2009
      DESCRIPTION
           "Initial version issued as part of RFC XXXX."
-- RFC Editor. Please replace XXXX with the RFC number for this
-- document and remove this note.

      ::= { mplsStdMIB YYY }

-- RFC Editor. Please replace YYY with the codepoint issued by IANA
-- and remove this note.
```

```
   -- Top level components of this MIB module.

   -- notifications
   mplsTeP2mpNotifications OBJECT IDENTIFIER ::= { mplsTeP2mpStdMIB 0 }
   -- tables, scalars
   mplsTeP2mpScalars       OBJECT IDENTIFIER ::= { mplsTeP2mpStdMIB 1 }
   mplsTeP2mpObjects       OBJECT IDENTIFIER ::= { mplsTeP2mpStdMIB 2 }
   -- conformance
   mplsTeP2mpConformance   OBJECT IDENTIFIER ::= { mplsTeP2mpStdMIB 3 }

   -- MPLS P2MP Tunnel scalars.

   mplsTeP2mpTunnelConfigured OBJECT-TYPE
      SYNTAX        Unsigned32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "The number of P2MP tunnels configured on this device. A
            tunnel is considered configured if the mplsTunnelRowStatus
            in MPLS-TE-STD-MIB is active(1)."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpScalars 1 }

   mplsTeP2mpTunnelActive OBJECT-TYPE
      SYNTAX        Unsigned32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "The number of P2MP tunnels active on this device. A
            tunnel is considered active if the mplsTunnelOperStatus
            in MPLS-TE-STD-MIB is up(1)."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpScalars 2 }

   mplsTeP2mpTunnelTotalMaxHops OBJECT-TYPE
      SYNTAX        Unsigned32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "The maximum number of hops that can be specified for an
            entire P2MP tunnel on this device. This object should be
            used in conjunction with mplsTunnelMaxHops in
```

```
            MPLS-TE-STD-MIB that is used in the context of P2MP tunnels
            to express the maximum number of hops to any individual
            destination of a P2MP tunnel that can be configured on this
            device. mplsTeP2mpTunnelTotalMaxHops would normally be set
            larger than or equal to mplsTunnelMaxHops."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpScalars 3 }

   -- End of MPLS Tunnel scalars.

   -- MPLS P2MP tunnel table.

   mplsTeP2mpTunnelTable OBJECT-TYPE
      SYNTAX        SEQUENCE OF MplsTeP2mpTunnelEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The mplsTeP2mpTunnelTable allows new P2MP MPLS tunnels to be
            created between an LSR and one or more remote end-points,
            and existing P2MP tunnels to be reconfigured or removed.

            This table sparse augments mplsTunnelTable in
            MPLS-TE-STD-MIB such that entries in that table can be
            flagged as point-to-multipoint, and can be configured and
            monitored appropriately."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpObjects 1 }

   mplsTeP2mpTunnelEntry OBJECT-TYPE
      SYNTAX        MplsTeP2mpTunnelEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "An entry in this table represents a P2MP MPLS tunnel.
            An entry can be created by a network administrator or by an
            SNMP agent as instructed by an MPLS signaling protocol.

            An entry in this table MUST correspond to an entry in the
            mplsTunnelTable in MPLS-TE-STD-MIB. This table shares index
            objects with that table and sparse augments that table.

            Thus, an entry in this table can only be created at the same
```

```
            time as or after a corresponding entry in mplsTunnelTable,
            and an entry in mplsTunnelTable cannot be deleted while a
            corresponding entry exists in this table.

            This table entry includes a row status object, but
            administrative and operational statuses should be taken from
            mplsTunnelAdminStatus and mplsTunnelOperStatus in the
            corresponding entry in mplsTunnelTable."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      INDEX {  mplsTunnelIndex,
               mplsTunnelInstance,
               mplsTunnelIngressLSRId,
               mplsTunnelEgressLSRId
            }
      ::= { mplsTeP2mpTunnelTable 1 }

   MplsTeP2mpTunnelEntry ::= SEQUENCE {
         mplsTeP2mpTunnelP2mpIntegrity      TruthValue,
         mplsTeP2mpTunnelBranchRole         INTEGER,
         mplsTeP2mpTunnelP2mpXcIndex        MplsIndexType,
         mplsTeP2mpTunnelRowStatus          RowStatus,
         mplsTeP2mpTunnelStorageType        StorageType
   }

   mplsTeP2mpTunnelP2mpIntegrity OBJECT-TYPE
      SYNTAX        TruthValue
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "Denotes whether or not P2MP Integrity is required for this
            tunnel.

            If P2MP integrity is operational on a P2MP tunnel then the
            failure of the path to any of the tunnel destinations should
            cause the teardown of the entire P2MP tunnel."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      DEFVAL { false }
      ::= { mplsTeP2mpTunnelEntry 2 }
```

```
   mplsTeP2mpTunnelBranchRole OBJECT-TYPE
      SYNTAX        INTEGER { notBranch(1),
                              branch(2),
                              bud(3) }
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "This value supplements the value in the object
            mplsTunnelRole in MPLS-TE-STD-MIB that indicates the role
            of this LSR in the tunnel represented by this entry in
            mplsTeP2mpTunnelTable.

            mplsTunnelRole may take any of the values:
               head(1),
               transit(2),
               tail(3),
               headTail(4)

            If this LSR is an ingress and there is exactly one
            out-segment, mplsTunnelRole should contain the value
            head(1), and mplsTeP2mpTunnelBranchRole should have the
            value notBranch(1).

            If this LSR is an ingress with more than one out segment,
            mplsTunnelRole should contain the value head(1), and
            mplsTeP2mpTunnelBranchRole should have the value branch(2).

            If this LSR is an ingress, an egress, and there is one or
            more out-segments, mplsTunnelRole should contain the value
            headTail(4), and mplsTeP2mpTunnelBranchRole should have the
            value bud(3).

            If this LSR is a transit with exactly one out-segment,
            mplsTunnelRole should contain the value transit(2), and
            mplsTeP2mpTunnelBranchRole should have the value
            notBranch(1).

            If this LSR is a transit with more than one out-segment,
            mplsTunnelRole should contain the value transit(2), and
            mplsTeP2mpTunnelBranchRole should have the value branch(2).

            If this LSR is a transit with one or more out-segments and
            is also an egress, mplsTunnelRole should contain the value
            transit(2), and mplsTeP2mpTunnelBranchRole should have the
            value bud(3).

            If this LSR is an egress with no out-segment and is not the
            ingress, mplsTunnelRole should contain the value tail(3),
```

```
            and mplsTeP2mpTunnelBranchRole should have the value
            notBranch(1).

            If this LSR is an egress and has one or more out-segments,
            mplsTunnelRole should contain the value transit(1), and
            mplsTeP2mpTunnelBranchRole should have the value bud(3)."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      DEFVAL { notBranch }
      ::= { mplsTeP2mpTunnelEntry 3 }

   mplsTeP2mpTunnelP2mpXcIndex OBJECT-TYPE
      SYNTAX        MplsIndexType
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This object contains the value of mplsXCIndex, the primary
            index of the mplsXCTable for all cross-connect entries for
            this P2MP LSP.

            If no XC entries have been created yet, this object must
            return zero.

            The set of entries in the mplsXCTable for this P2MP LSP can
            be walked by reading Get-or-GetNext starting with the three
            indexes to mplsXCTable set as:
              mplsXCIndex            = the value of this object
              mplsXCInSegmentIndex   = 0x0
              mplsXCOutSegmentIndex  = 0x0"
      REFERENCE
           "RFC 3813 - Multiprotocol Label Switching (MPLS) Label
            Switching (LSR) Router Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T.  Nadeau, June 2004."
      ::= { mplsTeP2mpTunnelEntry 4 }

   mplsTeP2mpTunnelRowStatus 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 is in active(1)
            state, no objects in that row can be modified by the agent
            except mplsTeP2mpTunnelRowStatus and
            mplsTeP2mpTunnelStorageType.
```

```
            This object and mplsTunnelRowStatus in the corresponding
            entry in mplsTunnelTable in MPLS-TE-STD-MIB should be
            managed together. No objects in a row in this table can be
            modified when the mplsTunnelRowStatus object in the
            corresponding row in mplsTunnelTable has value active(1).

            Note that no admin or oper status objects are provided in
            this table. The administrative and operational status of
            P2MP tunnels is taken from the values of
            mplsTunnelAdminStatus and mplsTunnelOperStatus in the
            corresponding row mplsTunnelTable."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpTunnelEntry 5 }

   mplsTeP2mpTunnelStorageType OBJECT-TYPE
      SYNTAX        StorageType
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "The storage type for this tunnel entry.
            Conceptual rows having the value 'permanent' need not allow
            write-access to any columnar objects in the row."
      DEFVAL { volatile }
      ::= { mplsTeP2mpTunnelEntry 6 }

   -- End of mplsTeP2mpTunnelTable

   -- MPLS P2MP tunnel destination table.

   mplsTeP2mpTunnelSubGroupIDNext OBJECT-TYPE
      SYNTAX        IndexIntegerNextFree (0..65535)
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This object contains an unused value for
            mplsTeP2mpTunnelDestSubGroupID, or a zero to indicate that
            none exists. Negative values are not allowed, as they do not
            correspond to valid values of
            mplsTeP2mpTunnelDestSubGroupID.

            Note that this object offers an unused value for an
            mplsTeP2mpTunnelDestSubGroupID value at the local LSR when
            it is a sub-group originator. In other cases, the value of
            mplsTeP2mpTunnelDestSubGroupID SHOULD be taken from the
            received value signaled by the signaling protocol and
```

```
            corresponds to the value in
            mplsTeP2mpTunnelDestSrcSubGroupID."
      ::= { mplsTeP2mpObjects 2 }

   mplsTeP2mpTunnelDestTable OBJECT-TYPE
      SYNTAX        SEQUENCE OF MplsTeP2mpTunnelDestEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The mplsTeP2mpTunnelDestTable allows new destinations of
            P2MP MPLS tunnels to be added to and removed from P2MP
            tunnels."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpObjects 3 }

   mplsTeP2mpTunnelDestEntry OBJECT-TYPE
      SYNTAX        MplsTeP2mpTunnelDestEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "An entry in this table represents a destination of a P2MP
            MPLS tunnel. An entry can be created by a network
            administrator or by an SNMP agent as instructed by an MPLS
            signaling protocol.

            Entries in this table share some index fields with the
            mplsTeP2mpTunnelTable and the mplsTunnelTable in
            MPLS-TE-STD-MIB. Entries in this table have no meaning
            unless there is a corresponding entry in
            mplsTeP2mpTunnelTable (which, itself, depends on a
            corresponding entry in mplsTunnelTable).

            Note that the same destination may be present more than once
            if it is in more than one sub-group as reflected by the
            mplsTeP2mpTunnelDestSrcSubGroupOriginType,
            mplsTeP2mpTunnelDestSrcSubGroupOrigin,
            mplsTeP2mpTunnelDestSrcSubGroupID,
            mplsTeP2mpTunnelDestSubGroupOriginType,
            mplsTeP2mpTunnelDestSubGroupOrigin, and
            mplsTeP2mpTunnelDestSubGroupID, index objects.

            Entries in this table may be created at any time. If created
            before an entry in the mplsTeP2mpTunnelTable the entries
            have no meaning, but may be kept ready for the creation of
            the P2MP tunnel. If created after the entry in
```

```
            mplsTeP2mpTunnelTable, entries in this table may reflect the
            addition of destinations to active P2MP tunnels. For this
            reason, entries in this table are equipped with row, admin,
            and oper status objects. "
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      INDEX {  mplsTunnelIndex,
               mplsTunnelInstance,
               mplsTunnelIngressLSRId,
               mplsTunnelEgressLSRId,
               mplsTeP2mpTunnelDestSrcSubGroupOriginType,
               mplsTeP2mpTunnelDestSrcSubGroupOrigin,
               mplsTeP2mpTunnelDestSrcSubGroupID,
               mplsTeP2mpTunnelDestSubGroupOriginType,
               mplsTeP2mpTunnelDestSubGroupOrigin,
               mplsTeP2mpTunnelDestSubGroupID,
               mplsTeP2mpTunnelDestDestinationType,
               mplsTeP2mpTunnelDestDestination
            }
      ::= { mplsTeP2mpTunnelDestTable 1 }

   MplsTeP2mpTunnelDestEntry ::= SEQUENCE {
         mplsTeP2mpTunnelDestSrcSubGroupOriginType  InetAddressType,
         mplsTeP2mpTunnelDestSrcSubGroupOrigin      InetAddress,
         mplsTeP2mpTunnelDestSrcSubGroupID          IndexInteger,
         mplsTeP2mpTunnelDestSubGroupOriginType     InetAddressType,
         mplsTeP2mpTunnelDestSubGroupOrigin         InetAddress,
         mplsTeP2mpTunnelDestSubGroupID             IndexInteger,
         mplsTeP2mpTunnelDestDestinationType        InetAddressType,
         mplsTeP2mpTunnelDestDestination            InetAddress,
         mplsTeP2mpTunnelDestBranchOutSegment       MplsIndexType,
         mplsTeP2mpTunnelDestHopTableIndex          MplsPathIndexOrZero,
         mplsTeP2mpTunnelDestPathInUse              MplsPathIndexOrZero,
         mplsTeP2mpTunnelDestCHopTableIndex         MplsPathIndexOrZero,
         mplsTeP2mpTunnelDestARHopTableIndex        MplsPathIndexOrZero,
         mplsTeP2mpTunnelDestTotalUpTime            TimeTicks,
         mplsTeP2mpTunnelDestInstanceUpTime         TimeTicks,
         mplsTeP2mpTunnelDestPathChanges            Counter32,
         mplsTeP2mpTunnelDestLastPathChange         TimeTicks,
         mplsTeP2mpTunnelDestCreationTime           TimeStamp,
         mplsTeP2mpTunnelDestStateTransitions       Counter32,
         mplsTeP2mpTunnelDestDiscontinuityTime      TimeStamp,
         mplsTeP2mpTunnelDestAdminStatus            INTEGER,
         mplsTeP2mpTunnelDestOperStatus             INTEGER,
         mplsTeP2mpTunnelDestRowStatus              RowStatus,
         mplsTeP2mpTunnelDestStorageType            StorageType
   }
   mplsTeP2mpTunnelDestSrcSubGroupOriginType OBJECT-TYPE
      SYNTAX        InetAddressType
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "This object identifies the type of address carried in
            mplsTeP2mpTunnelDestSrcSubGroupOrigin.

            Since the object mplsTeP2mpTunnelDestSrcSubGroupOrigin must
            conform to the protocol specification, this object must
            return either ipv4(1) or ipv6(2) at a transit or egress LSR.

            At an ingress LSR, there is no source sub-group and this
            object should return the value unknown(0)."
      ::= { mplsTeP2mpTunnelDestEntry 1 }

   mplsTeP2mpTunnelDestSrcSubGroupOrigin OBJECT-TYPE
      SYNTAX        InetAddress (SIZE(0|4|16))
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The TE Router ID (reachable and stable IP address) of the
            originator of the P2MP sub-group as received on a Path
            message by a transit or egress LSR.

            This object is interpreted in the context of
            mplsTeP2mpTunnelDestSrcSubGroupOriginType.

            The value of the sub-group originator used on outgoing Path
            messages is found in mplsTeP2mpTunnelDestSubGroupOrigin and
            is copied from this object unless this LSR is responsible
            for changing the sub-group ID.

            At an ingress LSR there is no received Path message.
            mplsTeP2mpTunnelDestSrcSubGroupOriginType should return
            unknown(0), and this object should return a zero-length
            string."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 2 }
```

```
   mplsTeP2mpTunnelDestSrcSubGroupID OBJECT-TYPE
      SYNTAX        IndexInteger (0..65535)
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The unique identifier assigned by the sub-group originator
            for this sub-group of this P2MP tunnel as received on a Path
            message by a transit or egress LSR.

            The value of the sub-group identifier used on outgoing Path
            messages is found in mplsTeP2mpTunnelDestSubGroupID and is
            copied from this object unless this LSR is responsible for
            changing the sub-group ID.

            At an ingress LSR there is no received Path message, and
            this object should return zero."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 3 }

   mplsTeP2mpTunnelDestSubGroupOriginType OBJECT-TYPE
      SYNTAX        InetAddressType
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "This object identifies the type of address carried in
            mplsTeP2mpTunnelDestSubGroupOrigin.

            This object must return either ipv4(1) or ipv6(2) in keeping
            with the protocol specification."
      ::= { mplsTeP2mpTunnelDestEntry 4 }

   mplsTeP2mpTunnelDestSubGroupOrigin OBJECT-TYPE
      SYNTAX        InetAddress (SIZE(4|16))
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The TE Router ID (reachable and stable IP address) of the
            originator of the P2MP sub-group. In many cases, this will
            be the ingress LSR of the P2MP tunnel and will be the
            received signaled value as available in
            mplsTeP2mpTunnelDestSrcSubGroupOrigin.

            When a signaling protocol is used, this object corresponds
            to the Sub-Group Originator field in the SENDER_TEMPLATE
```

```
            object.

            This object is interpreted in the context of
            mplsTeP2mpTunnelDestSubGroupOriginType."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 5 }

   mplsTeP2mpTunnelDestSubGroupID OBJECT-TYPE
      SYNTAX        IndexInteger (1..65535)
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "The unique identifier assigned by the sub-group originator
            for this sub-group of this P2MP tunnel.

            An appropriate value for this object during row creation
            when the sub-group origin in
            mplsTeP2mpTunnelDestSubGroupOrigin is the local LSR can
            be obtained by reading mplsTeP2mpTunnelSubGroupIDNext.

            At an egress, there is no downstream sub-group ID. This
            object should return the value received from upstream and
            reported in mplsTeP2mpTunnelDestSrcSubGroupID."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 6 }

   mplsTeP2mpTunnelDestDestinationType OBJECT-TYPE
      SYNTAX        InetAddressType
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "This object identifies the type of address carried in
            mplsTeP2mpTunnelDestDestination.

            This object forms part of the index of this table and can,
            therefore, not return the value unknown(0). Similarly, since
            the object mplsTeP2mpTunnelDestDestination must conform to
            the protocol specification, this object must return either
            ipv4(1) or ipv6(2)."
      ::= { mplsTeP2mpTunnelDestEntry 7 }
```

```
   mplsTeP2mpTunnelDestDestination OBJECT-TYPE
      SYNTAX        InetAddress (SIZE(4|16))
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "A single destination of this P2MP tunnel. That is, a
            routable TE address of a leaf. This will often be the TE
            Router ID of the leaf, but can be any interface address.

            When a signaling protocol is used, this object corresponds
            to the S2L Sub-LSP destination address field in the
            S2L_SUB_LSP object.

            This object is interpreted in the context of
            mplsTeP2mpTunnelDestDestinationType."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 8 }

   mplsTeP2mpTunnelDestBranchOutSegment OBJECT-TYPE
      SYNTAX        MplsIndexType
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This object identifies the outgoing branch from this LSR
            towards the destination represented by this table entry. It
            must be a unique identifier within the scope of this tunnel.

            If MPLS-LSR-STD-MIB is implemented, this object should
            contain an index into mplsOutSegmentTable.

            If MPLS-LSR-STD-MIB is not implemented, the LSR should
            assign a unique value to each branch of the tunnel.

            The value of this object is also used as an index into
            mplsTeP2mpTunnelBranchPerfTable."
         ::= { mplsTeP2mpTunnelDestEntry 9 }

   mplsTeP2mpTunnelDestHopTableIndex OBJECT-TYPE
      SYNTAX        MplsPathIndexOrZero
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "Index into the mplsTunnelHopTable entry that specifies the
            explicit route hops for this destination of the P2MP tunnel.
```

```
            This object represents the configured route for the branch
            of the P2MP tree to this destination and is meaningful only
            at the head-end (ingress or root) of the P2MP tunnel. Note
            that many such paths may be configured within the
            mplsTunnelHopTable for each destination, and that the object
            mplsTeP2mpTunnelDestPathInUse identifies which path has been
            selected for use."
      DEFVAL { 0 }
      ::= { mplsTeP2mpTunnelDestEntry 10 }

   mplsTeP2mpTunnelDestPathInUse OBJECT-TYPE
      SYNTAX        MplsPathIndexOrZero
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "This value denotes the configured path that was chosen as
            the explicit path to this destination of this P2MP tunnel.
            This value reflects the secondary index into
            mplsTunnelHopTable where the primary index comes from
            mplsTeP2mpTunnelDestHopTableIndex.

            The path indicated by this object might not exactly match
            the one signaled and recorded in mplsTunnelCHopTable as
            specific details of the path might be computed locally.

            Similarly, the path might not match the actual path in use
            as recorded in mplsTunnelARHopTable due to the fact that
            some details of the path may have been resolved within the
            network.

            A value of zero denotes that no path is currently in use or
            available."
      DEFVAL { 0 }
      ::= { mplsTeP2mpTunnelDestEntry 11 }

   mplsTeP2mpTunnelDestCHopTableIndex OBJECT-TYPE
      SYNTAX        MplsPathIndexOrZero
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Index into the mplsTunnelCHopTable that identifies the
            explicit path for this destination of the P2MP tunnel.

            This path is based on the chosen configured path identified
            by mplsTeP2mpTunnelDestHopTableIndex and
            mplsTeP2mpTunnelDestPathInUse, but may have been modified
            and automatically updated by the agent when computed hops
            become available or when computed hops get modified.
```

```
            If this destination is the destination of the 'first S2L
            sub-LSP' then this path will be signaled in the Explicit
            Route Object. If this destination is the destination of a
            'subsequent S2L sub-LSP' then this path will be signaled in
            a Secondary Explicit Route Object."
      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 12 }

   mplsTeP2mpTunnelDestARHopTableIndex OBJECT-TYPE
      SYNTAX        MplsPathIndexOrZero
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Index into the mplsTunnelARHopTable that identifies the
            actual hops traversed to this destination of the P2MP
            tunnel. This is automatically updated by the agent when the
            actual hops becomes available.

            If this destination is the destination of the 'first S2L
            sub-LSP' then this path will be signaled in the Recorded
            Route Object. If this destination is the destination of a
            'subsequent S2L sub-LSP' then this path will be signaled in
            a Secondary Recorded Route Object."

      REFERENCE
           "RFC 4875 - Extensions to Resource Reservation Protocol -
            Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
            Label Switched Paths (LSPs), R. Aggarwal, D. Papadimitriou,
            and S. Yasukawa, May 2007."
      ::= { mplsTeP2mpTunnelDestEntry 13 }

   mplsTeP2mpTunnelDestTotalUpTime OBJECT-TYPE
      SYNTAX        TimeTicks
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This value represents the aggregate up time for all
            instances of this tunnel to this destination, if this
            information is available.

            If this information is not available, this object MUST
            return a value of 0."
      ::= { mplsTeP2mpTunnelDestEntry 14 }
```

```
   mplsTeP2mpTunnelDestInstanceUpTime OBJECT-TYPE
      SYNTAX        TimeTicks
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This value identifies the total time that the currently
            active tunnel instance to this destination has had its
            operational status (mplsTeP2mpTunnelDestOperStatus) set to
            up(1) since it was last previously not up(1)."
      ::= { mplsTeP2mpTunnelDestEntry 15 }

   mplsTeP2mpTunnelDestPathChanges OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This object counts the number of times the actual path for
            this destination of this P2MP tunnel instance has changed.
            This object should be read in conjunction with
            mplsTeP2mpTunnelDestDiscontinuityTime."
      ::= { mplsTeP2mpTunnelDestEntry 16 }

   mplsTeP2mpTunnelDestLastPathChange OBJECT-TYPE
      SYNTAX        TimeTicks
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Specifies the time since the last change to the actual path
           for this destination of this P2MP tunnel instance."
      ::= { mplsTeP2mpTunnelDestEntry 17 }

   mplsTeP2mpTunnelDestCreationTime OBJECT-TYPE
      SYNTAX        TimeStamp
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Specifies the value of sysUpTime when the first instance of
            this tunnel came into existence for this destination. That
            is, when the value of mplsTeP2mpTunnelDestOperStatus was
            first set to up(1)."
      ::= { mplsTeP2mpTunnelDestEntry 18 }

   mplsTeP2mpTunnelDestStateTransitions OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "This object counts the number of times the status
```

```
            (mplsTeP2mpTunnelDestOperStatus) of this tunnel instance to
            this destination has changed.
            This object should be read in conjunction with
            mplsTeP2mpTunnelDestDiscontinuityTime."
      ::= { mplsTeP2mpTunnelDestEntry 19 }

   mplsTeP2mpTunnelDestDiscontinuityTime 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 row's Counter32 objects experienced
            a discontinuity. If no such discontinuity has occurred since
            the last re-initialization of the local management
            subsystem, then this object contains a zero value."
      ::= { mplsTeP2mpTunnelDestEntry 20 }

   mplsTeP2mpTunnelDestAdminStatus OBJECT-TYPE
      SYNTAX       INTEGER {
                      up(1),        -- ready to pass data
                      down(2),      -- out of service
                      testing(3)    -- in some test mode
                    }
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "Indicates the desired operational status of this
            destination of this P2MP tunnel."
      DEFVAL { up }
      ::= { mplsTeP2mpTunnelDestEntry 21 }

   mplsTeP2mpTunnelDestOperStatus OBJECT-TYPE
      SYNTAX        INTEGER {
                      up(1),             -- ready to pass data
                      down(2),           -- out of service
                      testing(3),        -- in some test mode
                      unknown(4),        -- status cannot be determined
                      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 destination
            of this P2MP tunnel. This object may be compared to
            mplsTunnelOperStatus that includes two other values:
              dormant(5)    -- some component is missing
```

```
              notPresent(6) -- down due to the state of
                            -- lower layer interfaces.
            These states do not apply to an individual destination of a
            P2MP MPLS-TE LSP and so are not included in this object."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpTunnelDestEntry 22 }

   mplsTeP2mpTunnelDestRowStatus OBJECT-TYPE
      SYNTAX        RowStatus
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION
           "This object is used to create, modify, and/or delete a row
            in this table. When a row in this table is in active(1)
            state, no objects in that row can be modified by SET
            operations except mplsTeP2mpTunnelDestAdminStatus and
            mplsTeP2mpTunnelDestStorageType."
      ::= { mplsTeP2mpTunnelDestEntry 23 }

   mplsTeP2mpTunnelDestStorageType OBJECT-TYPE
      SYNTAX        StorageType
      MAX-ACCESS    read-create
      STATUS        current
      DESCRIPTION  "The storage type for this table entry.

                    Conceptual rows having the value 'permanent' need
                    not allow write-access to any columnar objects in
                    the row."
      DEFVAL { volatile }
      ::= { mplsTeP2mpTunnelDestEntry 24 }

   -- End of mplsTeP2mpTunnelDestTable

   -- MPLS Tunnel Branch Performance Table.

   mplsTeP2mpTunnelBranchPerfTable  OBJECT-TYPE
      SYNTAX        SEQUENCE OF MplsTeP2mpTunnelBranchPerfEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "This table provides per-tunnel branch MPLS performance
            information.

            This table is not valid for switching types other than
            packet."
```

```
      ::= { mplsTeP2mpObjects 4 }

   mplsTeP2mpTunnelBranchPerfEntry OBJECT-TYPE
      SYNTAX        MplsTeP2mpTunnelBranchPerfEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "An entry in this table is created by the LSR for each
            downstream branch (out-segment) from this LSR for this P2MP
            tunnel.

            More than one destination as represented by an entry in the
            mplsTeP2mpTunnelDestTable may be reached through a single
            out-segment. More than one out-segment may belong to a
            single P2MP tunnel represented by an entry in
            mplsTeP2mpTunnelTable.

            Each entry in the table is indexed by the four identifiers
            of the P2MP tunnel, and the out-segment that identifies the
            outgoing branch."
      INDEX {  mplsTunnelIndex,
               mplsTunnelInstance,
               mplsTunnelIngressLSRId,
               mplsTunnelEgressLSRId,
               mplsTeP2mpTunnelBranchPerfBranch
            }
      ::= { mplsTeP2mpTunnelBranchPerfTable 1 }

   MplsTeP2mpTunnelBranchPerfEntry ::= SEQUENCE {
         mplsTeP2mpTunnelBranchPerfBranch        MplsIndexType,
         mplsTeP2mpTunnelBranchPerfPackets       Counter32,
         mplsTeP2mpTunnelBranchPerfHCPackets     Counter64,
         mplsTeP2mpTunnelBranchPerfErrors        Counter32,
         mplsTeP2mpTunnelBranchPerfBytes         Counter32,
         mplsTeP2mpTunnelBranchPerfHCBytes       Counter64,
         mplsTeP2mpTunnelBranchDiscontinuityTime TimeStamp
   }

   mplsTeP2mpTunnelBranchPerfBranch OBJECT-TYPE
      SYNTAX        MplsIndexType
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
           "This object identifies an outgoing branch from this LSR
            for this tunnel. Its value is unique within the context of
            the tunnel.

            If MPLS-LSR-STD-MIB is implemented, this object should
```

```
            contain an index into mplsOutSegmentTable.

            Under all circumstances, this object should contain
            the same value as mplsTeP2mpTunnelDestBranchOutSegment for
            destinations reached on this branch."
         ::= { mplsTeP2mpTunnelBranchPerfEntry 1 }

   mplsTeP2mpTunnelBranchPerfPackets OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Number of packets forwarded by the tunnel onto this branch.
            This object should represents the 32-bit value of the least
            significant part of the 64-bit value if both
            mplsTeP2mpTunnelBranchPerfHCPackets is returned.
            This object should be read in conjunction with
            mplsTeP2mpTunnelBranchDiscontinuityTime."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 2 }

   mplsTeP2mpTunnelBranchPerfHCPackets OBJECT-TYPE
      SYNTAX        Counter64
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "High capacity counter for number of packets forwarded by the
            tunnel onto this branch.
            This object should be read in conjunction with
            mplsTeP2mpTunnelBranchDiscontinuityTime."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 3 }

   mplsTeP2mpTunnelBranchPerfErrors OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Number of packets dropped because of errors or for other
            reasons, that were supposed to be forwarded onto this
            branch for this tunnel. This object should be read in
            conjunction with mplsTeP2mpTunnelBranchDiscontinuityTime."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 4 }

   mplsTeP2mpTunnelBranchPerfBytes OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "Number of bytes forwarded by the tunnel onto this branch.
```

```
            This object should represents the 32-bit value of the least
            significant part of the 64-bit value if both
            mplsTeP2mpTunnelBranchPerfHCBytes is returned.
            This object should be read in conjunction with
            mplsTeP2mpTunnelBranchDiscontinuityTime."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 5 }

   mplsTeP2mpTunnelBranchPerfHCBytes OBJECT-TYPE
      SYNTAX        Counter64
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
           "High capacity counter for number of bytes forwarded
            by the tunnel onto this branch.
            This object should be read in conjunction with
            mplsTeP2mpTunnelBranchDiscontinuityTime."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 6 }

   mplsTeP2mpTunnelBranchDiscontinuityTime 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 row's Counter32 or Counter64 objects
            experienced a discontinuity. If no such discontinuity has
            occurred since the last re-initialization of the local
            management subsystem, then this object contains a zero
            value."
      ::= { mplsTeP2mpTunnelBranchPerfEntry 7 }

   -- End of mplsTeP2mpTunnelBranchPerfTable

   -- Notifications.

   mplsTeP2mpTunnelNotificationEnable OBJECT-TYPE
      SYNTAX        TruthValue
      MAX-ACCESS    read-write
      STATUS        current
      DESCRIPTION
           "If this object is true(1), then it enables the generation of
            mplsTeP2mpTunnelDestUp and mplsTeP2mpTunnelDestDown
            notifications. Otherwise these notifications are not
            emitted.

            Note that when tunnels have large numbers of destinations,
            setting this object to true(1) may result in the generation
            of large numbers of notifications."
```

```
      DEFVAL { false }
      ::= { mplsTeP2mpObjects 5 }

   mplsTeP2mpTunnelDestUp NOTIFICATION-TYPE
      OBJECTS     {
         mplsTeP2mpTunnelDestAdminStatus,
         mplsTeP2mpTunnelDestOperStatus
      }
      STATUS      current
      DESCRIPTION
           "This notification is generated when a
            mplsTeP2mpTunnelDestOperStatus object for one of the
            destinations of one of the configured tunnels is about to
            leave the down(2) state and transition into some other
            state.  This other state is indicated by the included value
            of mplsTeP2mpTunnelDestOperStatus.

            This reporting of state transitions mirrors mplsTunnelUp."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpNotifications 1 }

   mplsTeP2mpTunnelDestDown NOTIFICATION-TYPE
      OBJECTS     {
         mplsTeP2mpTunnelDestAdminStatus,
         mplsTeP2mpTunnelDestOperStatus
      }
      STATUS      current
      DESCRIPTION
           "This notification is generated when a
            mplsTeP2mpTunnelDestOperStatus object for one of the
            destinations of one of the configured tunnels is about to
            enter the down(2) state from some other state. This other
            state is indicated by the included value of
            mplsTeP2mpTunnelDestOperStatus.

            This reporting of state transitions mirrors mplsTunnelDown."
      REFERENCE
           "RFC 3812 - Multiprotocol Label Switching (MPLS) Traffic
            Engineering (TE) Management Information Base (MIB),
            Srinivasan, C., Viswanathan, A., and T. Nadeau, June 2004."
      ::= { mplsTeP2mpNotifications 2 }

   -- End of notifications.
```

```
   --******************************************************************
   -- Module Conformance Statement
   --******************************************************************

   mplsTeP2mpGroups
      OBJECT IDENTIFIER ::= { mplsTeP2mpConformance 1 }

   mplsTeP2mpCompliances
      OBJECT IDENTIFIER ::= { mplsTeP2mpConformance 2 }

   --
   -- Full Compliance
   -- Compliance requirement for fully compliant implementations.
   -- Such implementations allow configuration of P2MP tunnels at
   -- head-end LSRs via SNMP, and monitoring of P2MP tunnels at all
   -- LSRs via SNMP.
   --

   mplsTeP2mpModuleFullCompliance MODULE-COMPLIANCE
      STATUS       current
      DESCRIPTION
           "Compliance statement for agents that provide full support
            for MPLS-TE-P2MP-STD-MIB. Such devices can be monitored and
            also be configured using this MIB module.
            The Module is implemented with support for read-create and
            read-write. In other words, both monitoring and
            configuration are available when using this
            MODULE-COMPLIANCE."

      MODULE -- this module

      MANDATORY-GROUPS {
         mplsTeP2mpGeneralGroup,
         mplsTeP2mpNotifGroup,
         mplsTeP2mpScalarGroup
      }

      -- mplsTeP2mpTunnelTable

      OBJECT       mplsTeP2mpTunnelRowStatus
      SYNTAX       RowStatus { active(1) }
      WRITE-SYNTAX RowStatus { createAndGo(4), destroy(6) }
      DESCRIPTION
          "Support for createAndWait and notInService is not
           required."

   ::= { mplsTeP2mpCompliances 1 }
```

```
   mplsTeP2mpModuleReadOnlyCompliance MODULE-COMPLIANCE
      STATUS       current
      DESCRIPTION
           "Compliance statement for agents that provide read-only
            support for MPLS-TE-P2MP-STD-MIB. Such devices can only be
            monitored using this MIB module.
            The Module is implemented with support for read-only. In
            other words, only monitoring is available by implementing
            this MODULE-COMPLIANCE."

      MODULE -- this module

      MANDATORY-GROUPS {
         mplsTeP2mpGeneralGroup,
         mplsTeP2mpScalarGroup,
         mplsTeP2mpNotifGroup
      }

      -- mplsTeP2mpTunnelTable

      OBJECT      mplsTeP2mpTunnelP2mpIntegrity
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      OBJECT      mplsTeP2mpTunnelBranchRole
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      OBJECT      mplsTeP2mpTunnelP2mpXcIndex
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      OBJECT      mplsTeP2mpTunnelRowStatus
      SYNTAX      RowStatus { active(1) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required, and active(1) is the
           only status that needs to be supported."

      OBJECT      mplsTeP2mpTunnelStorageType
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      -- mplsTeP2mpTunnelDestTable

      OBJECT      mplsTeP2mpTunnelDestHopTableIndex
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."
```

```
      OBJECT      mplsTeP2mpTunnelDestPathInUse
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      OBJECT      mplsTeP2mpTunnelDestAdminStatus
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

      OBJECT      mplsTeP2mpTunnelDestRowStatus
      SYNTAX      RowStatus { active(1) }
      MIN-ACCESS  read-only
      DESCRIPTION
          "Write access is not required, and active(1) is the
           only status that needs to be supported."

      OBJECT      mplsTeP2mpTunnelDestStorageType
      MIN-ACCESS  read-only
      DESCRIPTION "Write access is not required."

   ::= { mplsTeP2mpCompliances 2 }

   -- Units of conformance.

   mplsTeP2mpGeneralGroup OBJECT-GROUP
      OBJECTS {
         mplsTeP2mpTunnelConfigured,
         mplsTeP2mpTunnelActive,
         mplsTeP2mpTunnelTotalMaxHops,
         mplsTeP2mpTunnelP2mpIntegrity,
         mplsTeP2mpTunnelBranchRole,
         mplsTeP2mpTunnelP2mpXcIndex,
         mplsTeP2mpTunnelRowStatus,
         mplsTeP2mpTunnelStorageType,
         mplsTeP2mpTunnelSubGroupIDNext,
         mplsTeP2mpTunnelDestSrcSubGroupOriginType,
         mplsTeP2mpTunnelDestSrcSubGroupOrigin,
         mplsTeP2mpTunnelDestSrcSubGroupID,
         mplsTeP2mpTunnelDestSubGroupOriginType,
         mplsTeP2mpTunnelDestSubGroupOrigin,
         mplsTeP2mpTunnelDestSubGroupID,
         mplsTeP2mpTunnelDestDestinationType,
         mplsTeP2mpTunnelDestDestination,
         mplsTeP2mpTunnelDestBranchOutSegment,
         mplsTeP2mpTunnelDestHopTableIndex,
         mplsTeP2mpTunnelDestPathInUse,
         mplsTeP2mpTunnelDestCHopTableIndex,
         mplsTeP2mpTunnelDestARHopTableIndex,
         mplsTeP2mpTunnelDestTotalUpTime,
```

```
         mplsTeP2mpTunnelDestInstanceUpTime,
         mplsTeP2mpTunnelDestPathChanges,
         mplsTeP2mpTunnelDestLastPathChange,
         mplsTeP2mpTunnelDestCreationTime,
         mplsTeP2mpTunnelDestStateTransitions,
         mplsTeP2mpTunnelDestDiscontinuityTime,
         mplsTeP2mpTunnelDestAdminStatus,
         mplsTeP2mpTunnelDestOperStatus,
         mplsTeP2mpTunnelDestRowStatus,
         mplsTeP2mpTunnelDestStorageType,
         mplsTeP2mpTunnelBranchPerfPackets,
         mplsTeP2mpTunnelBranchPerfHCPackets,
         mplsTeP2mpTunnelBranchPerfErrors,
         mplsTeP2mpTunnelBranchPerfBytes,
         mplsTeP2mpTunnelBranchPerfHCBytes,
         mplsTeP2mpTunnelBranchDiscontinuityTime,
         mplsTeP2mpTunnelNotificationEnable
      }
   STATUS  current
   DESCRIPTION
        "Collection of objects needed for MPLS P2MP."
   ::= { mplsTeP2mpGroups 1 }

   mplsTeP2mpNotifGroup NOTIFICATION-GROUP
      NOTIFICATIONS {
         mplsTeP2mpTunnelDestUp,
         mplsTeP2mpTunnelDestDown
      }
   STATUS  current
   DESCRIPTION
        "Notifications implemented in this module."
   ::= { mplsTeP2mpGroups 2 }

   mplsTeP2mpScalarGroup OBJECT-GROUP
      OBJECTS {
         mplsTeP2mpTunnelConfigured,
         mplsTeP2mpTunnelActive,
         mplsTeP2mpTunnelTotalMaxHops
      }
      STATUS  current
      DESCRIPTION
           "Scalar objects needed to implement P2MP MPLS tunnels."
      ::= { mplsTeP2mpGroups 3 }

   END
```

8. Security Considerations

It is clear that this MIB module is potentially useful for the monitoring of P2MP MPLS TE tunnels. This MIB module can also be used for the 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 mplsTeP2mpTunnelTable and mplsTeP2mpTunnelDestTable contain objects that can be used to provision P2MP MPLS tunnels, the destinations of those tunnels, and the hops that those tunnels take through the network. Unauthorized access to objects in these tables could result in disruption of traffic in the network. This is especially true if a tunnel has already been established.

  The use of stronger mechanisms, such as SNMPv3 security, should be considered where possible. Specifically, the SNMPv3 View-based Access Control Model (VACM) and User-based Security Model (USM) MUST be used with any v3 agent which implements this MIB module. Administrators SHOULD also consider whether read access to these objects is allowed, since read access may be undesirable under certain circumstances as described below.

- The use of this MIB module depends on the use of certain objects within MPLS-TE-STD-MIB defined in [RFC3812]. Note that those objects are also subject to the same security considerations, and any vulnerability to those objects could compromise the P2MP MPLS tunnels and/or data in the network. The security section of [RFC3812] MUST be applied in conjunction with this security section.

- This MIB module does not depend on MPLS-LSR-STD-MIB, but may be used in conjunction with that MIB module. If MPLS-LSR-STD-MIB is implemented on an LSR, then access to its objects can compromise any P2MP MPLS tunnels that start or end on, or transit the LSR. MPLS-LSR-STD-MIB is defined in [RFC3813] which has its own security section that MUST be applied in conjunction with this security section if both MIB modules are supported.

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 even to 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 mplsTeP2mpTunnelTable, mplsTeP2mpTunnelDestTable, and mplsTeP2mpTunnelBranchPerfTable collectively show information about the P2MP MPLS tunnel, its route through the network, and its performance characteristics. Knowledge of this information could be used to compromise the network, or simply to breach confidentiality. If an Administrator does not want to reveal this information, these tables should be considered sensitive/vulnerable.

- The objects in MPLS-TE-STD-MIB also provide information about the P2MP MPLS tunnels defined in this MIB module. If an Administrator does not want to reveal this information, the security section of [RFC3812] should be applied.

- The objects in MPLS-LSR-STD-MIB, if implemented, may also provide information about the P2MP MPLS tunnels present at an LSR, especially the label swapping and cross-connect operations. If an Administrator does not want to reveal this information, the security section of [RFC3813] should be applied.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is still 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 that SNMPv3 be deployed and cryptographic security enabled. 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 only those principals (users) that have legitimate rights to those objects.

9. Acknowledgments

The authors wish to thank Tom Petch, Ben Niven-Jenkins, Markus Stenberg, and Subodh Kumar for their input to this work. Thanks to Zafar Ali for discussions. Nic Neate provided very many helpful suggestions. Thanks to Francis Dupont for a detailed GenArt review.

Joan Cucchiara provided a very thorough and helpful early MIB Doctor review which caught a lot of issues.

10. IANA Considerations

As requested in MPLS-TC-STD-MIB [RFC3811], MPLS-related standards track MIB modules should be rooted under the mplsStdMIB subtree.

There is one new MPLS MIB module contained in this document, and the following "IANA Considerations" subsection requests IANA for a new assignment under the mplsStdMIB subtree.

New assignments can only be made via a Standards Action as specified in [RFC5226].

10.1. IANA Considerations for MPLS-TE-P2MP-STD-MIB

IANA is requested to assign an oid under the mplsStdMIB subtree to the MPLS-TE-P2MP-STD-MIB module specified in this document.

-- RFC Editor. Please see the marker YYY in this document and replace it
-- with the value assigned by IANA.
-- Please remove this note.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.

[RFC3811] Nadeau, T. and J. Cucchiara, "Definition of Textual Conventions and for Multiprotocol Label Switching (MPLS) Management", RFC 3811, June 2004.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

[RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching (LSR) Router Management Information Base (MIB)", RFC 3813, June 2004.

[RFC3945] Mannie, E., Ed., "Generalized Multiprotocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "TextualConventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC4875] Aggarwal, R., Papadimitriou, D., and Yasukawa, S., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

11.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statement for Internet Standard Management Framework", RFC 3410, December 2002.

[RFC4221] Nadeau, T., Srinivasan, C., and A. Farrel, "Multiprotocol Label Switching (MPLS) Management Overview", RFC 4221, November 2005.

[RFC4461] S. Yasukawa, Editor "Signaling Requirements for Point-to-Multipoint Traffic Engineered MPLS LSPs", RFC 4461, April 2006.

[RFC4802] Nadeau, T. and A. Farrel, "Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base", RFC 4802, February 2007.

[RFC4803] Nadeau, T. and A. Farrel, "Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) Management Information Base", RFC 4803, February 2007.

[RFC5226] Narten, T. and H. Alvestrand., "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

12. Authors' Addresses

Adrian Farrel
Old Dog Consulting
Email: adrian@olddog.co.uk

Seisho Yasukawa
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-Shi, Tokyo 180-8585 Japan
Phone: +81 422 59 4769
EMail: s.yasukawa@hco.ntt.co.jp

Thomas D. Nadeau
BT
BT Centre
81 Newgate Street
EC1A 7AJ
London, UK
Email: tom.nadeau@bt.com

13. Intellectual Property

The IETF Trust 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 any IETF 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.

Copies of Intellectual Property 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 any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions.

For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

14. Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST 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 THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

15. Full Copyright Statement