Network Working Group
INTERNET-DRAFT
Intended Status: Standards Track
Expires: June 21, 2015

M.Venkatesan
Dell Inc.
Kannan KV Sampath
Redeem
Sam K. Aldrin
Huawei Technologies
Thomas D. Nadeau
Brocade

December 18, 2014

MPLS-TP Traffic Engineering (TE) Management Information Base (MIB) draft-ietf-mpls-tp-te-mib-11.txt

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 additional managed objects and textual conventions for Tunnels, Identifiers and Label Switching Router to support Multiprotocol Label Switching (MPLS) MIB modules for transport networks.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on June 21, 2015.

Table of Contents

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 additional textual conventions and managed objects for Tunnels, Identifiers and Label Switching Router to support Multiprotocol Label Switching (MPLS) MIB modules for transport networks. MIB modules defined in this document extend the existing MPLS MIB objects in such a way that they support MPLS-TP but also other MPLS networks as well. Hence, the MPLS-TP name is not included in the MIB module name.

As described in the MPLS Traffic Engineering (TE) Management Information Base (MIB) definition [RFC3812], MPLS traffic engineering is concerned with the creation and management of MPLS tunnels. This term is a shorthand for a combination of one or more LSPs linking an ingress and an egress LSR. Several types of point-to-point MPLS tunnels may be constructed between a pair of LSRs A and B:

- Unidirectional with a single LSP (say) from A to B.

- Associated bidirectional consisting of two separately routed LSPs, one linking A to B and the other linking B to A. Together the pair provide a single logical bidirectional transport path.

- Co-routed bidirectional consisting of an associated bidirectional tunnel but with the second LSP from B to A following the reverse of the path of the LSP from A to B, in terms of both nodes and links.

Tunnels may be either statically configured by management action or dynamically created using a LSP management protocol.

The existing MPLS TE MIB [RFC3812] and the Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base [RFC4802] address only a subset of the combinations of statically and dynamically configured tunnel types, catering for statically configured unidirectional tunnels together with dynamically configured unidirectional and co-routed bidirectional tunnels. They are also restricted to two end point LSRs identified by IP addresses.

The MPLS-TP TE MIB defined in this document extends the MIB modules defined in [RFC3812] to cover all six combinations (that is adding support for statically configured associated and co-routed bidirectional plus dynamically configured associated bidirectional tunnels). It also extends support to end points that are identified other than with IP addresses.

This support is provided by a suite of four MIB modules that are to be used in conjunction with the MIB modules defined in [RFC3812] and the companion document [RFC3813] for MPLS Transport Profile (MPLS-TP) tunnel management.

At the time of writing, SNMP SET is no longer recommended as a way to configure MPLS networks as was described in [RFC3812]. However, since the MIB modules specified in this document extend and are intended to work in parallel with the MIB modules for MPLS specified in [RFC3812], certain objects defined here are specified with MAX-ACCESS of read-write or read-create so that specifications of the base tables in [RFC3812] and the extensions in this document are consistent. Although the examples described in Section 9 specify means to configure MPLS-TP tunnels in a similar way to the examples in [RFC3812], this should be seen as indicating how the MIB values would be returned in the specified circumstances having been configured by alternative means.

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 [RFC2578], STD 58 [RFC2579] and STD 58 [RFC2580].

3. Overview

3.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3.2. Terminology

This document uses terminology from the Multiprotocol Label Switching Architecture [RFC3031], Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) [RFC3812], Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB) [RFC3813] and MPLS Transport Profile

(MPLS-TP) Identifiers [RFC6370].

3.3. Acronyms

CC: Country Code
ICC: ITU Carrier Code
LSP: Label Switching Path
LSR: Label Switching Router
MPLS-TP: MPLS Transport Profile
TE: Traffic Engineering
TP: Transport Profile

4. Motivations

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) [RFC3812] provides support for Traffic Engineering tunnels. In MPLS, the actual transport of packets is provided by Label Switched Paths (LSPs). A transport service may be composed of multiple LSPs. In order to clearly identify the MPLS-TP service, as defined in [RFC6370], we use the term "MPLS-TP Tunnel" or simply "tunnel". However, with MPLS-TP, the characteristics of the tunnels were enhanced. For example, MPLS-TP tunnels, are bidirectional in nature and could be used with non-IP identifiers for the tunnel end points. As the existing MPLS-TE-STD-MIB and GMPLS-TE-STD-MIB were defined mainly to support unidirectional tunnels and signaled co-routed bidirectional tunnel definitions respectively, these existing MIB modules are not sufficient to capture all the characteristics of the tunnels. Hence, enhancing the MIB modules to support MPLS-TP tunnels is required. As most of the attributes of MPLS Traffic Engineering tunnels are also applicable to MPLS-TP tunnels, it is optimal to re-use and extend the existing MIB module definition instead of defining a new MIB module.

This document defines four additional MIB modules, namely MPLS-TE-EXT-STD-MIB, MPLS-TC-EXT-STD-MIB, MPLS-ID-STD-MIB and MPLS-LSR-EXT-STD-MIB. As these additional MIB modules are required for MPLS-TP functionality, these are all defined in this document, instead of being documented separately.

5. Feature List

The MIBs in this document satisfy the following requirements and constraints:

The MIB modules, taken together, support statically configured and dynamically signaled point-to-point, co-routed bidirectional and associated bidirectional tunnels.

- The MPLS tunnels need not be interfaces, but it is possible to configure an MPLS TP tunnel as an interface. Same ifType 150,

as defined in section 8 of [RFC3812], will be used for MPLS-TP tunnels as well.

- The mplsTunnelTable [RFC3812] is also to be used for MPLS-TP tunnels.

- New MPLS-TP specific textual conventions and identifiers are required.

- The mplsTunnelTable is sparsely extended to support MPLS-TP tunnel specific objects.

- A node configuration table (mplsTunnelExtNodeConfigTable), as detailed in the below section 6.1.2, is used to translate the Global_ID::Node_ID or ICC_Operator_ID::Node_ID to the local identifier in order to index mplsTunnelTable.

- The mplsXCTable is sparsely extended to support MPLS-TP XC(Cross Connect) specific objects.

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

6. Outline

Traffic Engineering support for the MPLS-TP tunnels requires the set up of the co-routed or associated bidirectional tunnel. The tables and MIB modules that are mentioned in the below subsections support the functionality described in documents [RFC5654] and [RFC6370]. These tables support both IP compatible and ITU Carrier Code (ICC) based tunnel configurations.

The below Figure 1 depicts how the table references are followed in this MIB.

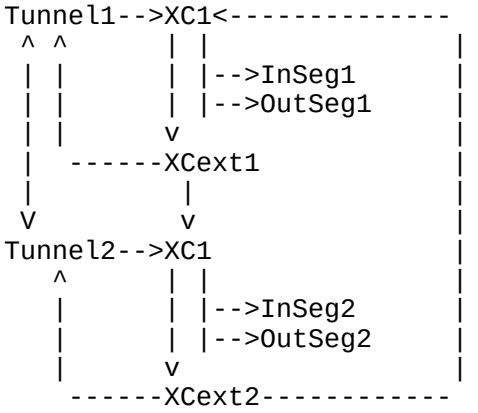

Figure 1: Table references of MIB modules

6.1 MIB Module Extensions

Four MIB modules are extended to support MPLS-TP tunnels, namely, MPLS-TE-EXT-STD-MIB, MPLS-TC-EXT-STD-MIB, MPLS-ID-STD-MIB and MPLS-LSR-EXT-STD-MIB. Following section provides the summary of changes.

6.1.1 Summary of MIB Module changes

- Node configuration table (mplsTunnelExtNodeConfigTable) for setting the local identifier for Tunnel Ingress and Egress identifiers.

- Node IP map table (mplsTunnelExtNodeIpMapTable) for querying the local identifier for a given Global_ID and Node_ID.

- Node ICC map table (mplsTunnelExtNodeIccMapTable) for querying the local identifier for a given ICC_Operator_ID and Node_ID.

- Tunnel extension table (mplsTunnelExtTable) for setting up MPLS-TP tunnels with sparse extension of mplsTunnelTable.

- Textual conventions and object definitions for MPLS-TP Tunnels

- Cross connect extension table (mplsXCExtTable) for setting up the MPLS-TP LSPs.

These tables are described in the subsequent sections.

6.2 MPLS-TE-EXT-STD-MIB

The TE MIB module extensions and details of the tables are described in the following sections.

6.2.1 mplsTunnelExtNodeConfigTable

The mplsTunnelExtNodeConfigTable is used to assign a local identifier for a given ICC_Operator_ID::Node_ID or Global_ID::Node_ID combination as defined in [RFC6923] and [RFC6370] respectively. The CC is a string of two characters, each being an uppercase Basic Latin alphabetic (i.e., A-Z). The ICC is a string of one to six characters, each an upper case Basic Latin alphabetic (i.e., A-Z) or numeric (i.e., 0-9). All of the characters are encoded using [T.50] as described in [RFC6370].

In the IP compatible mode, Global_ID::Node_ID, is used to uniquely identify a node. For each ICC_Operator_ID::Node_ID or Global_ID::Node_ID, there is a unique entry in the table representing a node. As the regular TE tunnels use IP address as LSR ID, the local identifier should be below the first valid IP address, which is 16777216[1.0.0.0]. Every node is assigned a local identifier within a range of 0 to 16777215. This local identifier is used for indexing into mplsTunnelTable as mplsTunnelIngressLSRId and mplsTunnelEgressLSRId.

For IP compatible environment, MPLS-TP tunnel is indexed by Tunnel Index, Tunnel Instance, Source Global_ID, Source Node_ID, Destination Global_ID and Destination Node_ID.

For ICC based environment, MPLS-TP tunnel is indexed by Tunnel Index, Tunnel Instance, Source CC, Source ICC, Source Node_ID, Destination CC, Destination ICC and Destination Node_ID.

As mplsTunnelTable is indexed by mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId, and mplsTunnelEgressLSRId, the MPLS-TP tunnel identifiers cannot be used directly.

The mplsTunnelExtNodeConfigTable will be used to store an entry for ICC_Operator_ID::Node_ID or Global_ID::Node_ID with a local identifier to be used as LSR ID in mplsTunnelTable.

6.2.2 mplsTunnelExtNodeIpMapTable

The read-only mplsTunnelExtNodeIpMapTable is used to query the local identifier assigned and stored in mplsTunnelExtNodeConfigTable for a given Global_ID::Node_ID. In order to query the local identifier, in the IP compatible mode, this table is indexed with Global_ID::Node_ID. In the IP compatible mode for a TP tunnel, Global_ID::Node_ID is used.

A separate query is made to get the local identifier of both Ingress and Egress Global_ID::Node_ID identifiers. These local identifiers are used as mplsTunnelIngressLSRId and mplsTunnelEgressLSRId, while indexing mplsTunnelTable.

6.2.3 mplsTunnelExtNodeIccMapTable

The read-only mplsTunnelExtNodeIccMapTable is used to query the local identifier assigned and stored in the mplsTunnelExtNodeConfigTable for a given ICC_Operator_ID::Node_ID.

A separate query is made to get the local identifier of both Ingress and Egress ICC_Operator_ID::Node_ID. These local identifiers are used as mplsTunnelIngressLSRId and mplsTunnelEgressLSRId, while indexing mplsTunnelTable.

6.2.4 mplsTunnelExtTable

This table sparsely extends the mplsTunnelTable in order to support MPLS-TP tunnels with additional objects. All the additional attributes specific to supporting TP tunnel are contained in this extended table and could be accessed with the mplsTunnelTable indices.

The gmplsTunnelReversePerfTable [RFC4802] should be used to provide per-tunnel packet performance information for the reverse direction of a bidirectional tunnel. It can be seen as supplementing the mplsTunnelPerfTable, which augments the mplsTunnelTable.

6.3 MPLS-TC-EXT-STD-MIB

This MIB module contains textual Conventions for LSPs of MPLS based transport networks.

6.4 MPLS-ID-STD-MIB

This MIB module contains generic object definitions for MPLS Traffic Engineering in transport networks.

6.5 MPLS-LSR-EXT-STD-MIB

This MIB module contains generic object definitions (Cross connect extension table - mplsXCExtTable, for setting up the MPLS-TP LSPs with sparse extension of mplsXCTable) for MPLS LSRs in transport networks.

6.6 The Use of RowPointer

This document follows the RowPointer usage as described in the section 10 of [RFC3812].

A new RowPointer object, mplsTunnelExtOppositeDirPtr, is added to mplsTunnelExtTable of MPLS-TE-EXT-STD-MIB MIB module. This RowPointer object points to the opposite direction tunnel entry.

Two additional RowPointers objects, mplsXCExtTunnelPointer and mplsXCExtOppositeDirXCPtr are added to mplsXCExtTable of MPLS-LSR-EXT-STD-MIB. The RowPointer mplsXCExtTunnelPointer is read-only object used to indicate the back pointer to the tunnel entry. The RowPointer mplsXCExtOppositeDirXCPtr object points to the opposite direction XC entry.

If these RowPointer returns zeroDotZero, it implies that there is no entry associated with the RowPointer object.

7. MIB Modules Interdependencies

This section provides an overview of the relationship between the MPLS-TP TE MIB module and other MPLS MIB modules.

The arrows in the following diagram show a 'depends on' relationship. A relationship "MIB module A depends on MIB module B" means that MIB module A uses an object, object identifier, or textual convention defined in MIB module B, or that MIB module A contains a pointer (index or RowPointer) to an object in MIB module B.

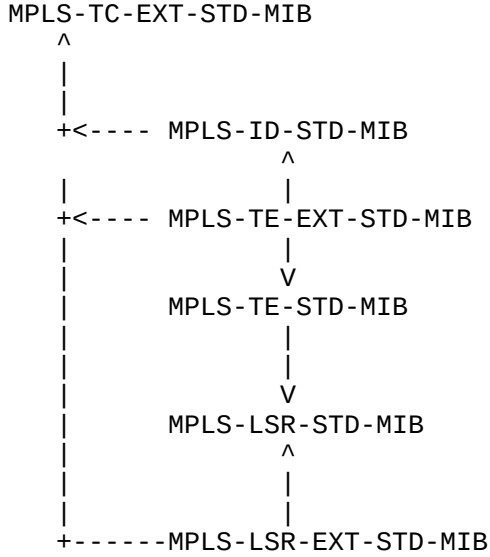

Figure 2: MIB modules interdependencies

Thus:

- All the new MPLS extension MIB modules depend on MPLS-TC-EXT-STD-MIB.
- MPLS-ID-STD-MIB contains references to objects in MPLS-TE-STD-MIB [RFC3812].
- MPLS-TE-EXT-STD-MIB contains references to objects in MPLS-TE-STD-MIB [RFC3812].
- MPLS-LSR-EXT-STD-MIB contains references to objects in MPLS-LSR-STD-MIB [RFC3813].

The mplsTunnelExtTable sparsely extends the mplsTunnelTable of MPLS-TE-STD-MIB [RFC3812]. This helps in associating the reverse direction tunnel information.

The mplsXCExtTable sparsely extends the mplsXCTable of MPLS-LSR-STD-MIB [RFC3813]. This helps in pointing back to the tunnel entry for easy tunnel access from XC entry.

Note that all of the MIB modules shown above in the figure also have a dependency on MPLS-TC-STD-MIB.

8. Dependencies between MIB Module Tables

The tables in MPLS-TE-EXT-STD-MIB are related as shown on the diagram below. The arrows indicate a reference from one table to another.

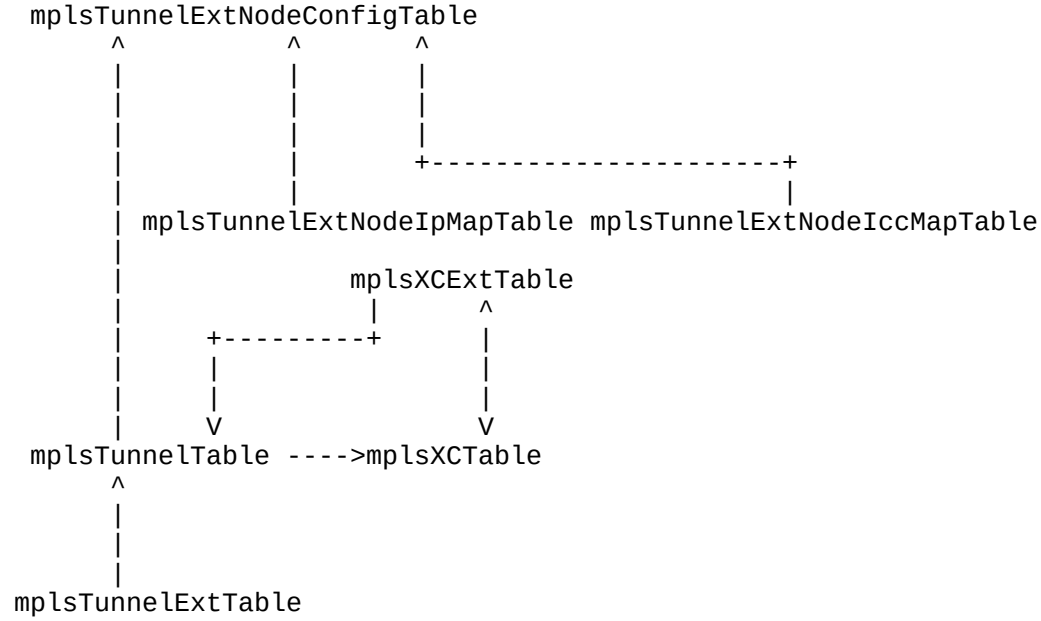

Figure 3: Dependencies between MIB module tables

An existing mplsTunnelTable uses the mplsTunnelExtNodeConfigTable table to map the Global_ID::Node_ID and/or ICC_Operator_ID::Node_ID with the local number in order to accommodate in the existing tunnel table's ingress/egress LSR-id.

New mplsTunnelExtTable table provides the reverse direction LSP information for the existing tunnel table in order to achieve bidirectional LSPs.

mplsXCExtTable sparsely extends the mplsLsrXCTable to provide backward reference to tunnel entry.

9. Example of MPLS-TP Tunnel Setup

In this section, we provide an example to configure MPLS-TP bidirectional tunnels with IP tunnel identifiers. This example provides the usage of MPLS-TP Tunnel MIB along with the extended MIB modules introduced in this document.

Do note that a MPLS-TP tunnel could be setup statically as well as signaled via control plane. This example considers accessing MIB objects on a head-end for a static and signaled MPLS-TP tunnels. This section shows the configuration of the forward and reverse direction MPLS-TP LSPs that run between East and West and vice-versa. Only objects relevant to MPLS-TP tunnels are illustrated here.

In mplsTunnelExtNodeConfigTable:

```
{
-- Non-IP Ingress LSR-Id (Index to the table)

  mplsTunnelExtNodeConfigLocalId              = 1,

  mplsTunnelExtNodeConfigGlobalId             = 1234,
  mplsTunnelExtNodeConfigNodeId               = 10,
-- Mandatory parameters needed to activate the row go here
  mplsTunnelExtNodeConfigRowStatus         = createAndGo (4)

-- Non-IP Egress LSR-Id (Index to the table)
  mplsTunnelExtNodeConfigLocalId              = 2,
  mplsTunnelExtNodeConfigGlobalId             = 1234,
  mplsTunnelExtNodeConfigNodeId               = 20,
-- Mandatory parameters needed to activate the row go here
  mplsTunnelExtNodeConfigRowStatus         = createAndGo (4)
}
```

This will create an entry in the mplsTunnelExtNodeConfigTable for a Global_ID::Node_ID. A separate entry is made for both Ingress LSR and Egress LSR.

The following read-only mplsTunnelExtNodeIpMapTable table is populated automatically upon creating an entry in mplsTunnelExtNodeConfigTable and this table is used to retrieve the local identifier for the given Global_ID::Node_ID.

In mplsTunnelExtNodeIpMapTable:

```
   {
   -- Global_ID (Index to the table)
     mplsTunnelExtNodeIpMapGlobalId             = 1234,
   -- Node Identifier (Index to the table)
     mplsTunnelExtNodeIpMapNodeId               = 10,
     mplsTunnelExtNodeIpMapLocalId              = 1

   -- Global_ID (Index to the table)
     mplsTunnelExtNodeIpMapGlobalId             = 1234,
   -- Node Identifier (Index to the table)
     mplsTunnelExtNodeIpMapNodeId               = 20,
     mplsTunnelExtNodeIpMapLocalId              = 2
   }
```

9.1. Example of MPLS-TP static co-routed bidirectional tunnel setup

The following denotes the co-routed bidirectional tunnel "head" entry.

9.1.1. mplsTunnelEntry

In mplsTunnelTable:

```
   {
     mplsTunnelIndex              = 1,
     mplsTunnelInstance           = 1,
   -- Local map number created in mplsTunnelExtNodeConfigTable for
   -- Ingress LSR-Id
     mplsTunnelIngressLSRId       = 1,

   -- Local map number created in mplsTunnelExtNodeConfigTable for
   -- Egress LSR-Id
     mplsTunnelEgressLSRId        = 2,
     mplsTunnelName               = "TP co-routed bidirectional LSP",
     mplsTunnelDescr              = "East to West",
     mplsTunnelIsIf               = true (1),
   --  RowPointer MUST point to the first accessible column
     mplsTunnelXCPointer          =
                                mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
     mplsTunnelSignallingProto    = none (1),
     mplsTunnelSetupPrio          = 0,
     mplsTunnelHoldingPrio        = 0,
     mplsTunnelSessionAttributes  = 0,
     mplsTunnelLocalProtectInUse  = false (0),
   --  RowPointer MUST point to the first accessible column
     mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.5,
     mplsTunnelInstancePriority   = 1,
     mplsTunnelHopTableIndex      = 1,
```

```
     mplsTunnelIncludeAnyAffinity = 0,
     mplsTunnelIncludeAllAffinity = 0,
     mplsTunnelExcludeAnyAffinity = 0,
     mplsTunnelRole               = head (1),
   -- Mandatory parameters needed to activate the row go here
     mplsTunnelRowStatus          = createAndGo (4)
   }
```

9.1.2. mplsTunnelExtEntry

```
   -- An MPLS extension table
   In mplsTunnelExtTable:
   {
     -- This opposite direction tunnel pointer may point to 0.0
     -- if co-routed bidirectional tunnel is managed by single tunnel
     -- entry
     mplsTunnelExtOppositeDirTnlPtr       = 0.0
     -- Set both the Ingress and Egress LocalId objects to TRUE as
     -- this tunnel entry uses the local identifiers.
     mplsTunnelExtIngressLSRLocalIdValid  = true,
     mplsTunnelExtEgressLSRLocalIdValid = true

   }
```

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the mplsInSegmentTable and mplsOutSegmentTable.

9.1.3. Forward direction mplsOutSegmentEntry

For the forward direction,

```
   In mplsOutSegmentTable:
   {
      mplsOutSegmentIndex          = 0x0000001,
      mplsOutSegmentInterface      = 13, -- outgoing interface
      mplsOutSegmentPushTopLabel   = true(1),
      mplsOutSegmentTopLabel       = 22, -- outgoing label

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

9.1.4. Reverse direction mplsInSegmentEntry

For the reverse direction,

```
In mplsInSegmentTable:
{
   mplsInSegmentIndex           = 0x0000001
   mplsInSegmentLabel           = 21, -- incoming label
   mplsInSegmentNPop            = 1,
   mplsInSegmentInterface       = 13, -- incoming interface

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

Next, two cross-connect entries are created in the mplsXCTable of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.1.5. Forward direction mplsXCEntry

```
In mplsXCTable:
{
   mplsXCIndex                = 0x01,
   mplsXCInSegmentIndex       = 0x00000000,
   mplsXCOutSegmentIndex      = 0x00000001,
   mplsXCLspId                = 0x0102 -- unique ID

   -- only a single outgoing label
   mplsXCLabelStackIndex      = 0x00,
   mplsXCRowStatus            = createAndGo(4)

}
```

9.1.6. Reverse direction mplsXCEntry

```
In mplsXCTable:
{
   mplsXCIndex                = 0x01,
   mplsXCInSegmentIndex       = 0x00000001,
   mplsXCOutSegmentIndex      = 0x00000000,
   mplsXCLspId                = 0x0102 -- unique ID
   -- only a single outgoing label
   mplsXCLabelStackIndex      = 0x00,
   mplsXCRowStatus            = createAndGo(4)
}
```

This table entry is extended by entry in the mplsXCExtTable. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsXCExtTable has the same index values as the entry in

the mplsXCTable.

9.1.7. Forward direction mplsXCExtEntry

```
In mplsXCExtTable (0x01, 0x00000000, 0x00000001)
{
  -- Back pointer from XC table to Tunnel table
  mplsXCExtTunnelPointer         = mplsTunnelName.1.1.1.2
  mplsXCExtOppositeDirXCPtr         =
                              mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0
}
```

9.1.8. Reverse direction mplsXCExtEntry

Next for the reverse direction:

```
In mplsXCExtTable (0x01, 0x00000001, 0x00000000)
{
  -- Back pointer from XC table to Tunnel table
  mplsXCExtTunnelPointer         = mplsTunnelName.1.1.1.2
  mplsXCExtOppositeDirXCPtr         =
                              mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1
}
```

9.2. Example of MPLS-TP static associated bidirectional tunnel setup

The MPLS-TP associated bidirectional tunnel is implemented by two different unidirectional tunnels [Forward and Reverse LSPs] and these are associated together using mplsTunnelExtTable. Two different tunnel entries to provide the forward and reverse directions MAY be used for co-routed bidirectional tunnels as well.

The following denotes the associated bidirectional forward tunnel "head" entry:

9.2.1. Forward direction mplsTunnelEntry

In mplsTunnelTable:

```
{
  mplsTunnelIndex             = 1,
  mplsTunnelInstance          = 1,
-- Local map number created in mplsTunnelExtNodeConfigTable for
-- Ingress LSR-Id
  mplsTunnelIngressLSRId      = 1,

-- Local map number created in mplsTunnelExtNodeConfigTable for
-- Egress LSR-Id
```

```
     mplsTunnelEgressLSRId        = 2,
     mplsTunnelName               = "TP associated bidirectional
                                    forward LSP",
     mplsTunnelDescr              = "East to West",
     mplsTunnelIsIf               = true (1),
   --  RowPointer MUST point to the first accessible column
     mplsTunnelXCPointer          =
                                 mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
     mplsTunnelSignallingProto    = none (1),
     mplsTunnelSetupPrio          = 0,
     mplsTunnelHoldingPrio        = 0,
     mplsTunnelSessionAttributes  = 0,
     mplsTunnelLocalProtectInUse  = false (0),
   --  RowPointer MUST point to the first accessible column
     mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.5,
     mplsTunnelInstancePriority   = 1,
     mplsTunnelHopTableIndex      = 1,
     mplsTunnelIncludeAnyAffinity = 0,

     mplsTunnelIncludeAllAffinity = 0,
     mplsTunnelExcludeAnyAffinity = 0,
     mplsTunnelRole               = head (1),
   -- Mandatory parameters needed to activate the row go here
     mplsTunnelRowStatus          = createAndGo (4)
   }
```

9.2.2. Forward direction mplsTunnelExtEntry

```
   For Associated bidirectional forward LSP,
   In mplsTunnelExtTable:
   {
     mplsTunnelExtOppositeDirPtr       = mplsTunnelName.2.1.2.1
    -- Set both the Ingress and Egress LocalId objects to TRUE as
     -- this tunnel entry uses the local identifiers.
     mplsTunnelExtIngressLSRLocalIdValid  = true,
     mplsTunnelExtEgressLSRLocalIdValid = true
   }
```

9.2.3. Forward direction mplsOutSegmentTable

```
   For the forward direction.

   In mplsOutSegmentTable:
   {
      mplsOutSegmentIndex          = 0x0000001,
      mplsOutSegmentInterface      = 13, -- outgoing interface
      mplsOutSegmentPushTopLabel   = true(1),
      mplsOutSegmentTopLabel       = 22, -- outgoing label
```

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

9.2.4. Forward direction mplsXCEntry

```
   In mplsXCTable:
   {
      mplsXCIndex                = 0x01,
      mplsXCInSegmentIndex       = 0x00000000,
      mplsXCOutSegmentIndex      = 0x00000001,
      mplsXCLspId                = 0x0102 -- unique ID
      -- only a single outgoing label
      mplsXCLabelStackIndex      = 0x00,
      mplsXCRowStatus            = createAndGo(4)

   }
```

9.2.5. Forward direction mplsXCExtEntry

```
   In mplsXCExtTable (0x01, 0x00000000, 0x00000001)
   {
     -- Back pointer from XC table to Tunnel table
     mplsXCExtTunnelPointer         = mplsTunnelName.1.1.1.2
     mplsXCExtOppositeDirXCPtr         =
                               mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0
   }
```

9.2.6. Reverse direction mplsTunnelEntry

The following denotes the configured associated bidirectional reverse tunnel "tail" entry:

```
       In mplsTunnelTable:

   {
     mplsTunnelIndex              = 2,
     mplsTunnelInstance           = 1,
   -- Local map number created in mplsTunnelExtNodeConfigTable for
   -- Ingress LSR-Id
     mplsTunnelIngressLSRId       = 2,
   -- Local map number created in mplsTunnelExtNodeConfigTable for
   -- Egress LSR-Id
     mplsTunnelEgressLSRId        = 1,
     mplsTunnelName               = "TP associated bidirectional
                                     reverse LSP",
     mplsTunnelDescr              = "West to East",
```

```
    mplsTunnelIsIf               = true (1),
  --  RowPointer MUST point to the first accessible column
    mplsTunnelXCPointer          =
                              mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0,
    mplsTunnelSignallingProto    = none (1),
    mplsTunnelSetupPrio          = 0,
    mplsTunnelHoldingPrio        = 0,
    mplsTunnelSessionAttributes  = 0,
    mplsTunnelLocalProtectInUse  = false (0),

  --  RowPointer MUST point to the first accessible column
    mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.5,
    mplsTunnelInstancePriority   = 1,
    mplsTunnelHopTableIndex      = 1,
    mplsTunnelIncludeAnyAffinity = 0,
    mplsTunnelIncludeAllAffinity = 0,
    mplsTunnelExcludeAnyAffinity = 0,
    mplsTunnelRole               = head (1),
  -- Mandatory parameters needed to activate the row go here

    mplsTunnelRowStatus          = createAndGo (4)
  }
```

9.2.7. Reverse direction mplsTunnelExtEntry

```
  For Associated bidirectional reverse LSP,
  In mplsTunnelExtTable:
  {
    mplsTunnelExtOppositeDirPtr       = mplsTunnelName.1.1.1.2
    -- Set both the Ingress and Egress LocalId objects to TRUE as
    -- this tunnel entry uses the local identifiers.
    mplsTunnelExtIngressLSRLocalIdValid  = true,
    mplsTunnelExtEgressLSRLocalIdValid = true
  }
```

9.2.8. Reverse direction mplsInSegmentEntry

```
  We must next create the appropriate in-segment and out-segment
  entries. These are done in [RFC3813] using the mplsInSegmentTable and
  mplsOutSegmentTable.

  In mplsInSegmentTable:
  {
     mplsInSegmentIndex           = 0x0000001
     mplsInSegmentLabel           = 21, -- incoming label
     mplsInSegmentNPop            = 1,
     mplsInSegmentInterface       = 13, -- incoming interface
```

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

Next, two cross-connect entries are created in the mplsXCTable of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.2.9. Reverse direction mplsXCEntry

```
   In mplsXCTable:
   {
      mplsXCIndex                = 0x01,
      mplsXCInSegmentIndex       = 0x00000001,
      mplsXCOutSegmentIndex      = 0x00000000,
      mplsXCLspId                = 0x0102 -- unique ID
      -- only a single outgoing label
      mplsXCLabelStackIndex      = 0x00,

      mplsXCRowStatus            = createAndGo(4)
   }
```

This table entry is extended by entry in the mplsXCExtTable. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsXCExtTable has the same index values as the entry in the mplsXCTable.

9.2.10. Reverse direction mplsXCExtEntry

Next for the reverse direction:

```
   In mplsXCExtTable (0x01, 0x00000001, 0x00000000)
   {
     -- Back pointer from XC table to Tunnel table
     mplsXCExtTunnelPointer         = mplsTunnelName.2.1.2.1
     mplsXCExtOppositeDirXCPtr        =
                              mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1
   }
```

9.3. Example of MPLS-TP signaled co-routed bidirectional tunnel setup

The following denotes the co-routed bidirectional tunnel "head" entry and in intermediate and tail-end nodes, the tunnel table and its associated tables are created by the local management subsystem (e.g. agent) when the MPLS TP tunnel is signaled successfully.

Refer [RFC3812] and [RFC4802] for signaled tunnel table configuration examples.

9.3.1. mplsTunnelEntry

In mplsTunnelTable:

```
{
  mplsTunnelIndex              = 1,
  mplsTunnelInstance           = 0,
-- Local map number created in mplsTunnelExtNodeConfigTable for
-- Ingress LSR-Id, for the intermediate and tail-end nodes,
-- the local management entity is expected to pick a first available
-- local identifier which is not used in mplsTunnelTable.
  mplsTunnelIngressLSRId       = 1,

-- Local map number created in mplsTunnelExtNodeConfigTable for
-- Egress LSR-Id
  mplsTunnelEgressLSRId        = 2,
  mplsTunnelName               = "TP co-routed bidirectional LSP",
  mplsTunnelDescr              = "East to West",
  mplsTunnelIsIf               = true (1),

--  RowPointer MUST point to the first accessible column
  mplsTunnelXCPointer          =
                              mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
  mplsTunnelSignallingProto    = none (1),
  mplsTunnelSetupPrio          = 0,
  mplsTunnelHoldingPrio        = 0,
  mplsTunnelSessionAttributes  = 0,
  mplsTunnelLocalProtectInUse  = false (0),
--  RowPointer MUST point to the first accessible column
  mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.5,
  mplsTunnelInstancePriority   = 1,
  mplsTunnelHopTableIndex      = 1,
  mplsTunnelIncludeAnyAffinity = 0,
  mplsTunnelIncludeAllAffinity = 0,
  mplsTunnelExcludeAnyAffinity = 0,
  mplsTunnelRole               = head (1),
-- Mandatory parameters needed to activate the row go here
  mplsTunnelRowStatus          = createAndGo (4)
}
```

9.3.2. mplsTunnelExtEntry

```
-- An MPLS extension table
In mplsTunnelExtTable:
{
```

```
    -- This opposite direction tunnel pointer may point to 0.0
    -- if co-routed bidirectional tunnel is managed by single tunnel
    -- entry
    mplsTunnelExtOppositeDirTnlPtr       = 0.0
    -- Set both the Ingress and Egress LocalId objects to TRUE as
    -- this tunnel entry uses the local identifiers.
    mplsTunnelExtIngressLSRLocalIdValid  = true,
    mplsTunnelExtEgressLSRLocalIdValid = true
  }
```

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the mplsInSegmentTable and mplsOutSegmentTable.

9.3.3. Forward direction mplsOutSegmentEntry

The forward direction mplsOutSegmentTable will be populated automatically based on the information received from the signaling protocol.

9.3.4. Reverse direction mplsInSegmentEntry

The reverse direction mplsOutSegmentTable will be populated automatically based on the information received from the signaling protocol.

Next, two cross-connect entries are created in the mplsXCTable of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.3.5. Forward direction mplsXCEntry

The forward direction mplsXCEntry will be populated as soon as the forward path label information is available.

9.3.6. Reverse direction mplsXCEntry

The reverse direction mplsXCEntry will be populated as soon as the reverse path label information is available.

This table entry is extended by entry in the mplsXCExtTable. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsXCExtTable has the same index values as the entry in the mplsXCTable.

9.3.7. Forward direction mplsXCExtEntry

Once the forward path information is negotiated using signaling

protocol, the forward direction mplsXCExtEntry will be created for associating the opposite direction XC entry and tunnel table entry.

9.3.8. Reverse direction mplsXCExtEntry

Once the reverse path information is negotiated using signaling protocol, the reverse direction mplsXCExtEntry will be created for associating the opposite direction XC entry and tunnel table entry.

10. MPLS Textual Convention Extension MIB definitions

```
  MPLS-TC-EXT-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, Unsigned32
      FROM SNMPv2-SMI               -- [RFC2578]

   TEXTUAL-CONVENTION
      FROM SNMPv2-TC                -- [RFC2579]

   mplsStdMIB
      FROM MPLS-TC-STD-MIB          -- [RFC3811]

   ;

mplsTcExtStdMIB MODULE-IDENTITY

   LAST-UPDATED
      "201412180000Z" -- December 18, 2014
   ORGANIZATION
      "Multiprotocol Label Switching (MPLS) Working Group"
   CONTACT-INFO
      "
             Venkatesan Mahalingam
             Dell Inc,
             5450 Great America Parkway,
             Santa Clara, CA 95054, USA
         Email: venkat.mahalingams@gmail.com

             Kannan KV Sampath
             Redeem,
             India
         Email: kannankvs@gmail.com

             Sam Aldrin
             Huawei Technologies
             2330 Central Express Way,
             Santa Clara, CA 95051, USA
```

```
          Email:  aldrin.ietf@gmail.com

                Thomas D. Nadeau
          Email: tnadeau@lucidvision.com
         "
      DESCRIPTION
          "Copyright (c) 2014 IETF Trust and the persons identified
           as the document authors.  All rights reserved.

           This MIB module contains Textual Conventions for LSPs of MPLS
           based transport networks."

      -- Revision history.

      REVISION

         "201412180000Z" -- December 18, 2014
      DESCRIPTION
           "MPLS Textual Convention Extensions"

      ::= { mplsStdMIB www } -- www to be replaced with correct value

   MplsGlobalId ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
           "This object contains the Textual Convention for IP based
            operator unique identifier (Global_ID), the Global_ID can
            contain the 2-octet or 4-octet value of the operator's
            Autonomous System Number (ASN).

            When the Global_ID is derived from a 2-octet AS number,
            the two high-order octets of this 4-octet identifier
            MUST be set to zero(0x00). Further ASN 0 is reserved.
            The size of the Global_ID string MUST be zero if
            the Global_ID is invalid.

            Note that a Global_ID of zero is limited to entities
            contained within a single operator and MUST NOT be used
            across an Network-to-Network Interface (NNI). A non-zero
            Global_ID MUST be derived from an ASN owned by
            the operator."
      REFERENCE
           "MPLS Transport Profile (MPLS-TP) Identifiers, [RFC6370]
            Section 3"
      SYNTAX  OCTET STRING (SIZE (4))

   MplsCcId ::= TEXTUAL-CONVENTION
   STATUS      current
```

```
   DESCRIPTION
        "The CC (Country Code) is a string of two characters, each
         being an uppercase Basic Latin alphabetic (i.e., A-Z).
         The characters are encoded using ITU-T Recommendation T.50.
         The size of the CC string MUST be zero if the CC identifier
         is invalid."
   REFERENCE
        "MPLS-TP Identifiers Following ITU-T Conventions,
         RFC 6923, Section 3.
         International Reference Alphabet (IRA) (Formerly
         International Alphabet No. 5 or IA5) - Information
         technology - 7-bit coded character set for information
         exchange, ITU-T Recommendation T.50, September 1992. "
   SYNTAX  OCTET STRING (SIZE (0|2))

   MplsIccId ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION
        "The ICC is a string of one to six characters, each
         an upper case Basic Latin alphabetic (i.e., A-Z) or
         numeric (i.e., 0-9).  The characters are encoded
         using ITU-T Recommendation T.50. The size of
         the ICC string MUST be zero if the ICC identifier
         is invalid."
   REFERENCE
        "MPLS-TP Identifiers Following ITU-T Conventions,
         RFC6923, Section 3.
         International Reference Alphabet (IRA) (Formerly
         International Alphabet No. 5 or IA5) - Information
         technology - 7-bit coded character set for information
         exchange, ITU-T Recommendation T.50, September 1992. "
   SYNTAX  OCTET STRING (SIZE (0|1..6))

   MplsNodeId ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "d"
      STATUS      current
      DESCRIPTION
          "The Node_ID is assigned within the scope of
           the Global_ID/ICC_Operator_ID.

           When IPv4 addresses are in use, the value of this object
           can be derived from the LSR's IPv4 loop back address.
           When IPv6 addresses are in use, the value of this object
           can be a 32-bit value unique within the scope of
           a Global_ID.

           Note that, when IP reachability is not needed, the 32-bit
           Node_ID is not required to have any association
```

```
           with the IPv4 address space. The value of 0 indicates
           the invalid Node identifier."
       REFERENCE
           "MPLS Transport Profile (MPLS-TP) Identifiers, [RFC6370]
            Section 4"
       SYNTAX  Unsigned32 (0|1..4294967295)

      -- MPLS-TC-EXT-STD-MIB module ends
      END
```

11. MPLS Identifier MIB definitions

```
    MPLS-ID-STD-MIB DEFINITIONS ::= BEGIN

    IMPORTS
     MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI                                 -- [RFC2578]
     MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF                                -- [RFC2580]
     mplsStdMIB
        FROM MPLS-TC-STD-MIB                            -- [RFC3811]
     MplsGlobalId, MplsCcId, MplsIccId, MplsNodeId
        FROM MPLS-TC-EXT-STD-MIB
     ;

   mplsIdStdMIB MODULE-IDENTITY
     LAST-UPDATED
        "201412120000Z" -- December 12, 2014
     ORGANIZATION
        "Multiprotocol Label Switching (MPLS) Working Group"

     CONTACT-INFO
        "
                Venkatesan Mahalingam
                Dell Inc,
                5450 Great America Parkway,
                Santa Clara, CA 95054, USA
     Email: venkat.mahalingams@gmail.com

                Kannan KV Sampath

                Redeem,
                India
          Email: kannankvs@gmail.com


                Sam Aldrin
                Huawei Technologies
```

```
               2330 Central Express Way,
               Santa Clara, CA 95051, USA
            Email:  aldrin.ietf@gmail.com

               Thomas D. Nadeau
            Email: tnadeau@lucidvision.com
          "
        DESCRIPTION
            "Copyright (c) 2014 IETF Trust and the persons identified
             as the document authors.  All rights reserved.

             This MIB module contains generic object definitions for
             MPLS Traffic Engineering in transport networks."

        -- Revision history.

        REVISION
            "201412120000Z" -- December 12, 2014
        DESCRIPTION
             "This MIB modules defines the MIB objects for MPLS-TP
              identifiers"

        ::= { mplsStdMIB xxx } -- xxx to be replaced with correct value

        -- notifications
        mplsIdNotifications OBJECT IDENTIFIER ::= { mplsIdStdMIB 0 }
        -- tables, scalars
        mplsIdObjects       OBJECT IDENTIFIER ::= { mplsIdStdMIB 1 }
        -- conformance
        mplsIdConformance   OBJECT IDENTIFIER ::= { mplsIdStdMIB 2 }


        -- MPLS common objects

      mplsIdGlobalId OBJECT-TYPE
           SYNTAX      MplsGlobalId
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "This object allows the operator or service provider to
                assign a unique operator identifier also called MPLS-TP
                Global_ID.
                If this value is used in mplsTunnelExtNodeConfigGlobalId
                for mapping Global_ID::Node_ID with the local identifier
                then this object value MUST NOT be changed."
          ::= { mplsIdObjects 1 }

      mplsIdNodeId OBJECT-TYPE
```

```
        SYNTAX      MplsNodeId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
           "This object allows the operator or service provider to
            assign a unique MPLS-TP Node_ID. The Node_ID is assigned
            within the scope of the Global_ID/ICC_Operator_ID.
            If this value is used in mplsTunnelExtNodeConfigNodeId
            for mapping Global_ID::Node_ID with the local identifier
            then this object value SHOULD NOT be changed.
            If this value is used in mplsTunnelExtNodeConfigNodeId
            for mapping ICC_Operator_ID::Node_ID with the local
            identifier then this object value MUST NOT be changed."
       ::= { mplsIdObjects 2 }

   mplsIdCc OBJECT-TYPE
        SYNTAX      MplsCcId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
           "This object allows the operator or service provider to
            assign a Country Code (CC) to the node. Global
            uniqueness of ICC is assured by concatenating the ICC
            with a Country Code (CC).
            If this value is used in mplsTunnelExtNodeConfigCcId
            for mapping ICC_Operator_ID::Node_ID with the local
            identifier then this object value MUST NOT be changed."
       REFERENCE
            "MPLS-TP Identifiers Following ITU-T Conventions,
             [RFC6923] Section 3"
           ::= { mplsIdObjects 3 }


   mplsIdIcc OBJECT-TYPE
        SYNTAX      MplsIccId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
           "This object allows the operator or service provider to
            assign a unique MPLS-TP ITU-T Carrier Code (ICC) to
            the node. Together, the CC and the ICC form
            the ICC_Operator_ID as CC::ICC.
            If this value is used in mplsTunnelExtNodeConfigIccId
            for mapping ICC_Operator_ID::Node_ID with the local
            identifier then this object value MUST NOT be changed."
       REFERENCE
            "MPLS-TP Identifiers Following ITU-T Conventions,
             [RFC6923] Section 3"
```

```
          ::= { mplsIdObjects 4 }


    -- Module compliance.

   mplsIdCompliances
      OBJECT IDENTIFIER ::= { mplsIdConformance 1 }

   mplsIdGroups
      OBJECT IDENTIFIER ::= { mplsIdConformance 2 }

   -- Compliance requirement for fully compliant implementations.

   mplsIdModuleFullCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
           "Compliance statement for agents that provide full
             support of the MPLS-ID-STD-MIB module."

      MODULE -- this module


         -- The mandatory group has to be implemented by all LSRs that
         -- originate, terminate, or act as transit for MPLS-TP tunnels.

         GROUP mplsIdIpOperatorGroup
         DESCRIPTION
             "This group is mandatory for devices which support
              IP based identifier configuration."

         GROUP mplsIdIccOperatorGroup
         DESCRIPTION
             "This group is mandatory for devices which support
              ICC based identifier configuration."

          ::= { mplsIdCompliances 1 }

          -- Compliance requirement for read-only implementations.

         mplsIdModuleReadOnlyCompliance MODULE-COMPLIANCE
            STATUS current
            DESCRIPTION
                 "Compliance statement for agents that only provide
                  read-only support for the MPLS-ID-STD-MIB module."

         MODULE -- this module

         GROUP mplsIdIpOperatorGroup
```

```
        DESCRIPTION
            "This group is mandatory for devices which support
             IP based identifier configuration."

        GROUP mplsIdIccOperatorGroup
        DESCRIPTION
            "This group is mandatory for devices which support
             ICC based identifier configuration."

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

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

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

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

        ::= { mplsIdCompliances 2 }

    -- Units of conformance.

        mplsIdIpOperatorGroup OBJECT-GROUP
              OBJECTS { mplsIdGlobalId,
                        mplsIdNodeId
              }
              STATUS  current
              DESCRIPTION
                  "The objects in this group are optional for ICC based
                   node."
              ::= { mplsIdGroups 1 }

        mplsIdIccOperatorGroup OBJECT-GROUP
              OBJECTS { mplsIdNodeId,
                        mplsIdCc,
                        mplsIdIcc
```

```
            }
            STATUS  current
            DESCRIPTION
               "The objects in this group are optional for IP based
                node."
            ::= { mplsIdGroups 2 }

    -- MPLS-ID-STD-MIB module ends
    END
```

12. MPLS LSR Extension MIB definitions

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

   IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE
         FROM SNMPv2-SMI                                  -- [RFC2578]
      MODULE-COMPLIANCE, OBJECT-GROUP
         FROM SNMPv2-CONF                                 -- [RFC2580]
      mplsStdMIB
         FROM MPLS-TC-STD-MIB                             -- [RFC3811]
      RowPointer
         FROM SNMPv2-TC                                   -- [RFC2579]
      mplsXCIndex, mplsXCInSegmentIndex, mplsXCOutSegmentIndex,
      mplsInterfaceGroup, mplsInSegmentGroup, mplsOutSegmentGroup,
      mplsXCGroup, mplsLsrNotificationGroup
         FROM MPLS-LSR-STD-MIB;                           -- [RFC3813]

   mplsLsrExtStdMIB MODULE-IDENTITY
      LAST-UPDATED
         "201412120000Z" -- December 12, 2014
      ORGANIZATION
         "Multiprotocol Label Switching (MPLS) Working Group"
      CONTACT-INFO
         "
                Venkatesan Mahalingam
                Dell Inc,
                5450 Great America Parkway,
                Santa Clara, CA 95054, USA
          Email: venkat.mahalingams@gmail.com

                Kannan KV Sampath
                Redeem,
                India
          Email: kannankvs@gmail.com

                Sam Aldrin
                Huawei Technologies
```

```
                2330 Central Express Way,
                Santa Clara, CA 95051, USA

           Email:  aldrin.ietf@gmail.com

                Thomas D. Nadeau
          Email: tnadeau@lucidvision.com
         "
      DESCRIPTION
          "Copyright (c) 2014 IETF Trust and the persons identified
           as the document authors.  All rights reserved.

           This MIB module contains generic object definitions for
           MPLS LSR in transport networks."

      -- Revision history.

      REVISION
        "201412120000Z" -- December 12, 2014
      DESCRIPTION
           "MPLS LSR specific MIB objects extension"

      ::= { mplsStdMIB yyy } -- yyy to be replaced with correct value


   -- notifications
   mplsLsrExtNotifications OBJECT IDENTIFIER ::= { mplsLsrExtStdMIB 0 }

   -- tables, scalars
   mplsLsrExtObjects       OBJECT IDENTIFIER
                            ::= { mplsLsrExtStdMIB 1 }
   -- conformance
   mplsLsrExtConformance   OBJECT IDENTIFIER
                            ::= { mplsLsrExtStdMIB 2 }

   -- MPLS LSR common objects
   mplsXCExtTable  OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsXCExtEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
          "This table sparse augments the mplsXCTable of
           MPLS-LSR-STD-MIB [RFC3813] to provide MPLS-TP specific
           information about associated tunnel information"
       REFERENCE
          "1. Multiprotocol Label Switching (MPLS) Label Switching
           Router (LSR) Management Information Base (MIB), RFC 3813."
   ::= { mplsLsrExtObjects 1 }
```

```
   mplsXCExtEntry  OBJECT-TYPE
       SYNTAX        MplsXCExtEntry
       MAX-ACCESS    not-accessible

       STATUS        current
       DESCRIPTION

          "An entry in this table sparsely extends the cross connect
           information represented by an entry in
           the mplsXCTable in MPLS-LSR-STD-MIB [RFC3813] through
           a sparse augmentation.  An entry can be created by
           a network operator via SNMP SET commands, or in
           response to signaling protocol events."
       REFERENCE
          "1. Multiprotocol Label Switching (MPLS) Label Switching
           Router (LSR) Management Information Base (MIB), RFC 3813."


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

   MplsXCExtEntry ::= SEQUENCE {
      mplsXCExtTunnelPointer        RowPointer,
      mplsXCExtOppositeDirXCPtr     RowPointer
   }

   mplsXCExtTunnelPointer OBJECT-TYPE
       SYNTAX        RowPointer
       MAX-ACCESS    read-only
       STATUS        current
       DESCRIPTION
          "This read-only object indicates the back pointer to
           the tunnel entry segment.
           The only valid value for Tunnel Pointer is
           mplsTunnelTable entry."
       REFERENCE
          "1. Multiprotocol Label Switching (MPLS) Label Switching
           Router (LSR) Management Information Base (MIB), RFC 3813."
    ::= { mplsXCExtEntry 1 }

   mplsXCExtOppositeDirXCPtr OBJECT-TYPE
       SYNTAX        RowPointer
       MAX-ACCESS    read-create
       STATUS        current
       DESCRIPTION
          "This object indicates the pointer to the opposite
           direction XC entry.  This object cannot be modified if
```

```
            mplsXCRowStatus for the corresponding entry in the
            mplsXCTable is active(1). If this pointer is not set or
            removed, mplsXCOperStatus should be set to down(2)."
        REFERENCE
           "1. Multiprotocol Label Switching (MPLS) Label Switching
            Router (LSR) Management Information Base (MIB), RFC 3813."
    ::= { mplsXCExtEntry 2 }

    mplsLsrExtCompliances
       OBJECT IDENTIFIER ::= { mplsLsrExtConformance 1 }

    mplsLsrExtGroups
       OBJECT IDENTIFIER ::= { mplsLsrExtConformance 2 }

    -- Compliance requirement for fully compliant implementations.

    mplsLsrExtModuleFullCompliance MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
           "Compliance statement for agents that provide full support
            for MPLS-LSR-EXT-STD-MIB.
            The mandatory group has to be implemented by all LSRs
            that originate, terminate, or act as transit for
            TE-LSPs/tunnels.
            In addition, depending on the type of tunnels supported,
            other groups become mandatory as explained below."

     MODULE MPLS-LSR-STD-MIB -- The MPLS-LSR-STD-MIB, RFC3813

     MANDATORY-GROUPS {
       mplsInSegmentGroup,
       mplsOutSegmentGroup,
       mplsXCGroup,
       mplsLsrNotificationGroup
     }

     MODULE -- this module




     MANDATORY-GROUPS    {
       mplsXCExtGroup
     }

     ::= { mplsLsrExtCompliances 1 }

    -- Compliance requirement for implementations that provide
```

```
    -- read-only access.

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


     MODULE MPLS-LSR-STD-MIB

     MANDATORY-GROUPS {
         mplsInterfaceGroup,
         mplsInSegmentGroup,
         mplsOutSegmentGroup
     }

     MODULE -- this module

     GROUP mplsXCExtReadOnlyObjectsGroup
     DESCRIPTION
           "This group is mandatory for devices which support
            opposite direction XC configuration of tunnels."

     -- mplsXCExtTable
          OBJECT mplsXCExtOppositeDirXCPtr
          MIN-ACCESS   read-only
          DESCRIPTION
              "Write access is not required.
               This object indicates the pointer to the opposite
               direction XC entry. The only valid value for XC
               Pointer is mplsXCTable entry."
          ::= { mplsLsrExtCompliances 2 }

     -- Units of conformance.

     mplsXCExtGroup  OBJECT-GROUP
     OBJECTS {
         mplsXCExtTunnelPointer,
         mplsXCExtOppositeDirXCPtr
     }
     STATUS  current
     DESCRIPTION
         "This object should be supported in order to access
         the tunnel entry from XC entry."
     ::= { mplsLsrExtGroups 1 }
```

```
    mplsXCExtReadOnlyObjectsGroup OBJECT-GROUP
    OBJECTS {
        mplsXCExtTunnelPointer,
        mplsXCExtOppositeDirXCPtr
    }
    STATUS  current
    DESCRIPTION
        "This Object is needed to associate the opposite direction
        (forward/reverse) XC entry."
    ::= { mplsLsrExtGroups 2 }


   -- MPLS-LSR-EXT-STD-MIB module ends
   END
```

13. MPLS Tunnel Extension MIB definitions

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

    IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE
          FROM SNMPv2-SMI                                  -- [RFC2578]
       MODULE-COMPLIANCE, OBJECT-GROUP
          FROM SNMPv2-CONF                                 -- [RFC2580]
       TruthValue, RowStatus, RowPointer, StorageType
          FROM SNMPv2-TC                                   -- [RFC2579]
       IndexIntegerNextFree
          FROM DIFFSERV-MIB                                -- [RFC3289]
       MplsGlobalId, MplsNodeId, MplsCcId, MplsIccId
          FROM MPLS-TC-EXT-STD-MIB
       mplsStdMIB, MplsTunnelIndex, MplsTunnelInstanceIndex,
       MplsExtendedTunnelId
          FROM MPLS-TC-STD-MIB                             -- [RFC3811]
       mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
       mplsTunnelEgressLSRId
          FROM MPLS-TE-STD-MIB                             -- [RFC3812]
       ;

    mplsTeExtStdMIB MODULE-IDENTITY
       LAST-UPDATED
           "201412120000Z" -- December 12, 2014
       ORGANIZATION
          "Multiprotocol Label Switching (MPLS) Working Group"
       CONTACT-INFO
          "
                 Venkatesan Mahalingam
                 Dell Inc,
```

```
                5450 Great America Parkway,
                Santa Clara, CA 95054, USA
            Email: venkat.mahalingams@gmail.com

                Kannan KV Sampath
                Redeem,
                India
            Email: kannankvs@gmail.com

                Sam Aldrin
                Huawei Technologies
                2330 Central Express Way,
                Santa Clara, CA 95051, USA
            Email:  aldrin.ietf@gmail.com

                Thomas D. Nadeau
             Email: tnadeau@lucidvision.com
              "
            DESCRIPTION
                "Copyright (c) 2014 IETF Trust and the persons identified
                 as the document authors.  All rights reserved.

                 This MIB module contains generic object definitions for
                 MPLS Traffic Engineering in transport networks."


            -- Revision history.

            REVISION
               "201412120000Z" -- December 12, 2014

            DESCRIPTION
                 "MPLS TE MIB objects extension"

            ::= { mplsStdMIB zzz } -- zzz to be replaced
                                   -- with correct value

         -- Top level components of this MIB module.


         -- tables, scalars
         mplsTeExtObjects       OBJECT IDENTIFIER
                                        ::= { mplsTeExtStdMIB 0 }
         -- conformance
         mplsTeExtConformance   OBJECT IDENTIFIER
                                        ::= { mplsTeExtStdMIB 1 }

        -- Start of MPLS Transport Profile Node configuration table
```

```
    mplsTunnelExtNodeConfigLocalIdNext OBJECT-TYPE
     SYNTAX        IndexIntegerNextFree (0..16777215)
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
        "This object contains an unused value for
         mplsTunnelExtNodeConfigLocalId, or a zero to indicate
         that none exist. Negative values are not allowed,
         as they do not correspond to valid values of
         mplsTunnelExtNodeConfigLocalId."
      ::= { mplsTeExtObjects 1 }

      mplsTunnelExtNodeConfigTable OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsTunnelExtNodeConfigEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
         "This table allows the operator to map a node or
          LSR Identifier (IP compatible [Global_ID::Node_ID] or
          ICC based [ICC_Operator_ID::Node_ID]) with a local
          identifier.

          This table is created to reuse the existing
          mplsTunnelTable for MPLS based transport network
          tunnels also.

          Since the MPLS tunnel's Ingress/Egress LSR identifiers'
          size (Unsigned32) value is not compatible for
          MPLS-TP tunnel i.e. Global_ID::Node_ID of size 8 bytes and
          ICC_Operator_ID::Node_ID of size 12 bytes, there exists a
          need to map the Global_ID::Node_ID or ICC_Operator_ID::Node_ID
          with the local identifier of size 4 bytes (Unsigned32) value
          in order to index (Ingress/Egress LSR identifier)
          the existing mplsTunnelTable."

       ::= { mplsTeExtObjects 2 }

      mplsTunnelExtNodeConfigEntry OBJECT-TYPE
       SYNTAX        MplsTunnelExtNodeConfigEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
          "An entry in this table represents a mapping
          identification for the operator or service provider
          with node or LSR.

          As per [RFC6370], IP compatible mapping is represented
          as Global_ID::Node_ID.
```

```
          As per [RFC6923], the CC and the ICC form the ICC_Operator_ID
          as CC::ICC and ICC compatible mapping is represented
          as ICC_Operator_ID::Node_ID.

          Note: Each entry in this table should have a unique
          [Global_ID and Node_ID] or [CC::ICC and Node_ID] combination."
          INDEX { mplsTunnelExtNodeConfigLocalId }
          ::= { mplsTunnelExtNodeConfigTable 1 }

      MplsTunnelExtNodeConfigEntry ::= SEQUENCE {
            mplsTunnelExtNodeConfigLocalId     MplsExtendedTunnelId,
            mplsTunnelExtNodeConfigGlobalId    MplsGlobalId,
            mplsTunnelExtNodeConfigCcId        MplsCcId,
            mplsTunnelExtNodeConfigIccId       MplsIccId,
            mplsTunnelExtNodeConfigNodeId      MplsNodeId,
            mplsTunnelExtNodeConfigIccValid    TruthValue,
            mplsTunnelExtNodeConfigStorageType StorageType,
            mplsTunnelExtNodeConfigRowStatus   RowStatus

      }

      mplsTunnelExtNodeConfigLocalId  OBJECT-TYPE
         SYNTAX        MplsExtendedTunnelId
         MAX-ACCESS    not-accessible
         STATUS        current
         DESCRIPTION
           "This object is used in accommodating the bigger
            size Global_ID::Node_ID and/or the ICC_Operator_ID::Node_ID
            with lower size LSR identifier in order to index
            the mplsTunnelTable.

            The Local Identifier is configured between 0 and 16777215,
            as valid IP address range starts from 16777216(01.00.00.00).
            This range is chosen to determine whether the
            mplsTunnelTable's Ingress/Egress LSR-id is an IP address or
            Local identifier. If the configured range is not an
            IP address, the operator is expected to retrieve the
            complete information (Global_ID::Node_ID or
            ICC_Operator_ID::Node_ID) from
            mplsTunnelExtNodeConfigTable.
            This way, existing mplsTunnelTable is reused for
            bidirectional tunnel extensions for MPLS based transport
            networks.

            The Local Identifier allows the operator to assign
            a unique identifier to map Global_ID::Node_ID and/or
            ICC_Operator_ID::Node_ID. As this Local Identifier is unique
            within the node and the same syntax of this object can be
```

```
            used for MPLS-TE tunnel also, it is up to the operator/local
            management entity to choose non-conflicting value for
            indexing the MPLS and MPLS-TP tunnel entries."
         ::= { mplsTunnelExtNodeConfigEntry 1 }


      mplsTunnelExtNodeConfigGlobalId  OBJECT-TYPE
         SYNTAX        MplsGlobalId
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
           "This object indicates the Global Operator Identifier.
            This object has no meaning when
            mplsTunnelExtNodeConfigIccValid is set true."
         REFERENCE
               "MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]
                Section 3."
         ::= { mplsTunnelExtNodeConfigEntry 2 }

      mplsTunnelExtNodeConfigCcId OBJECT-TYPE
           SYNTAX      MplsCcId
           MAX-ACCESS  read-create
           STATUS      current
           DESCRIPTION
              "This object allows the operator or service provider to
               configure a unique MPLS-TP ITU-T Country Code (CC)
               either for Ingress ID or Egress ID.

               This object has no meaning when
               mplsTunnelExtNodeConfigIccValid is set false."
              REFERENCE
                 "MPLS-TP Identifiers Following ITU-T Conventions,
                 [RFC6923] Section 3"
         ::= { mplsTunnelExtNodeConfigEntry 3 }

         mplsTunnelExtNodeConfigIccId OBJECT-TYPE
              SYNTAX      MplsIccId
              MAX-ACCESS  read-create
              STATUS      current
              DESCRIPTION
                 "This object allows the operator or service provider to
                  configure a unique MPLS-TP ITU-T Carrier Code (ICC)
                  either for Ingress ID or Egress ID.

                 This object has no meaning when
                 mplsTunnelExtNodeConfigIccValid is set false."
              REFERENCE
                 "MPLS-TP Identifiers Following ITU-T Conventions,
```

```
                [RFC6923] Section 3"
        ::= { mplsTunnelExtNodeConfigEntry 4 }

        mplsTunnelExtNodeConfigNodeId  OBJECT-TYPE
           SYNTAX        MplsNodeId
           MAX-ACCESS    read-create
           STATUS        current
           DESCRIPTION
              "This object indicates the Node_ID within the scope
               of a Global_ID or ICC_Operator_ID."
           REFERENCE
               "MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]
                Section 4."
           ::= { mplsTunnelExtNodeConfigEntry 5 }

        mplsTunnelExtNodeConfigIccValid  OBJECT-TYPE
           SYNTAX        TruthValue
           MAX-ACCESS    read-create
           STATUS        current
           DESCRIPTION
              "Denotes whether or not this entry uses
               mplsTunnelExtNodeConfigCcId,
               mplsTunnelExtNodeConfigIccId and
               mplsTunnelExtNodeConfigNodeId for mapping
               the ICC based identifiers with the local identifier.
               Note that if this variable is set to false then the
               mplsTunnelExtNodeConfigGlobalId and
               mplsTunnelExtNodeConfigNodeId objects should have
               the valid information."
           DEFVAL { false }
             ::= { mplsTunnelExtNodeConfigEntry 6 }

        mplsTunnelExtNodeConfigStorageType OBJECT-TYPE
           SYNTAX        StorageType
           MAX-ACCESS    read-create
           STATUS        current
           DESCRIPTION
            "This variable indicates the storage type for this
             object.
             Conceptual rows having the value 'permanent'
             need not allow write-access to any columnar
             objects in the row."
           DEFVAL { volatile }
           ::= { mplsTunnelExtNodeConfigEntry 7 }

      mplsTunnelExtNodeConfigRowStatus OBJECT-TYPE
           SYNTAX        RowStatus
           MAX-ACCESS    read-create
```

```
        STATUS       current
        DESCRIPTION
           "This object allows the operator to create, modify,
            and/or delete a row in this table."
        ::= { mplsTunnelExtNodeConfigEntry 8 }


    -- End of MPLS Transport Profile Node configuration table

    -- Start of MPLS Transport Profile Node IP compatible
    -- mapping table

   mplsTunnelExtNodeIpMapTable OBJECT-TYPE
      SYNTAX        SEQUENCE OF MplsTunnelExtNodeIpMapEntry

      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
          "This read-only table allows the operator to retrieve
           the local identifier for a given Global_ID::Node_ID in an IP
           compatible operator environment.

           This table MAY be used in on-demand and/or proactive

           OAM operations to get the Ingress/Egress LSR identifier
           (Local Identifier) from Src-Global_Node_ID
           or Dst-Global_Node_ID. The Ingress and Egress LSR
           identifiers are used to retrieve the tunnel entry.

           This table returns nothing when the associated entry
           is not defined in mplsTunnelExtNodeConfigTable."
      ::= { mplsTeExtObjects 3 }

     mplsTunnelExtNodeIpMapEntry OBJECT-TYPE
      SYNTAX        MplsTunnelExtNodeIpMapEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
            "An entry in this table represents a mapping of
             Global_ID::Node_ID with the local identifier.

             An entry in this table is created automatically when
             the Local identifier is associated with Global_ID and
             Node_Id in the mplsTunnelExtNodeConfigTable.

             Note: Each entry in this table should have a unique
             Global_ID and Node_ID combination."
       INDEX { mplsTunnelExtNodeIpMapGlobalId,
```

```
            mplsTunnelExtNodeIpMapNodeId
          }
    ::= { mplsTunnelExtNodeIpMapTable 1 }

MplsTunnelExtNodeIpMapEntry ::= SEQUENCE {
      mplsTunnelExtNodeIpMapGlobalId    MplsGlobalId,
      mplsTunnelExtNodeIpMapNodeId      MplsNodeId,
      mplsTunnelExtNodeIpMapLocalId     MplsExtendedTunnelId
}

mplsTunnelExtNodeIpMapGlobalId  OBJECT-TYPE
   SYNTAX        MplsGlobalId
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
     "This object indicates the Global_ID."
   ::= { mplsTunnelExtNodeIpMapEntry 1 }

mplsTunnelExtNodeIpMapNodeId  OBJECT-TYPE
   SYNTAX        MplsNodeId
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
     "This object indicates the Node_ID within the

      operator."
   ::= { mplsTunnelExtNodeIpMapEntry 2 }

mplsTunnelExtNodeIpMapLocalId  OBJECT-TYPE
   SYNTAX        MplsExtendedTunnelId
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
     "This object contains an IP compatible local identifier
      which is defined in mplsTunnelExtNodeConfigTable."
   ::= { mplsTunnelExtNodeIpMapEntry 3 }

-- End MPLS Transport Profile Node IP compatible table

-- Start of MPLS Transport Profile Node ICC based table

mplsTunnelExtNodeIccMapTable OBJECT-TYPE
 SYNTAX        SEQUENCE OF MplsTunnelExtNodeIccMapEntry
 MAX-ACCESS    not-accessible
 STATUS        current
 DESCRIPTION
    "This read-only table allows the operator to retrieve
     the local identifier for a given ICC_Operator_ID::Node_ID
```

```
           in an ICC operator environment.

           This table MAY be used in on-demand and/or proactive
           OAM operations to get the Ingress/Egress LSR
           identifier (Local Identifier) from Src-ICC
           or Dst-ICC. The Ingress and Egress LSR
           identifiers are used to retrieve the tunnel entry.
           This table returns nothing when the associated entry
           is not defined in mplsTunnelExtNodeConfigTable."
       ::= { mplsTeExtObjects 4 }

      mplsTunnelExtNodeIccMapEntry OBJECT-TYPE
       SYNTAX        MplsTunnelExtNodeIccMapEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
             "An entry in this table represents a mapping of
              ICC_Operator_ID::Node_ID with the local identifier.

              An entry in this table is created automatically when
              the Local identifier is associated with
              ICC_Operator_ID::Node_ID in
              the mplsTunnelExtNodeConfigTable."
        INDEX { mplsTunnelExtNodeIccMapCcId,
                mplsTunnelExtNodeIccMapIccId,
                mplsTunnelExtNodeIccMapNodeId }
        ::= { mplsTunnelExtNodeIccMapTable 1 }


      MplsTunnelExtNodeIccMapEntry ::= SEQUENCE {
            mplsTunnelExtNodeIccMapCcId       MplsCcId,
            mplsTunnelExtNodeIccMapIccId      MplsIccId,
            mplsTunnelExtNodeIccMapNodeId     MplsNodeId,
            mplsTunnelExtNodeIccMapLocalId    MplsExtendedTunnelId
      }

      mplsTunnelExtNodeIccMapCcId OBJECT-TYPE
           SYNTAX      MplsCcId
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
              "This object allows the operator or service provider to
               configure a unique MPLS-TP ITU-T Country Code (CC)
               either for Ingress or Egress LSR ID.

               The CC is a string of two alphabetic characters
               represented with upper case letters (i.e., A-Z)."
           ::= { mplsTunnelExtNodeIccMapEntry 1 }
```

```
          mplsTunnelExtNodeIccMapIccId OBJECT-TYPE
               SYNTAX      MplsIccId
               MAX-ACCESS  not-accessible
               STATUS      current
               DESCRIPTION
                  "This object allows the operator or service provider
                   to configure a unique MPLS-TP ITU-T Carrier
                   Code (ICC) either for Ingress or Egress LSR ID.

                  The ICC is a string of one to six characters, each
                  character being either alphabetic (i.e.  A-Z) or
                  numeric (i.e. 0-9) characters. Alphabetic characters
                  in the ICC should be represented with upper case
                  letters."
          ::= { mplsTunnelExtNodeIccMapEntry 2 }

          mplsTunnelExtNodeIccMapNodeId  OBJECT-TYPE
             SYNTAX        MplsNodeId
             MAX-ACCESS    not-accessible
             STATUS        current
             DESCRIPTION
               "This object indicates the Node_ID within the
                ICC based operator."
          ::= { mplsTunnelExtNodeIccMapEntry 3}

      mplsTunnelExtNodeIccMapLocalId  OBJECT-TYPE
         SYNTAX        MplsExtendedTunnelId
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
           "This object contains an ICC based local identifier
            which is defined in mplsTunnelExtNodeConfigTable."
      ::= { mplsTunnelExtNodeIccMapEntry 4 }

   -- End MPLS Transport Profile Node ICC based table

   -- Start of MPLS Tunnel table extension

     mplsTunnelExtTable OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsTunnelExtEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
         "This table represents extensions to mplsTunnelTable
          in order to support MPLS-TP tunnels.

          As per MPLS-TP Identifiers [RFC6370], LSP_ID for IP based
          co-routed bidirectional tunnel,
```

```
          A1-{Global_ID::Node_ID::Tunnel_Num}::Z9-{Global_ID::
          Node_ID::Tunnel_Num}::LSP_Num

          LSP_ID for IP based associated bidirectional tunnel,
          A1-{Global_ID::Node_ID::Tunnel_Num::LSP_Num}::
          Z9-{Global_ID::Node_ID::Tunnel_Num::LSP_Num}

          mplsTunnelTable is reused for forming the LSP_ID
          as follows,

          Source Tunnel_Num is mapped with mplsTunnelIndex,
          Source Node_ID is mapped with
          mplsTunnelIngressLSRId, Destination Node_ID is
          mapped with mplsTunnelEgressLSRId LSP_Num is mapped with
          mplsTunnelInstance.

          Source Global_ID::Node_ID and/or ICC_Operator_ID::Node_ID and
          Destination Global_ID::Node_ID and/or ICC_Operator_ID::Node-ID
          are maintained in the mplsTunnelExtNodeConfigTable and
          mplsTunnelExtNodeConfigLocalId is used to create an entry
          in mplsTunnelTable."
       REFERENCE
             "MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]."
       ::= { mplsTeExtObjects 5 }

      mplsTunnelExtEntry OBJECT-TYPE
      SYNTAX        MplsTunnelExtEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
            "An entry in this table represents MPLS-TP
             specific additional tunnel configurations."
      INDEX {
        mplsTunnelIndex,
        mplsTunnelInstance,
        mplsTunnelIngressLSRId,
        mplsTunnelEgressLSRId
       }
       ::= { mplsTunnelExtTable 1 }

      MplsTunnelExtEntry ::= SEQUENCE {
           mplsTunnelExtOppositeDirPtr          RowPointer,
           mplsTunnelExtOppositeDirTnlValid     TruthValue,
           mplsTunnelExtDestTnlIndex            MplsTunnelIndex,
           mplsTunnelExtDestTnlLspIndex         MplsTunnelInstanceIndex,
           mplsTunnelExtDestTnlValid            TruthValue,
           mplsTunnelExtIngressLSRLocalIdValid  TruthValue,
           mplsTunnelExtEgressLSRLocalIdValid   TruthValue
```

```
      }

      mplsTunnelExtOppositeDirPtr  OBJECT-TYPE
         SYNTAX        RowPointer
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "This object points to the opposite direction tunnel entry."
      ::= { mplsTunnelExtEntry 1 }

      mplsTunnelExtOppositeDirTnlValid  OBJECT-TYPE
         SYNTAX        TruthValue
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "Denotes whether or not this tunnel uses
             mplsTunnelExtOppositeDirPtr for identifying the opposite
             direction tunnel information. Note that if this variable
             is set to true then the mplsTunnelExtOppositeDirPtr should
             point to the first accessible row of the valid opposite
             direction tunnel."
         DEFVAL { false }
           ::= { mplsTunnelExtEntry 2 }

      mplsTunnelExtDestTnlIndex  OBJECT-TYPE
         SYNTAX        MplsTunnelIndex
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "This object is applicable only for the bidirectional
             tunnel that has the forward and reverse LSPs in the
             different tunnel entries.

             The values of this object and the
             mplsTunnelExtDestTnlLspIndex object together can be used
             to identify an opposite direction LSP i.e. if the
             mplsTunnelIndex and mplsTunnelInstance hold the value
             for forward LSP, this object and
             mplsTunnelExtDestTnlLspIndex can be used to retrieve
             the reverse direction LSP and vice versa.

             This object and mplsTunnelExtDestTnlLspIndex values
             provide the first two indices of tunnel entry and
             the remaining indices can be derived as follows,
             the Ingress and Egress Identifiers should be
             swapped in order to index the other direction tunnel."
            ::= { mplsTunnelExtEntry 3 }
```

```
      mplsTunnelExtDestTnlLspIndex  OBJECT-TYPE
         SYNTAX        MplsTunnelInstanceIndex
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "This object is applicable only for the bidirectional
             tunnel that has the forward and reverse LSPs in the
             different tunnel entries. This object holds
             the instance index of the opposite direction tunnel."
            ::= { mplsTunnelExtEntry 4 }

      mplsTunnelExtDestTnlValid  OBJECT-TYPE
         SYNTAX        TruthValue
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "Denotes whether or not this tunnel uses
             mplsTunnelExtDestTnlIndex and
             mplsTunnelExtDestTnlLspIndex for identifying
             the opposite direction tunnel information. Note that if
             this variable is set to true then the
             mplsTunnelExtDestTnlIndex and
             mplsTunnelExtDestTnlLspIndex objects should have
             the valid opposite direction tunnel indices."
         DEFVAL { false }
           ::= { mplsTunnelExtEntry 5 }

      mplsTunnelExtIngressLSRLocalIdValid OBJECT-TYPE
         SYNTAX        TruthValue
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
          "This object denotes whether the mplsTunnelIngressLSRId
           contains the local value, which is used to reference
           the complete Ingress Global_ID::Node_ID or ICC_Operator_ID
           from the mplsTunnelExtNodeConfigTable.

           If this object is set to FALSE, mplsTunnelExtNodeConfigTable
           will not contain an entry to reference local identifier with
           Global_ID::Node_ID or ICC_Operator_ID::Node_ID value.

           This object is set to FALSE for legacy implementations like
           MPLS TE tunnels where mplsTunnelIngressId itself provides
           complete Ingress LSRId."
         REFERENCE
           "MPLS-TE-STD-MIB [RFC3812], Section 11.
            mplsTunnelIngressLSRId object in mplsTunnelTable."
         DEFVAL { false }
```

```
          ::= { mplsTunnelExtEntry 6 }

      mplsTunnelExtEgressLSRLocalIdValid OBJECT-TYPE
         SYNTAX        TruthValue
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
          "This object denotes whether the mplsTunnelEgressLSRId
           contains the local value, which is used to reference
           the complete Egress Global_ID::Node_ID or
           ICC_Operator_ID::Node_ID from
           the mplsTunnelExtNodeConfigTable.

           If this object is set to FALSE, mplsTunnelExtNodeConfigTable
           will not contain an entry to reference local identifier with
           Global_ID::Node_ID or ICC_Operator_ID::Node_ID value.

           This object is set to FALSE for legacy implementations like
           MPLS TE tunnels where mplsTunnelEgressId itself provides
           complete Egress LSRId."
         REFERENCE
           "MPLS-TE-STD-MIB [RFC3812], Section 11.
            mplsTunnelEgressLSRId object in mplsTunnelTable."
         DEFVAL { false }
           ::= { mplsTunnelExtEntry 7 }

       -- End of MPLS Tunnel table extension

      -- Module compliance.

      mplsTeExtCompliances
         OBJECT IDENTIFIER ::= { mplsTeExtConformance 1 }

      mplsTeExtGroups
         OBJECT IDENTIFIER ::= { mplsTeExtConformance 2 }

      -- Compliance requirement for fully compliant implementations.

      mplsTeExtModuleFullCompliance MODULE-COMPLIANCE
         STATUS current
         DESCRIPTION
              "Compliance statement for agents that provide full
                support the MPLS-TE-EXT-STD-MIB module."

         MODULE -- this module

            -- The mandatory group has to be implemented by all
            -- LSRs that originate/terminate MPLS-TP tunnels.
```

```
            -- In addition, depending on the type of tunnels
            -- supported, other groups become mandatory as

            -- explained below.

            MANDATORY-GROUPS    {
               mplsTunnelExtGroup
            }

            GROUP mplsTunnelExtIpOperatorGroup
            DESCRIPTION
                "This group is mandatory for devices which support
                 configuration of IP based identifier tunnels."

            GROUP mplsTunnelExtIccOperatorGroup
            DESCRIPTION
                "This group is mandatory for devices which support
                 configuration of ICC based tunnels."

             ::= { mplsTeExtCompliances 1 }

        -- Compliance requirement for read-only implementations.

        mplsTeExtModuleReadOnlyCompliance MODULE-COMPLIANCE
           STATUS current
           DESCRIPTION
               "Compliance statement for agents that only provide
                read-only support for MPLS-TE-EXT-STD-MIB module."

           MODULE -- this module

        MANDATORY-GROUPS    {
           mplsTunnelExtGroup
        }

        GROUP mplsTunnelExtIpOperatorGroup
        DESCRIPTION
            "This group is mandatory for devices which support
             configuration of IP based identifier tunnels."

        GROUP mplsTunnelExtIccOperatorGroup
        DESCRIPTION
            "This group is mandatory for devices which support
             configuration of ICC based tunnels."

        -- mplsTunnelExtTable

        OBJECT      mplsTunnelExtOppositeDirPtr
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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



         ::= { mplsTeExtCompliances 2 }

       -- Units of conformance.

       mplsTunnelExtGroup OBJECT-GROUP
          OBJECTS {
            mplsTunnelExtOppositeDirPtr,
            mplsTunnelExtOppositeDirTnlValid,
            mplsTunnelExtDestTnlIndex,
            mplsTunnelExtDestTnlLspIndex,
            mplsTunnelExtDestTnlValid,
            mplsTunnelExtIngressLSRLocalIdValid,
            mplsTunnelExtEgressLSRLocalIdValid
          }

          STATUS  current
          DESCRIPTION
               "Necessary, but not sufficient, set of objects to
                 implement tunnels.  In addition, depending on the
                 operating environment, the following groups are
                 mandatory."
          ::= { mplsTeExtGroups 1 }

     mplsTunnelExtIpOperatorGroup  OBJECT-GROUP
        OBJECTS { mplsTunnelExtNodeConfigLocalIdNext,
```

```
                mplsTunnelExtNodeConfigGlobalId,
                mplsTunnelExtNodeConfigNodeId,
                mplsTunnelExtNodeIpMapLocalId,
                mplsTunnelExtNodeConfigStorageType,
                mplsTunnelExtNodeConfigRowStatus
        }
        STATUS  current
        DESCRIPTION
             "Object(s) needed to implement IP compatible tunnels."
        ::= { mplsTeExtGroups 2 }

     mplsTunnelExtIccOperatorGroup  OBJECT-GROUP
        OBJECTS { mplsTunnelExtNodeConfigLocalIdNext,
                  mplsTunnelExtNodeConfigCcId,
                  mplsTunnelExtNodeConfigIccId,
                  mplsTunnelExtNodeConfigNodeId,
                  mplsTunnelExtNodeConfigIccValid,
                  mplsTunnelExtNodeIccMapLocalId,
                  mplsTunnelExtNodeConfigStorageType,
                  mplsTunnelExtNodeConfigRowStatus
        }
        STATUS  current
        DESCRIPTION
             "Object(s) needed to implement ICC based tunnels."
        ::= { mplsTeExtGroups 3 }

     -- MPLS-TE-EXT-STD-MIB module ends
     END
```

14. Security Consideration

This document follows the security consideration mentioned in the section 12 of [RFC3812]. These security considerations are also applicable to the MIB objects and tables defined in this draft, which are identified as below.

- The common objects mplsIdGlobalId, mplsIdNodeId, mplsIdCc, and mplsIdIcc are used to define the identity of an MPLS-TP node for OAM purposes. If write-access is allowed to these objects it offers the possibility for incorrect values to be entered that will confuse the information returned by OAM functions and possibly prevent OAM from operating correctly. Furthermore, there is the possibility of inducing one node to impersonate another with confusing results.

- mplsTunnelExtNodeConfigTable, mplsTunnelExtTable and mplsXCExtTable collectively contain objects to provision MPLS-TP

tunnels, tunnel hops, and tunnel resources.

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

- mplsTunnelExtNodeConfigTable, mplsTunnelExtTable, and mplsXCExtTable collectively show the MPLS-TP tunnel network topology characteristics. If an Administrator does not want to reveal this information, then these tables should be considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

15. IANA Considerations

As described in [RFC4221], [RFC6639] and as requested in the MPLS-TC-STD-MIB [RFC3811], MPLS related standards track MIB modules should be rooted under the mplsStdMIB subtree. There are 4 MPLS MIB Modules contained in this document, each of the following "IANA Considerations" subsections requests IANA for a new assignment under the mplsStdMIB subtree. New assignments can only be made via a

Standards Action as specified in [RFC5226].

15.1. IANA Considerations for MPLS-TC-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-TC-EXT-STD-MIB module specified in this document.

15.2. IANA Considerations for MPLS-ID-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-ID-STD-MIB module specified in this document.

15.3. IANA Considerations for MPLS-LSR-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-LSR-EXT-STD-MIB module specified in this document.

15.4. IANA Considerations for MPLS-TE-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-TE-EXT-STD-MIB module specified in this document.

16. References

16.1. 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.

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

[RFC3811] Nadeau, T., Ed., and J. Cucchiara, Ed., "Definitions of

Textual Conventions (TCs) 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.

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

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, September 2011.

[RFC6923] Winter, R., Gray, E., Helvoort, H., and M. Betts, "MPLS-TP Identifiers Following ITU-T Conventions", RFC 6923, May 2013

[T.50] "International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information exchange", ITU-T Recommendation T.50, September 1992.

16.2. Informative References

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

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3826] Blumenthal, U., F. Maino and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.

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

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

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.

[RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed.,Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, September 2009.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, July 2011.

[RFC6639] Venkatesan, M., King, D., "Multiprotocol Label Switching Transport Profile (MPLS-TP) MIB-Based Management Overview", RFC 6639, June 2012

17. Acknowledgments

The authors would like to thank Francesco Fondelli, Josh Littlefield, Agrahara Kiran Koushik, Metrri Jain, Muly Ilan, Randy Presuhn, Elwyn Davies, Tom Taylor and Pete Resnick for their valuable review and comments. A special thanks to Joan Cucchiara and Adrian Farrel for really getting the MIB modules into shape.

18. Authors’ Addresses

Venkatesan Mahalingam
Dell Inc.
5450 Great America Parkway,
Santa Clara, CA 95054, USA
Email: venkat.mahalingams@gmail.com

Sam Aldrin
Huawei Technologies
2330 Central Express Way,
Santa Clara, CA 95051, USA
Email: aldrin.ietf@gmail.com

Thomas D. Nadeau

Brocade
Email: tnadeau@lucidvision.com

Kannan KV Sampath
Redeem
India
Email: kannankvs@gmail.com