Network Working Group
INTERNET-DRAFT
Intended Status: Standards Track
Expires: June 15, 2012

M.Venkatesan
Kannan KV Sampath
Aricent
Sam K. Aldrin
Huawei Technologies
Thomas D. Nadeau
CA Technologies

December 15, 2011

# MPLS-TP Traffic Engineering (TE) Management Information Base (MIB) draft-ietf-mpls-tp-te-mib-01.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 managed objects of Tunnels, Identifiers, Label Switch Router and Textual conventions for Multiprotocol Label Switching (MPLS) based Transport Profile (TP).

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 managed objects of Tunnels, Identifiers, Label Switch Router and Textual conventions for Multiprotocol Label Switching (MPLS) based Transport Profile (TP).

This MIB module should be used in conjunction with the MPLS traffic Engineering MIB [RFC3812] and companion document MPLS Label Switch Router MIB [RFC3813] for MPLS based traffic engineering configuration and management.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC2119.

## 2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC3410 [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 STD58, RFC2580.

## 3. Overview

### 3.1 Conventions used in this document

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

### 3.2 Terminology

This document uses terminology from the MPLS architecture document [RFC3031], MPLS Traffic Engineering Management information [RFC3812], MPLS Label Switch Router MIB [RFC3813] and MPLS-TP Identifiers document [TPIDS].

### 3.3 Acronyms

GMPLS: Generalized Multi-Protocol Label Switching
ICC: ITU Carrier Code
IP: Internet Protocol
LSP: Label Switching Path
LSR: Label Switching Router
MIB: Management Information Base
MPLS: Multi-Protocol Label Switching
MPLS-TP: MPLS Transport Profile
OSPF: Open Shortest Path First
PW: Pseudowire
TE: Traffic Engineering
TP: Transport Profile

## 4. Motivations

The existing MPLS TE [RFC3812] and GMPLS MIBs [RFC4802] do not support the transport network requirements of NON-IP based management and static bidirectional tunnels.

## 5. Feature List

The MPLS transport profile MIB module is designed to satisfy the following requirements and constraints:

The MIB module supports point-to-point, co-routed bi-directional associated bi-directional tunnels.

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

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

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

- A node configuration table (mplsNodeConfigTable) is used to translate the Global_Node_ID or ICC to the local identifier in order to index mplsTunnelTable.

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

## 6. Brief description of MIB Objects

The objects described in this section support the functionality described in documents [RFC5654] and [TPIDS]. The tables support

both IP compatible and ICC based tunnel configurations.

## 6.1. mplsNodeConfigTable

The mplsNodeConfigTable is used to assign a local identifier for a given ICC or Global_Node_ID combination as defined in [TPIDS]. An 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. In the IP compatible mode, Global_Node_ID, is used to uniquely identify a node.

Each ICC or Global_Node_ID contains one unique entry in the table representing a node. 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 ICC and Destination ICC.

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

The mplsNodeConfigTable will be used to store an entry for ICC or Global_Node_ID with a local identifier to be used as LSR ID in mplsTunnelTable. 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].

## 6.2. mplsNodeIpMapTable

The read-only mplsNodeIpMaptable is used to query the local identifier assigned and stored in mplsNodeConfigTable for a given Global_Node_ID. In order to query the local identifier, in the IP compatible mode, this table is indexed with Global_Node_ID. In the IP compatible mode for a TP tunnel, Global_Node_ID is used.

A separate query is made to get the local identifier of both Ingress and Egress Global_Node_ID identifiers. These local

identifiers are used as mplsTunnelIngressLSRId and mplsTunnelEgressLSRId, while indexing mplsTunnelTable.

### 6.3. mplsNodeIccMapTable

The read-only mplsNodeIccMapTable is used to query the local identifier assigned and stored in the mplsNodeConfigTable for a given ICC.

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

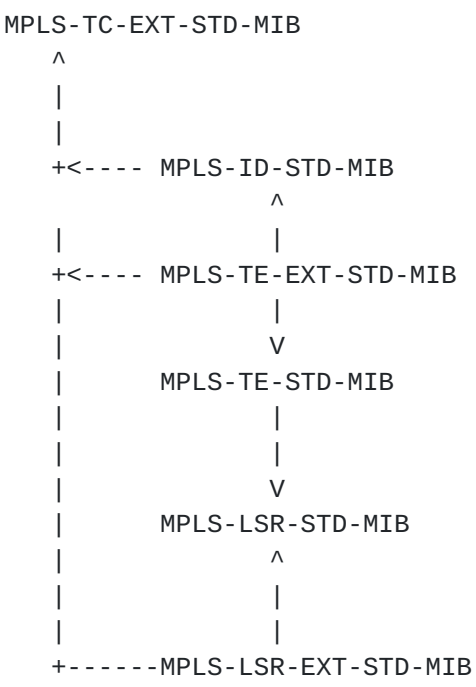
### 6.4.mplsTunnelExtTable

mplsTunnelExtTable extends the mplsTunnelTable to add MPLS-TP tunnel specific additional objects. All the additional attributes specific to TP tunnel are contained in this extended table and could be accessed with the mplsTunnelTable indices.

## 7. MIB Module 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.




Thus:

- All the new MPLS extension MIB modules depend on MPLS-TC-EXT-STD-MIB.

- MPLS-TE-STD-MIB [RFC3812] contains references to objects in MPLS-ID-STD-MIB.

- 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].

MPLS-TE-STD-MIB [RFC 3812] is extended by MPLS-TE-EXT-STD-MIB mib module for associating the reverse direction tunnel information.

Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsTunnelExtTable has the same index values as the in the mplsTunnelTable.

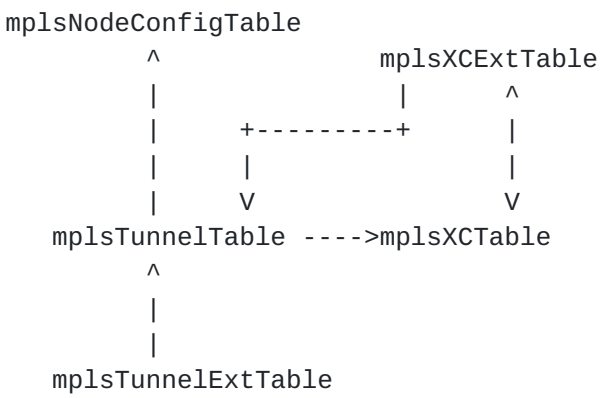
MPLS-LSR-STD-MIB [RFC 3813] is extended by MPLS-LSR-EXT-STD-MIB mib module for pointing back to the tunnel entry for easy tunnel access from XC entry.

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 in the mplsXCTable.

## 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.




An existing mplsTunnelTable uses the mplsNodeConfigTable table to map the Global_Node_ID and/or ICC 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 is extended from mplsLsrXCTable to provide backward reference to tunnel entry.

## 9. Example of MPLS-TP tunnel setup

In this section, we provide an example of the IP based MPLS-TP co-routed bidirectional tunnel setup. This example provides the usage of MPLS-TP Tunnel MIB along with the extended new 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 configuration on a head-end LSR to setup a static MPLS-TP tunnel. Only relevant objects which are applicable for MPLS-TP tunnel are illustrated here.

In mplsNodeConfigTable:

```
{
-- Non-IP Ingress LSR-Id (Index to the table)
  mplsNodeConfigLocalId              = 1,
```

```
  mplsNodeConfigGlobalId            = 1234,
  mplsNodeConfigNodeId              = 10,
-- Mandatory parameters needed to activate the row go here
  mplsNodeConfigRowStatus         = createAndGo (4)

-- Non-IP Egress LSR-Id (Index to the table)
  mplsNodeConfigLocalId             = 2,
  mplsNodeConfigGlobalId            = 1234,
  mplsNodeConfigNodeId              = 20,
-- Mandatory parameters needed to activate the row go here
  mplsNodeConfigRowStatus         = createAndGo (4)
}
```

This will create an entry in the mplsNodeConfigTable for a Global_Node_ID. A separate entry is made for both Ingress LSR and Egress LSR.

The following read-only mplsNodeIpMapTable table is populated automatically upon creating an entry in mplsNodeConfigTable and this table is used to retrieve the local identifier for the given Global_Node_ID.

In mplsNodeIpMapTable:

```
{
-- Global_ID (Index to the table)
  mplsNodeIpMapGlobalId             = 1234,
-- Node Identifier (Index to the table)
  mplsNodeIpMapNodeId               = 10,
  mplsNodeIpMapLocalId              = 1

-- Global_ID (Index to the table)
  mplsNodeIpMapGlobalId             = 1234,
-- Node Identifier (Index to the table)
  mplsNodeIpMapNodeId               = 20,
  mplsNodeIpMapLocalId              = 2
}
```

The following denotes the configured tunnel "head" entry:

In mplsTunnelTable:

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

```
      -- Local map number created in mplsNodeConfigTable for Egress
         LSR-Id
        mplsTunnelEgressLSRId        = 2,
        mplsTunnelName               = "TP forward LSP",
        mplsTunnelDescr              = "East to West",
        mplsTunnelIsIf               = true (1),
      --  RowPointer MUST point to the first accessible column
        mplsTunnelXCPointer          =
                                   mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.12,
        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)
      }

       In mplsTunnelTable:

      {
        mplsTunnelIndex              = 1,
        mplsTunnelInstance           = 2,
      -- Local map number created in mplsNodeConfigTable for Ingress
         LSR-Id
        mplsTunnelIngressLSRId       = 1,
      -- Local map number created in mplsNodeConfigTable for Egress
         LSR-Id
        mplsTunnelEgressLSRId        = 2,
        mplsTunnelName               = "TP reverse LSP",
        mplsTunnelDescr              = "West to East",
        mplsTunnelIsIf               = true (1),
      --  RowPointer MUST point to the first accessible column
        mplsTunnelXCPointer          =
                                   mplsXCLspId.4.0.0.0.1.4.0.0.0.16.1.0,
        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)
}

Now the TP specific Tunnel parameters are configured in
the extended Tunnel table

In mplsTunnelExtTable:
{
  Index = same as one used for mplsTunnelTable,
  -- As per [TPIDS] LSP_ID is defined as follows,
  -- For corouted bidirectional tunnel
  -- LSP_ID => East-Global_Node_ID::East-Tunnel_Num::
  --           West-Global_Node_ID::West-Tunnel_Num::LSP_Num
  -- LSP_ID of this tunnel: 1234_10::1::1234_20::1::0
  -- Where,
  -- LSP_Num - 0 indicates the configured head end tunnel.

  -- West tunnel number is assigned in the destination
  -- tunnel index,
  -- single LSP number is common for both forward and reverse
  -- directions, as the single tunnel head entry originates
  -- both the forward and reverse LSPs.
  -- mplsTunnelExtDestTnlIndex = West-Tunnel_Num
  -- mplsTunnelExtDestTnlLspIndex = LSP_Num

  mplsTunnelExtDestTnlIndex       = 1,
  mplsTunnelExtDestTnlLspIndex    = 0

   -- For associated bidirectional tunnel
   -- LSP_ID => East-Global_Node_ID::East-Tunnel_Num::
   --           East-LSP_Num::West-Global_Node_ID::
   --           West-Tunnel_Num::West-LSP_Num
   -- West tunnel number is assigned in the destination
   -- tunnel index, since the head end tunnel is different for
   -- both the forward and reverse direction LSPs,
   -- Destination LSP index points the reverse direction LSP
   -- in a different tunnel.
   -- mplsTunnelExtDestTnlIndex = West-Tunnel_Num
   -- mplsTunnelExtDestTnlLspIndex = West-LSP_Num
```

```
}
```

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

For the forward direction.

In mplsOutSegmentTable:

```
{
   mplsOutSegmentIndex          = 0x00000012,
   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)
}
```

For the reverse direction.

In mplsInSegmentTable:

```
{
   mplsInSegmentIndex           = 0x00000016
   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.

In mplsXCTable:

```
{
   mplsXCIndex                = 0x01,
   mplsXCInSegmentIndex       = 0x00000000,
   mplsXCOutSegmentIndex      = 0x00000012,
   mplsXCLspId                = 0x0102 -- unique ID
   -- only a single outgoing label
   mplsXCLabelStackIndex      = 0x00,
   mplsXCRowStatus            = createAndGo(4)
```

```
    }

    In mplsXCTable:
    {
       mplsXCIndex               = 0x01,
       mplsXCInSegmentIndex      = 0x00000016,
       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.

    First for the forward direction:

    In mplsXCExtTable
    {
      -- Back pointer from XC table to Tunnel table
      mplsXCExtTunnelPointer         = mplsTunnelName.1.1.1.2
    }

    Next for the reverse direction:

    In mplsXCExtTable
    {
      -- Back pointer from XC table to Tunnel table
      mplsXCExtTunnelPointer         = mplsTunnelName.1.2.1.2
    }
```

## 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
         "201106160000Z" -- June 16, 2011
      ORGANIZATION
         "Multiprotocol Label Switching (MPLS) Working Group"
      CONTACT-INFO
         "
               Venkatesan Mahalingam
               Aricent,
               India
          Email: venkat.mahalingams@gmail.com

               Kannan KV Sampath
               Aricent,
               India
          Email: Kannan.Sampath@aricent.com

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

               Thomas D. Nadeau
               CA Technologies
               273 Corporate Drive, Portsmouth, NH, USA
          Email: thomas.nadeau@ca.com
        "
      DESCRIPTION
          "Copyright (c) 2011 IETF Trust and the persons identified
           as the document authors.  All rights reserved.

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

      -- Revision history.

      REVISION
         "201106160000Z" -- June 16, 2011
      DESCRIPTION
           "MPLS Textual Convention Extensions"

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

   MplsGlobalId ::= TEXTUAL-CONVENTION
```

```
      STATUS      current
      DESCRIPTION
           "This object contains the Textual Convention of 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).

            It is expected that the Global_ID will be derived from
            the globally unique ASN of the autonomous system hosting
            the PEs containing the actual AIIs.
            The presence of a Global_ID based on the operator's
            ASN ensures that the AII will be globally unique.

            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.
            Further ASN 0 is reserved.  A Global_ID of zero means that
            no Global_ID is present.  Note that a Global_ID of zero
            is limited to entities contained within a single operator
            and MUST NOT be used across an NNI.
            A non-zero Global_ID MUST be derived from an ASN owned by
            the operator."
      SYNTAX  OCTET STRING (SIZE (4))

   MplsNodeId ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "d"
      STATUS      current
      DESCRIPTION
           "The Node_ID is assigned within the scope of the Global_ID.
            The value 0(or 0.0.0.0 in dotted decimal notation) is
            reserved and MUST NOT be used.

            When IPv4 addresses are in use, the value of this object can
            be derived from the LSR's /32 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 reach ability is not needed, the 32-bit
            Node_ID is not required to have any association
            with the IPv4 address space."
      SYNTAX  Unsigned32

   MplsIccId ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
           "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."
      SYNTAX  OCTET STRING (SIZE (1..6))

   MplsLocalId ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "d"
      STATUS      current
      DESCRIPTION
         "This textual convention is used in accommodating the bigger
          size Global_Node_ID and/or ICC with lower size LSR identifier
          in order to index the mplsTunnelTable.

          The Local Identifier is configured between 1 and 16777215,
          as valid IP address range starts from 16777216 (01.00.00.00).
          This range is chosen to identify the mplsTunnelTable's
          Ingress/Egress LSR-id is IP address or Local identifier,
          if the configured range is not IP address, administrator is
          expected to retrieve the complete information (Global_Node_ID
          or ICC) from mplsNodeConfigTable. This way, existing
          mplsTunnelTable is reused for bidirectional tunnel extensions
          for MPLS based transport networks.

          This Local Identifier allows the administrator to assign
          a unique identifier to map Global_Node_ID and/or ICC."
      SYNTAX  Unsigned32(1..16777215)

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

## 11. MPLS Identifier MIB definitions

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

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

   mplsIdStdMIB MODULE-IDENTITY
     LAST-UPDATED
```

```
      "201106160000Z" -- June 16, 2011
    ORGANIZATION
      "Multiprotocol Label Switching (MPLS) Working Group"

    CONTACT-INFO
      "
            Venkatesan Mahalingam
            Aricent,
            India


       Email: venkat.mahalingams@gmail.com

            Kannan KV Sampath
            Aricent,
            India
       Email: Kannan.Sampath@aricent.com

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

            Thomas D. Nadeau
            CA Technologies
            273 Corporate Drive, Portsmouth, NH, USA
       Email: thomas.nadeau@ca.com
     "
    DESCRIPTION
       "Copyright (c) 2011 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
      "201106160000Z" -- June 16, 2011
    DESCRIPTION
        "MPLS identifiers mib object extension"

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

    -- traps
    mplsIdNotifications OBJECT IDENTIFIER ::= { mplsIdStdMIB 0 }
    -- tables, scalars
```

```
     mplsIdObjects       OBJECT IDENTIFIER ::= { mplsIdStdMIB 1 }
     -- conformance
     mplsIdConformance   OBJECT IDENTIFIER ::= { mplsIdStdMIB 2 }


     -- MPLS common objects

    mplsGlobalId OBJECT-TYPE
        SYNTAX      MplsGlobalId
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION



            "This object allows the administrator to assign a unique
             operator identifier also called MPLS-TP Global_ID."
        REFERENCE
            "MPLS-TP Identifiers [TPIDS]."
       ::= { mplsIdObjects 1 }

   mplsIcc 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 a
             network."
        REFERENCE
            "MPLS-TP Identifiers [TPIDS]."
       ::= { mplsIdObjects 2 }

   mplsNodeId 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."

        REFERENCE
            "MPLS-TP Identifiers [TPIDS]."
       ::= { mplsIdObjects 3 }

    -- Module compliance.
```

```
   mplsIdGroups
      OBJECT IDENTIFIER ::= { mplsIdConformance 1 }


   mplsIdCompliances
      OBJECT IDENTIFIER ::= { mplsIdConformance 2 }

   -- Compliance requirement for fully compliant implementations.

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


      MODULE -- this module

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

         MANDATORY-GROUPS    {
            mplsIdScalarGroup
         }

          ::= { mplsIdCompliances 1 }

   -- Compliance requirement for read-only implementations.

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

      MODULE -- this module

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

         MANDATORY-GROUPS    {
            mplsIdScalarGroup
         }

        ::= { mplsIdCompliances 2 }

     -- Units of conformance.
```

```
    mplsIdScalarGroup OBJECT-GROUP
      OBJECTS { mplsGlobalId,
                mplsNodeId,
                mplsIcc

      }
      STATUS  current
      DESCRIPTION
          "Scalar object needed to implement MPLS TP path."
      ::= { mplsIdGroups 1 }

    -- 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,
      mplsInSegmentGroup, mplsOutSegmentGroup, mplsXCGroup,
      mplsPerfGroup, mplsLsrNotificationGroup
         FROM MPLS-LSR-STD-MIB;                             -- [RFC3813]

   mplsLsrExtStdMIB MODULE-IDENTITY
      LAST-UPDATED
         "201106160000Z" -- June 16, 2011
      ORGANIZATION
         "Multiprotocol Label Switching (MPLS) Working Group"
      CONTACT-INFO
         "
                Venkatesan Mahalingam
                Aricent,
                India
          Email: venkat.mahalingams@gmail.com

                Kannan KV Sampath
                Aricent,
                India
          Email: Kannan.Sampath@aricent.com
```

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

Email: aldrin.ietf@gmail.com

Thomas D. Nadeau
CA Technologies
273 Corporate Drive, Portsmouth, NH, USA
Email: thomas.nadeau@ca.com

```
      "
   DESCRIPTION
       "Copyright (c) 2011 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
      "201106160000Z" -- June 16, 2011
   DESCRIPTION
        "MPLS LSR specific mib objects extension"

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

-- traps
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 [RFC 3813] 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 extends the cross connect
           information represented by an entry in
           the mplsXCTable in MPLS-LSR-STD-MIB [RFC 3813] through
           a sparse augmentation.  An entry can be created by a network
           administrator 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
   }

   mplsXCExtTunnelPointer OBJECT-TYPE
       SYNTAX        RowPointer
       MAX-ACCESS    read-create
       STATUS        current
       DESCRIPTION
          "This object indicates the back pointer to the tunnel entry
           segment.  This object cannot be modified if
           mplsXCRowStatus for the corresponding entry in the
           mplsXCTable is active(1)."
       REFERENCE
          "1. Multiprotocol Label Switching (MPLS) Label Switching
           Router (LSR) Management Information Base (MIB), RFC 3813."
    ::= { mplsXCExtEntry 1 }

    mplsLsrExtGroups
       OBJECT IDENTIFIER ::= { mplsLsrExtConformance 1 }
    mplsLsrExtCompliances
       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,
       mplsPerfGroup,
       mplsLsrNotificationGroup
     }

     MODULE -- this module




     MANDATORY-GROUPS    {
       mplsXCExtGroup
     }

     OBJECT      mplsXCExtTunnelPointer
     SYNTAX      RowPointer
     MIN-ACCESS  read-only
     DESCRIPTION
         "The only valid value for Tunnel Pointer is mplsTunnelTable
          entry."

     ::= { 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,

        mplsXCGroup,
        mplsPerfGroup
    }

    MODULE -- this module

    MANDATORY-GROUPS {
       mplsXCExtGroup
    }

    OBJECT      mplsXCExtTunnelPointer
    SYNTAX      RowPointer
    MIN-ACCESS  read-only
    DESCRIPTION
       "The only valid value for Tunnel Pointer is mplsTunnelTable
        entry."

    ::= { mplsLsrExtCompliances 2 }



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

   -- 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, Unsigned32, Gauge32,
          NOTIFICATION-TYPE
```

```
          FROM SNMPv2-SMI                                   -- [RFC2578]
       MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
          FROM SNMPv2-CONF                                  -- [RFC2580]
       RowStatus, StorageType
          FROM SNMPv2-TC                                    -- [RFC2579]
       MplsLocalId, MplsGlobalId, MplsNodeId, MplsIccId
          FROM MPLS-TC-EXT-STD-MIB

       mplsStdMIB, MplsTunnelIndex, MplsTunnelInstanceIndex
          FROM MPLS-TC-STD-MIB                              -- [RFC3811]
       mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
       mplsTunnelEgressLSRId
          FROM MPLS-TE-STD-MIB                              -- [RFC3812]
       ;

    mplsTeExtStdMIB MODULE-IDENTITY
       LAST-UPDATED
          "201106160000Z" -- June 16, 2011
       ORGANIZATION
          "Multiprotocol Label Switching (MPLS) Working Group"
       CONTACT-INFO
          "
                Venkatesan Mahalingam
                Aricent,
                India
           Email: venkat.mahalingams@gmail.com

                Kannan KV Sampath
                Aricent,


                India
           Email: Kannan.Sampath@aricent.com

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

                Thomas D. Nadeau
                CA Technologies
                273 Corporate Drive, Portsmouth, NH, USA
           Email: thomas.nadeau@ca.com
         "
       DESCRIPTION
           "Copyright (c) 2011 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
           "201106160000Z" -- June 16, 2011

        DESCRIPTION
             "MPLS TE mib objects extension"

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

     -- Top level components of this MIB module.

     -- traps
     mplsTeExtNotifications OBJECT IDENTIFIER ::= { mplsTeExtStdMIB 0 }
     -- tables, scalars
     mplsTeExtObjects       OBJECT IDENTIFIER ::= { mplsTeExtStdMIB 1 }
     -- conformance
     mplsTeExtConformance   OBJECT IDENTIFIER ::= { mplsTeExtStdMIB 2 }

    -- Start of MPLS Transport Profile Node configuration table
      mplsNodeConfigTable OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsNodeConfigEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
             "This table allows the administrator to map a node or LSR
              Identifier (IP compatible [Global_Node_ID] or ICC) 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_Node_Id of size 8 bytes and
              ICC of size 6 bytes, there exists a need to map the
              Global_Node_ID or ICC with the local identifier of size
              4 bytes (Unsigned32) value in order
              to index (Ingress/Egress LSR identifier)
              the existing mplsTunnelTable."
       ::= { mplsTeExtObjects 1 }

      mplsNodeConfigEntry OBJECT-TYPE
       SYNTAX        MplsNodeConfigEntry
```

```
       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 [TPIDS], this mapping is

              represented as Global_Node_ID or ICC.

              Note: Each entry in this table should have a unique
              Global_ID and Node_ID combination."
        INDEX { mplsNodeConfigLocalId }
        ::= { mplsNodeConfigTable 1 }

      MplsNodeConfigEntry ::= SEQUENCE {
            mplsNodeConfigLocalId     MplsLocalId,
            mplsNodeConfigGlobalId    MplsGlobalId,
            mplsNodeConfigNodeId      MplsNodeId,
            mplsNodeConfigIccId       MplsIccId,
            mplsNodeConfigRowStatus   RowStatus,
            mplsNodeConfigStorageType StorageType

      }

      mplsNodeConfigLocalId  OBJECT-TYPE
         SYNTAX        MplsLocalId
         MAX-ACCESS    not-accessible
         STATUS        current
         DESCRIPTION
           "This object allows the administrator to assign a unique
            local identifier to map Global_Node_ID or ICC."
         ::= { mplsNodeConfigEntry 1 }


      mplsNodeConfigGlobalId  OBJECT-TYPE
         SYNTAX        MplsGlobalId
         MAX-ACCESS    read-write
         STATUS        current
         DESCRIPTION
           "This object indicates the Global Operator Identifier.
            This object value should be zero when
            mplsNodeConfigIccId is configured with non-null value."
         REFERENCE
               "MPLS-TP Identifiers [TPIDS]."
         ::= { mplsNodeConfigEntry 2 }
```

```
      mplsNodeConfigNodeId  OBJECT-TYPE
         SYNTAX        MplsNodeId
         MAX-ACCESS    read-write
         STATUS        current
         DESCRIPTION
            "This object indicates the Node_ID within the operator.
            This object value should be zero when mplsNodeConfigIccId
            is configured with non-null value."
         REFERENCE

               "MPLS-TP Identifiers [TPIDS]."
         ::= { mplsNodeConfigEntry 3 }

      mplsNodeConfigIccId OBJECT-TYPE
           SYNTAX      MplsIccId
           MAX-ACCESS  read-write
           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 value should be zero when
                mplsNodeConfigGlobalId and mplsNodeConfigNodeId are
                assigned with non-zero value."
           REFERENCE
               "MPLS-TP Identifiers [TPIDS]."
         ::= { mplsNodeConfigEntry 4 }

      mplsNodeConfigRowStatus OBJECT-TYPE
         SYNTAX        RowStatus
         MAX-ACCESS    read-create
         STATUS        current
         DESCRIPTION
            "This object allows the administrator to create, modify,
             and/or delete a row in this table."
         ::= { mplsNodeConfigEntry 5 }


      mplsNodeConfigStorageType 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 }
        ::= { mplsNodeConfigEntry 6 }

     -- End of MPLS Transport Profile Node configuration table

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

    mplsNodeIpMapTable OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsNodeIpMapEntry

       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
             "This read-only table allows the administrator to retrieve
              the local identifier for a given Global_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 and 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 mplsNodeConfigTable."
       ::= { mplsTeExtObjects 2 }

      mplsNodeIpMapEntry OBJECT-TYPE
       SYNTAX        MplsNodeIpMapEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
             "An entry in this table represents a mapping of
              Global_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 mplsNodeConfigTable.




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

```
        ::= { mplsNodeIpMapTable 1 }

      MplsNodeIpMapEntry ::= SEQUENCE {
            mplsNodeIpMapGlobalId    MplsGlobalId,
            mplsNodeIpMapNodeId      MplsNodeId,
            mplsNodeIpMapLocalId     MplsLocalId
      }

      mplsNodeIpMapGlobalId  OBJECT-TYPE
         SYNTAX        MplsGlobalId
         MAX-ACCESS    not-accessible

         STATUS        current
         DESCRIPTION
           "This object indicates the Global_ID."
         ::= { mplsNodeIpMapEntry 1 }

      mplsNodeIpMapNodeId  OBJECT-TYPE
         SYNTAX        MplsNodeId
         MAX-ACCESS    not-accessible
         STATUS        current
         DESCRIPTION
           "This object indicates the Node_ID within the
            operator."
         ::= { mplsNodeIpMapEntry 2 }

      mplsNodeIpMapLocalId  OBJECT-TYPE
         SYNTAX        MplsLocalId
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
           "This object contains an IP compatible local identifier
            which is defined in mplsNodeConfigTable."
         ::= { mplsNodeIpMapEntry 3 }

      -- End MPLS Transport Profile Node IP compatible table

      -- Start of MPLS Transport Profile Node ICC based table

      mplsNodeIccMapTable OBJECT-TYPE
       SYNTAX        SEQUENCE OF MplsNodeIccMapEntry
       MAX-ACCESS    not-accessible
       STATUS        current
       DESCRIPTION
             "This read-only table allows the administrator to retrieve
              the local identifier for a given ICC operator 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 and 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 mplsNodeConfigTable."
     ::= { mplsTeExtObjects 3 }

   mplsNodeIccMapEntry OBJECT-TYPE
     SYNTAX        MplsNodeIccMapEntry

     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
           "An entry in this table represents a mapping of ICC with
            the local identifier.

            An entry in this table is created automatically when
            the Local identifier is associated with ICC in
            the mplsNodeConfigTable."
      INDEX { mplsNodeIccMapIccId }
      ::= { mplsNodeIccMapTable 1 }

   MplsNodeIccMapEntry ::= SEQUENCE {
         mplsNodeIccMapIccId      MplsIccId,
         mplsNodeIccMapLocalId    MplsLocalId
   }

   mplsNodeIccMapIccId 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."
        ::= { mplsNodeIccMapEntry 1 }

   mplsNodeIccMapLocalId  OBJECT-TYPE
      SYNTAX        MplsLocalId
      MAX-ACCESS    read-only
```

```
        STATUS       current
        DESCRIPTION
          "This object contains an ICC based local identifier
           which is defined in mplsNodeConfigTable."
        ::= { mplsNodeIccMapEntry 2 }

   -- 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 MPLS-TP specific extensions to
              mplsTunnelTable.

              As per MPLS-TP Identifiers [TPIDS] draft,  LSP_ID is

              Src-Global_Node_ID::Src-Tunnel_Num::Dst-Global_Node_ID::
              Dst-Tunnel_Num::LSP_Num for IP operator and

              Src-ICC::Src-Tunnel_Num::Dst-ICC::Dst-Tunnel_Num::LSP_Num
              for ICC operator,

              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_Node_ID and/or ICC and Destination
              Global_Node_ID and/or ICC are maintained in the
              mplsNodeConfigTable and mplsNodeConfigLocalId is
              used to create an entry in mplsTunnelTable."
       REFERENCE
             "MPLS-TP Identifiers [TPIDS]."
       ::= { mplsTeExtObjects 4 }

      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 {
            mplsTunnelExtDestTnlIndex   MplsTunnelIndex,
            mplsTunnelExtDestTnlLspIndex  MplsTunnelInstanceIndex
      }

      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
               same tunnel or in the different tunnels.

               This object holds the same value as that of the
               mplsTunnelIndex of mplsTunnelEntry if the forward and
               reverse LSPs are in the same tunnel. Otherwise,
               this object holds the value of the other direction
               associated LSP's mplsTunnelIndex from a different tunnel.

               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,
               if both the forward and reverse LSPs are present in
               the same tunnel, the opposite direction LSP's Ingress and
               Egress Identifier will be same for both the LSPs,
```

```
              else the Ingress and Egress Identifiers should be
              swapped in order to index the other direction tunnel.

              The value of zero for this object is invalid."
       ::= { mplsTunnelExtEntry 1 }

     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
           same tunnel or in the different tunnels.

           This object should contain different value if both the
           forward and reverse LSPs present in the same tunnel.

           This object can contain same value or different values if
           the forward and reverse LSPs present in the different
           tunnels.

           The value of zero for this object is valid for the
           configured tunnel."
       ::= { mplsTunnelExtEntry 2 }

    -- End of MPLS Tunnel table extension

   -- Notifications.
   -- Notifications objects need to be added here.
   -- End of notifications.

   -- Module compliance.

   mplsTeExtGroups
      OBJECT IDENTIFIER ::= { mplsTeExtConformance 1 }

   mplsTeExtCompliances
      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 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 2 }

       -- Units of conformance.


       mplsTunnelExtGroup OBJECT-GROUP
          OBJECTS {
             mplsTunnelExtDestTnlIndex,
             mplsTunnelExtDestTnlLspIndex
        }
        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 { mplsNodeConfigGlobalId,
                  mplsNodeConfigNodeId,
                  mplsNodeConfigRowStatus,
                  mplsNodeConfigStorageType,
                  mplsNodeIpMapLocalId
        }
        STATUS  current
        DESCRIPTION
             "Object(s) needed to implement IP compatible tunnels."
        ::= { mplsTeExtGroups 2 }

     mplsTunnelExtIccOperatorGroup  OBJECT-GROUP
        OBJECTS { mplsNodeConfigIccId,
                  mplsNodeConfigRowStatus,
                  mplsNodeConfigStorageType,
                  mplsNodeIccMapLocalId
        }
```

```
        STATUS  current
        DESCRIPTION
             "Object(s) needed to implement ICC based tunnels."
        ::= { mplsTeExtGroups 3 }

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

## 14. Security Consideration

There is a number of management objects defined in this MIB module that has a MAX-ACCESS clause of read-write.. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

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:

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

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

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

## 15. IANA Considerations

To be added in a later version of 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.

### 16.2 Informative References

[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.

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

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

[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.

[TPIDS] M. Bocci, et al, "MPLS-TP Identifiers", draft-ietf-mpls-tp-identifiers-03, October 25, 2010

## 17. Acknowledgments

To be added in a later version of this document.

## 18. Authors' Addresses

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

Thomas D. Nadeau
CA Technologies
273 Corporate Drive, Portsmouth, NH, USA

Email: thomas.nadeau@ca.com

Venkatesan Mahalingam
Aricent
India
Email: venkat.mahalingams@gmail.com

Kannan KV Sampath
Aricent
India
Email: Kannan.Sampath@aricent.com