Internet Working Group
Internet Draft
Proposed Status: Standards Track
Expires: December 2005

Thomas D. Nadeau, Ed.
Cisco Systems, Inc.

Adrian Farrel, Ed.
Old Dog Consulting

June 2005

Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base

draft-ietf-ccamp-gmpls-te-mib-09.txt

Status of this Memo

Abstract

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

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 for modeling Generalized Multiprotocol Label Switching (GMPLS) [RFC3945] based traffic engineering. The tables and objects defined in this document extend those defined in the equivalent document for MPLS traffic engineering [RFC3812], and management of GMPLS traffic engineering is built on management of MPLS traffic engineering.

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

Comments should be made direct to the CCAMP mailing list at ccamp@ops.ietf.org.

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

1.1. Migration Strategy

This MIB module extends the traffic engineering MIB module defined for use with MPLS [RFC3812]. It provides additions for support of GMPLS tunnels.

The companion document for modeling and managing GMPLS based LSRs [GMPLSLSRMIB] extends MPLS LSR MIB [RFC3813] with the same intentions.

Textual conventions and OBJECT-IDENTIFIERS are defined in [RFC3811] and [GMPLSTCMIB].

2. Terminology

This document uses terminology from the MPLS architecture document [RFC3031], from the GMPLS architecture document [RFC3945], and from the MPLS Traffic Engineering MIB [RFC3812]. Some frequently used terms are described next.

An explicitly routed LSP (ERLSP) is referred to as a GMPLS tunnel. It consists of in-segment(s) and/or out-segment(s) at the egress/ingress LSRs, each segment being associated with one GMPLS enabled interface. These are also referred to as tunnel segments.

Additionally, at an intermediate LSR, we model a connection as consisting of one or more in-segments and/or one or more out-segments. The binding or interconnection between in-segments and out-segments in performed using a cross-connect.

These segment and cross-connect objects are defined in the MPLS Label Switch Router MIB [RFC3813], but see also the GMPLS Label Switch Router MIB [GMPLSLSRMIB] for the GMPLS-specific extensions to these objects.

3. The SNMP Management Framework

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

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the

Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. Outline

Support for GMPLS traffic-engineered tunnels requires the following configuration.

- Setting up tunnels with appropriate MPLS configuration parameters using [RFC3812].
- Extending the tunnels with GMPLS configuration parameters.
- Configuring tunnel loose and strict source routed hops.

These actions may need to be accompanied with corresponding actions using [RFC3813] and [GMPLSLSRMIB] to establish and configure tunnel segments, if this is done manually. Also, the in-segment and out-segment performance tables, mplsInSegmentPerfTable and mplsOutSegmentPerfTable [RFC3813], should be used to determine performance of the tunnels and tunnel segments although it should be noted that those tables may not be appropriate for measuring performance on some types of GMPLS links.

4.1. Summary of GMPLS Traffic Engineering MIB Module

The MIB objects for performing the actions listed above that cannot be performed solely using the MIB objects defined in [RFC3812] consist of the following tables.

- Tunnel Table (gmplsTunnelTable) for providing GMPLS-specific tunnel configuration parameters.
- Tunnel specified, actual, and computed hop tables (gmplsTunnelHopTable, gmplsTunnelARHopTable, and gmplsTunnelCHopTable) for providing additional configuration of strict and loose source routed tunnel hops.
- Performance and error reporting tables (gmplsTunnelReversePerfTable and gmplsTunnelErrorTable).

These tables are described in the subsequent sections.

Additionally, this MIB module contains a new Notification.

- The GMPLS Tunnel Down Notification (gmplsTunnelDown) is intended to be used in place of the mplsTunnelDown Notification defined in [RFC3812]. As well as indicating that a tunnel has transitioned to operational down state, this new Notificaiton indicates the cause of the failure.

5. Brief Description of GMPLS TE MIB Objects

The objects described in this section support the functionality described in [RFC3473] and [RFC3472] for GMPLS tunnels. The tables support both manually configured and signaled tunnels.

5.1. gmplsTunnelTable

The gmplsTunnelTable extends the MPLS traffic engineering MIB module to allow GMPLS tunnels to be created between an LSR and a remote endpoint, and existing GMPLS tunnels to be reconfigured or removed.

Note that we only support point-to-point tunnel segments, although multi-point-to-point and point-to-multi-point connections are supported by an LSR acting as a cross-connect.

Each tunnel can thus have one out-segment originating at an LSR and/or one in-segment terminating at that LSR.

Three objects within this table utilize enumerations in order to map to enumerations that are used in GMPLS signaling. In order to protect this MIB module from changes (in particular, extensions) to the range of enumerations supported by the signaling protocols, these MIB objects use Textual Conventions defined by IANA. For further details, see the IANA Considerations section of this document.

5.2. gmplsTunnelHopTable

The gmplsTunnelHopTable is used to indicate additional parameters for the hops, strict or loose, of a GMPLS tunnel defined in gmplsTunnelTable, when it is established using signaling. Multiple tunnels may share hops by pointing to the same entry in this table.

5.3. gmplsTunnelARHopTable

The gmplsTunnelARHopTable is used to indicate the actual hops traversed by a tunnel as reported by the signaling protocol after the tunnel is setup. The support of this table is optional since not all GMPLS signaling protocols support this feature.

5.4. gmplsTunnelCHoptable

The gmplsTunnelCHopTable lists the actual hops computed by a constraint-based routing algorithm based on the gmplsTunnelHopTable. The support of this table is optional since not all implementations support computation of hop lists using a constraint-based routing protocol.

5.5. gmplsTunnelErrorTable

The gmplsTunnelErrorTable provides access to information about the last error that occurred on each tunnel known about by the MIB. It indicates the nature of the error, when and how it was reported and can give recovery advice through a display string.

5.6. gmplsTunnelReversePerfTable

gmplsTunnelReversePerfTable provides additional counters to measure the performance of bidirectional GMPLS tunnels in which packets are visible. It supplements the counters in mplsTunnelPerfTable and augments gmplsTunnelTable.

Note that not all counters may be appropriate or available for some types of tunnel.

6. Cross-referencing to the gmplsLabelTable

The gmplsLabelTable is found in a MIB module in [GMPLSLSRMIB] and provides a way to model labels in a GMPLS system where labels might not be simple 32 bit integers.

The hop tables in this document (gmplsHopTable, gmplsCHopTable and gmplsARHopTable) and the segment tables in the [RFC3813] (mplsInSegmentTable and mplsOutSegmentTable) contain objects with syntax MplsLabel.

MplsLabel (defined in [RFC3811]) is a 32-bit integer that is capable of representing any MPLS label and most GMPLS labels. However, some GMPLS labels are larger than 32 bits and may be of arbitrary length. Further, some labels that may be safely encoded in 32 bits are constructed from multiple sub-fields. Additionally, some GMPLS technologies support the concatenation of individual labels to represent a data flow carried as multiple sub-flows.

These GMPLS cases require that something other than a simple 32-bit integer is made available to represent the labels. This is achieved through the gmplsLabelTable contained in [GMPLSLSRMIB].

The tables in this document and [RFC3813] that include objects with syntax MplsLabel also include companion objects that are row pointers. If the row pointer is set to zeroDotZero (0.0) then object of syntax MplsLabel contains the label encoded as a 32-bit integer. But otherwise the row pointer indicates a row in another MIB table that includes the label. In these cases, the row pointer may indicate a row in the gmplsLabelTable.

This provides both a good way to support legacy systems that implement the previous version of this MIB module [RFC3812], and a significant simplification in GMPLS systems that are limited to a single, simple label type.

Note that gmplsLabelTable supports concatenated labels through the use of a label sub-index (gmplsLabelSubindex).

7. Example of GMPLS Tunnel Setup

This section contains an example of which MIB objects should be modified to create a GMPLS tunnel. This example shows a best effort, loosely routed, bidirectional traffic engineered tunnel, which spans two hops of a simple network, uses Generalized Label requests with Lambda encoding, has label recording and shared link layer protection. Note that these objects should be created on the "head-end" LSR.

```
First in the mplsTunnelTable:
{
  mplsTunnelIndex                = 1,
  mplsTunnelInstance             = 1,
  mplsTunnelIngressLSRId         = 123.123.125.1,
  mplsTunnelEgressLSRId          = 123.123.126.1,
  mplsTunnelName                 = "My first tunnel",
  mplsTunnelDescr                = "Here to there and back again",
  mplsTunnelIsIf                 = true (1),
  mplsTunnelXCPointer            = mplsXCIndex.3.0.0.12,
  mplsTunnelSignallingProto      = none (1),
  mplsTunnelSetupPrio            = 0,
  mplsTunnelHoldingPrio          = 0,
  mplsTunnelSessionAttributes    = recordRoute (4),
  mplsTunnelOwner                = snmp (2),
  mplsTunnelLocalProtectInUse    = false (2),
  mplsTunnelResourcePointer      = mplsTunnelResourceIndex.6,
  mplsTunnelInstancePriority     = 1,
  mplsTunnelHopTableIndex        = 1,

  mplsTunnelPrimaryInstance      = 0,
  mplsTunnelIncludeAnyAffinity   = 0,
  mplsTunnelIncludeAllAffinity   = 0,
  mplsTunnelExcludeAnyAffinity   = 0,
  mplsTunnelPathInUse            = 1,
  mplsTunnelRole                 = head(1),
  mplsTunnelRowStatus            = createAndWait (5),
}

In gmplsTunnelTable(1,1,123.123.125.1,123.123.126.1):
{
  gmplsTunnelUnnumIf             = true (1),
  gmplsTunnelAttributes          = labelRecordingRequired (1),
  gmplsTunnelLSPEncoding         = tunnelLspLambda,
  gmplsTunnelSwitchingType       = lsc,
  gmplsTunnelLinkProtection      = shared (2),
  gmplsTunnelGPid                = lambda,
  gmplsTunnelSecondary           = false (2),
  gmplsTunnelDirection           = bidirectional (1)
  gmplsTunnelPathComp            = explicit(2),
  gmplsTunnelUpNotRecip          = 0x7B7B7D01,
  gmplsTunnelDownNotRecip        = 0x00000000,
  gmplsTunnelAdminStatusFlags    = 0,
  gmplsTunnelExtraParamsPtr      = 0.0
}
```

Entries in the mplsTunnelResourceTable, mplsTunnelHopTable and gmplsTunnelHopTable are created and activated at this time.

```
   In mplsTunnelResourceTable:
   {
     mplsTunnelResourceIndex        = 6,
     mplsTunnelResourceMaxRate      = 0,
     mplsTunnelResourceMeanRate     = 0,
     mplsTunnelResourceMaxBurstSize = 0,
     mplsTunnelResourceRowStatus    = createAndGo (4)
   }
```

The next two instances of mplsTunnelHopEntry are used to denote the hops this tunnel will take across the network.

The following denotes the beginning of the network, or the first hop. We have used the fictitious LSR identified by "123.123.125.1" as our example head-end router.

```
   In mplsTunnelHopTable:
   {
     mplsTunnelHopListIndex         = 1,
     mplsTunnelPathOptionIndex      = 1,
     mplsTunnelHopIndex             = 1,
     mplsTunnelHopAddrType          = ipV4 (1),
     mplsTunnelHopIpv4Addr          = 123.123.125.1,
     mplsTunnelHopIpv4PrefixLen     = 9,
     mplsTunnelHopType              = strict (1),
     mplsTunnelHopRowStatus         = createAndWait (5),
   }
```

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

```
   In mplsTunnelHopTable:
   {
     mplsTunnelHopListIndex         = 1,
     mplsTunnelPathOptionIndex      = 1,
     mplsTunnelHopIndex             = 2,
     mplsTunnelHopAddrType          = ipV4 (1),
     mplsTunnelHopIpv4Addr          = 123.123.126.1,
     mplsTunnelHopIpv4PrefixLen     = 9,
     mplsTunnelHopType              = loose (2),
     mplsTunnelHopRowStatus         = createAndGo (4)
   }
```

Now an associated entry in the gmplsTunnelHopTable is created to provide additional GMPLS hop configuration indicating that the first hop is an unnumbered link using explicit forward and reverse labels. An entry in the gmplsLabelTable is created first to include the explicit label.

```
In gmplsLabelTable:
{
  gmplsLabelInterface            = 2,
  gmplsLabelIndex                = 1,
  gmplsLabelSubindex             = 0,
  gmplsLabelType                 = gmplsFreeformGeneralizedLabel(3),
  gmplsLabelFreeform             = 0xFEDCBA9876543210
  gmplsLabelRowStatus            = createAndGo(4)
}
```

```
In gmplsTunnelHopTable(1,1,1):
{
  gmplsTunnelHopLabelStatuses    = forwardPresent(0)
                                             +reversePresent(1),
  gmplsTunnelHopExpLabelPtr      = gmplsLabelTable (2, 1, 0)
  gmplsTunnelHopExpRvrsLabelPtr  = gmplsLabelTable (2, 1, 0)
}
```

The first hop is now activated:

```
In mplsTunnelHopTable(1,1,1):
{
  mplsTunnelHopRowStatus         = active (1)
}
```

No gmplsTunnelHopEntry is created for the second hop as it contains no special GMPLS features.

Finally the mplsTunnelEntry is activated:

```
In mplsTunnelTable(1,1,123.123.125.1,123.123.126.1)
{
  mplsTunnelRowStatus            = active(1)
}
```

8. GMPLS Traffic Engineering MIB Module

```
   GMPLS-TE-STD-MIB DEFINITIONS ::= BEGIN

   IMPORTS
     MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
     Unsigned32, Counter32,
     Counter64, IpAddress, zeroDotZero
       FROM SNMPv2-SMI                              -- RFC2578
     MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
       FROM SNMPv2-CONF                             -- RFC2580
     TruthValue, TimeStamp, DisplayString, RowPointer
       FROM SNMPv2-TC                               -- RFC2579
     InetAddress, InetAddressType
       FROM INET-ADDRESS-MIB                        -- RFC4001
     mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
     mplsTunnelEgressLSRId, mplsTunnelHopListIndex,
     mplsTunnelHopPathOptionIndex, mplsTunnelHopIndex,
     mplsTunnelARHopListIndex, mplsTunnelARHopIndex,
     mplsTunnelCHopListIndex, mplsTunnelCHopIndex,
     mplsTunnelEntry,
     mplsTunnelAdminStatus, mplsTunnelOperStatus
       FROM MPLS-TE-STD-MIB                         -- RFC3812
     mplsStdMIB
       FROM MPLS-TC-STD-MIB                         -- RFC3811
     IANAGmplsLSPEncoding, IANAGmplsSwitchingType, IANAGmplsGPid,
     IANAGmplsAdminStatusFlags
       FROM IANA-GMPLS-MIB
   ;

   gmplsTeStdMIB MODULE-IDENTITY
         LAST-UPDATED
           "200505200001Z" -- 20 May 2005 00:00:01 GMT
         ORGANIZATION
           "Common Control And Measurement Plane (CCAMP) Working Group"
         CONTACT-INFO
           "       Thomas D. Nadeau
                   Cisco Systems, Inc.
            Email: tnadeau@cisco.com

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

            Comments about this document should be emailed direct to the
            CCAMP working group mailing list at ccamp@ops.ietf.org"
```

```
         DESCRIPTION
           "Copyright (C) The Internet Society (2005). The
            initial version of this MIB module was published
            in RFC xxxx. For full legal notices see the RFC
            itself or see: http://www.ietf.org/copyrights/ianamib.html

            This MIB module contains managed object definitions
            for GMPLS Traffic Engineering (TE) as defined in:
            1. Generalized Multi-Protocol Label Switching (GMPLS)
               Signaling Functional Description, Berger, L. (Editor),
               RFC 3471, January 2003.
            2. Generalized MPLS Signaling - RSVP-TE Extensions, Berger,
               L. (Editor), RFC 3473, January 2003."

   -- Revision history.
         REVISION
           "200505200001Z" -- 20 May 2005 00:00:01 GMT
         DESCRIPTION
           -- RFC Editor: Please see the IANA Considerations Section.
           -- RFC-editor please fill in XXXX
           "Initial version issued as part of RFC XXXX."
   ::= { mplsStdMIB XXX }

   -- Top level components of this MIB.

   -- Notifications

   gmplsTeNotifications OBJECT IDENTIFIER ::= { gmplsTeStdMIB 0 }

   -- tables, scalars

   gmplsTeScalars OBJECT IDENTIFIER ::= { gmplsTeStdMIB 1 }
   gmplsTeObjects OBJECT IDENTIFIER ::= { gmplsTeStdMIB 2 }

   -- conformance

   gmplsTeConformance OBJECT IDENTIFIER ::= { gmplsTeStdMIB 3 }

   -- GMPLS Tunnel scalars.

   gmplsTunnelsConfigured OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "The number of GMPLS tunnels configured on this device. A GMPLS
        tunnel is considered configured if an entry for the tunnel
        exists in the gmplsTunnelTable and the associated
        mplsTunnelRowStatus is active(1)."
   ::= { gmplsTeScalars 1 }
```

```
   gmplsTunnelsActive OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "The number of GMPLS tunnels active on this device. A GMPLS
        tunnel is considered active if there is an entry in the
        gmplsTunnelTable and the associated mplsTunnelOperStatus for the
        tunnel is up(1)."
   ::= { gmplsTeScalars 2 }

   -- End of GMPLS Tunnel scalars.

   -- GMPLS tunnel table.

   gmplsTunnelTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "The gmplsTunnelTable 'extends' the mplsTunnelTable. It allows
        GMPLS tunnels to be created between an LSR and a remote
        endpoint, and existing tunnels to be reconfigured or removed.

        Note that only point-to-point tunnel segments are supported,
        although multi-point-to-point and point-to-multi-point
        connections are supported by an LSR acting as a cross-connect.
        Each tunnel can thus have one out-segment originating at this
        LSR and/or one in-segment terminating at this LSR.

        The row status of an entry in this table is controlled by
        mplsTunnelRowStatus in the corresponding entry in
        mplsTunnelTable. That is, it is not permitted to create a row in
        this table, nor to modify an existing row, when the
        corresponding mplsTunnelRowStatus has value active(1).

        The exception to this rule is the gmplsTunnelAdminStatusFlags
        object, which can be modified whilst the tunnel is active."
   ::= { gmplsTeObjects 1 }
```

```
   gmplsTunnelEntry OBJECT-TYPE
     SYNTAX  GmplsTunnelEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "An entry in this table in association with the corresponding
        entry in the mplsTunnelTable represents a GMPLS tunnel.

        An entry can be created by a network administrator or by an SNMP
        agent as instructed by a signaling protocol."
     INDEX {
       mplsTunnelIndex,
       mplsTunnelInstance,
       mplsTunnelIngressLSRId,
       mplsTunnelEgressLSRId
     }
   ::= { gmplsTunnelTable 1 }

   GmplsTunnelEntry ::= SEQUENCE {
     gmplsTunnelUnnumIf            TruthValue,
     gmplsTunnelAttributes         BITS,
     gmplsTunnelLSPEncoding        IANAGmplsLSPEncoding,
     gmplsTunnelSwitchingType      IANAGmplsSwitchingType,
     gmplsTunnelLinkProtection     BITS,
     gmplsTunnelGPid               IANAGmplsGPid,
     gmplsTunnelSecondary          TruthValue,
     gmplsTunnelDirection          INTEGER,
     gmplsTunnelPathComp           INTEGER,
     gmplsTunnelUpNotRecip         IpAddress,
     gmplsTunnelDownNotRecip       IpAddress,
     gmplsTunnelAdminStatusFlags   IANAGmplsAdminStatusFlags,
     gmplsTunnelExtraParamsPtr     RowPointer
   }
```

```
   gmplsTunnelUnnumIf OBJECT-TYPE
     SYNTAX  TruthValue
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Denotes whether or not this tunnel corresponds to an unnumbered
        interface represented in the interfaces group table.

        This object is only used if mplsTunnelIsIf is set to 'true'.

        If both this object and the mplsTunnelIsIf object are set to
        'true', the originating LSR adds an LSP_TUNNEL_INTERFACE_ID
        object to the outgoing Path message.

        This object contains information that is only used by the
        terminating LSR."
     REFERENCE
       "Signalling Unnumbered Links in RSVP-TE, Kompella, K.
        and Rekhter, Y., RFC 3477, January 2003."
     DEFVAL  { false }
   ::= { gmplsTunnelEntry 1 }

   gmplsTunnelAttributes OBJECT-TYPE
     SYNTAX BITS {
       labelRecordingDesired (0)
     }
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "This bitmask indicates optional parameters for this tunnel.
        These bits should be taken in addition to those defined in
        mplsTunnelSessionAttributes in order to determine the full set
        of options to be signaled (for example SESSION_ATTRIBUTES flags
        in RSVP-TE). The following describes these bitfields:

        labelRecordingDesired
          This flag indicates that label information should be included
          when doing a route record. This bit is not valid unless the
          recordRoute bit is set."
     REFERENCE
       "RSVP-TE: Extensions to RSVP for LSP Tunnels, Awduche et al.,
        RFC 3209, December 2001."
     DEFVAL  { { } }
   ::= { gmplsTunnelEntry 2 }
```

```
   gmplsTunnelLSPEncoding OBJECT-TYPE
     SYNTAX  IANAGmplsLSPEncoding
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "This object indicates the encoding of the LSP being requested.

        A value of 'tunnelLspNotGmpls' indicates that GMPLS signaling is
        not in use. Some objects in this MIB module may be of use for
        MPLS signaling extensions that do not use GMPLS signaling. By
        setting this object to 'tunnelLspNotGmpls', an application may
        indicate that only those objects meaningful in MPLS should be
        examined.

        The values to use are defined in the textual convention
        IANAGmplsLSPEncoding found in the IANA-GMPLS-MIB MIB module."
     DEFVAL  { tunnelLspNotGmpls }
   ::= { gmplsTunnelEntry 3 }

   gmplsTunnelSwitchingType OBJECT-TYPE
     SYNTAX  IANAGmplsSwitchingType
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Indicates the type of switching that should be performed on
        a particular link. This field is needed for links that
        advertise more than one type of switching capability.

        The values to use are defined in the textual convention
        IANAGmplsSwitchingType found in the IANA-GMPLS-MIB MIB module.

        This object is only meaningful if gmplsTunnelLSPEncoding is not
        set to 'tunnelLspNotGmpls'."
     DEFVAL  { unknown }
   ::= { gmplsTunnelEntry 4 }
```

```
   gmplsTunnelLinkProtection OBJECT-TYPE
     SYNTAX  BITS {
       extraTraffic(0),
       unprotected(1),
       shared (2),
       dedicatedOneToOne (3),
       dedicatedOnePlusOne(4),
       enhanced(5)
     }
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "This bitmask indicates the level of link protection required. A
        value of zero (no bits set) indicates that any protection may be
        used. The following describes these bitfields:

        extraTraffic
          Indicates that the LSP should use links that are protecting
          other (primary) traffic. Such LSPs may be preempted when the
          links carrying the (primary) traffic being protected fail.

        unprotected
          Indicates that the LSP should not use any link layer
          protection.

        shared
          Indicates that a shared link layer protection scheme, such as
          1:N protection, should be used to support the LSP.

        dedicatedOneToOne
          Indicates that a dedicated link layer protection scheme,
          i.e., 1:1 protection, should be used to support the LSP.

        dedicatedOnePlusOne
          Indicates that a dedicated link layer protection scheme,
          i.e., 1+1 protection, should be used to support the LSP.

        enhanced
          Indicates that a protection scheme that is more reliable than
          Dedicated 1+1 should be used, e.g., 4 fiber BLSR/MS-SPRING.

        This object is only meaningful if gmplsTunnelLSPEncoding is
        not set to 'tunnelLspNotGmpls'."
     REFERENCE
        "Berger, L., et al., Generalized Multi-Protocol
         Label Switching (GMPLS) Signaling Functional
         Description, RFC 3471, January 2003."
     DEFVAL  { { } }
   ::= { gmplsTunnelEntry 5 }
```

```
   gmplsTunnelGPid OBJECT-TYPE
     SYNTAX  IANAGmplsGPid
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "This object indicates the payload carried by the LSP. It is only
        required when GMPLS will be used for this LSP.

        The values to use are defined in the textual convention
        IANAGmplsGPid found in the IANA-GMPLS-MIB MIB module.

        This object is only meaningful if gmplsTunnelLSPEncoding is not
        set to 'tunnelLspNotGmpls'."
     DEFVAL  { unknown }
   ::= { gmplsTunnelEntry 6 }

   gmplsTunnelSecondary OBJECT-TYPE
     SYNTAX  TruthValue
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Indicates that the requested LSP is a secondary LSP.

        This object is only meaningful if gmplsTunnelLSPEncoding is not
        set to 'tunnelLspNotGmpls'."
     REFERENCE
       "Berger, L., et al., Generalized Multi-Protocol
        Label Switching (GMPLS) Signaling Functional
        Description, RFC 3471, January 2003."
     DEFVAL  { false }
   ::= { gmplsTunnelEntry 7 }

   gmplsTunnelDirection OBJECT-TYPE
     SYNTAX  INTEGER {
       forward (0),
       bidirectional (1)
     }
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Whether this tunnel carries forward data only (is
        unidirectional) or is bidirectional.

        Values of this object other than 'forward' are meaningful
        only if gmplsTunnelLSPEncoding is not set to
        'tunnelLspNotGmpls'."
     DEFVAL { forward }
   ::= { gmplsTunnelEntry 8 }
```

```
   gmplsTunnelPathComp OBJECT-TYPE
     SYNTAX  INTEGER {
       dynamicFull(1),   -- CSPF fully computed
       explicit(2),      -- fully specified path
       dynamicPartial(3) -- CSPF partially computed
     }
     MAX-ACCESS read-create
     STATUS current
     DESCRIPTION
       "This value instructs the source node on how to perform path
        computation on the explicit route specified by the associated
        entries in the gmplsTunnelHopTable.

        dynamicFull
          The user specifies at least the source and
          destination of the path and expects that the CSPF
          will calculate the remainder of the path.

        explicit
          The user specifies the entire path for the tunnel to
          take. This path may contain strict or loose hops.
          Evaluation of the explicit route will be performed
          hop by hop through the network.

        dynamicPartial
          The user specifies at least the source and
          destination of the path and expects that the CSPF
          will calculate the remainder of the path. The path
          computed by CSPF is allowed to be only partially
          computed allowing the remainder of the path to be
          filled in across the network.

        This object deprecates mplsTunnelHopEntryPathComp."
     DEFVAL { dynamicFull }
   ::= { gmplsTunnelEntry 9 }

   gmplsTunnelUpNotRecip OBJECT-TYPE
     SYNTAX  IpAddress
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Indicates the address of the upstream recipient for Notify
        messages relating to this tunnel. This object is only valid when
        signaling a tunnel using RSVP. It is also not valid at the tail
        end of the tunnel. If set to 0, no Notify Request object will be
        included in outgoing Path messages."
     REFERENCE
       "Generalized MPLS Signaling - RSVP-TE Extensions, Berger,
        L. (Editor), RFC 3473, January 2003."
```

```
      DEFVAL { '00000000'H } -- 0.0.0.0
    ::= { gmplsTunnelEntry 10 }
```

```
   gmplsTunnelDownNotRecip OBJECT-TYPE
     SYNTAX  IpAddress
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "Indicates the address of the downstream recipient for Notify
        messages relating to this tunnel.

        This object is only valid when signaling a tunnel using RSVP. It
        is also not valid at the head end of the tunnel.

        If set to 0, no Notify Request object will be included in
        outgoing Resv messages."
     REFERENCE
       "Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
        (Editor), RFC 3473, January 2003."
     DEFVAL { '00000000'H } -- 0.0.0.0
   ::= { gmplsTunnelEntry 11 }

   gmplsTunnelAdminStatusFlags OBJECT-TYPE
      SYNTAX   IANAGmplsAdminStatusFlags
      MAX-ACCESS   read-create
      STATUS       current
      DESCRIPTION
        "Determines the setting of the Admin Status flags in the
         Admin Status object or TLV, as described in RFC 3471. Setting
         this field to a non-zero value will result in the inclusion of
         the admin status object on signaling messages.

         The values to use are defined in the textual convention
         IANAGmplsAdminStatusFlags found in the IANA-GMPLS-MIB MIB
         module.

         This value of this object can be modified when the
         corresponding mplsTunnelRowStatus and mplsTunnelAdminStatus
         is active(1). By doing so, a new signaling message will be
         triggered including the requested Admin Status object or
         TLV."
     REFERENCE
       "Berger, L., et al., Generalized Multi-Protocol Label Switching
        (GMPLS) Signaling Functional Description, RFC 3471,
        January 2003."
     DEFVAL  { { } }
     ::= { gmplsTunnelEntry 12 }
```

```
   gmplsTunnelExtraParamsPtr  OBJECT-TYPE
     SYNTAX       RowPointer
     MAX-ACCESS   read-create
     STATUS       current
     DESCRIPTION
       "Some Tunnels will run over transports that can usefully support
        technology-specific additional parameters (for example, SONET
        resource usage). Such parameters can be supplied in an external
        table and referenced from here.

        A value of zeroDotzero in this attribute indicates that there
        is no such additional information."
     DEFVAL  { zeroDotZero }
     ::= { gmplsTunnelEntry 13 }

   -- End of gmplsTunnelTable

   -- Begin gmplsTunnelHopTable

   gmplsTunnelHopTable  OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelHopEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "The gmplsTunnelHopTable 'extends' the mplsTunnelHopTable. It is
        used to indicate the explicit labels to be used in an explicit
        path for a GMPLS tunnel defined in mplsTunnelTable and
        gmplsTunnelTable, when it is established using signaling. It
        does not insert new hops, but does define new values for hops
        defined in mplsTunnelHopTable.

        Each row in this table is indexed by the same indexes as
        mplsTunnelHopTable. It is acceptable for some rows in
        mplsTunnelHopTable to have corresponding entries in this table
        and some to have no corresponding entry in this table.

        The storage type for an entry in this table is inherited from
        mplsTunnelHopStorageType in the corresponding entry in
        mplsTunnelHopTable.

        The row status of an entry in this table is controlled by
        mplsTunnelHopRowStatus in the corresponding entry in
        mplsTunnelHopTable. That is, it is not permitted to create a row
        in this table, nor to modify an existing row, when the
        corresponding mplsTunnelHopRowStatus has value active(1)."
     REFERENCE
       "Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
        (Editor), RFC 3473, January 2003."
   ::= { gmplsTeObjects 2 }
```

```
   gmplsTunnelHopEntry  OBJECT-TYPE
     SYNTAX  GmplsTunnelHopEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "An entry in this table represents additions to a tunnel hop
        defined in mplsTunnelHopEntry. At an ingress to a tunnel an
        entry in this table is created by a network administrator for an
        ERLSP to be set up by a signaling protocol. At transit and
        egress nodes an entry in this table may be used to represent the
        explicit path instructions received using the signaling
        protocol."
     INDEX {
       mplsTunnelHopListIndex,
       mplsTunnelHopPathOptionIndex,
       mplsTunnelHopIndex
     }
   ::= { gmplsTunnelHopTable 1 }

   GmplsTunnelHopEntry ::= SEQUENCE {
     gmplsTunnelHopLabelStatuses     BITS,
     gmplsTunnelHopExpLabel          Unsigned32,
     gmplsTunnelHopExpLabelPtr       RowPointer,
     gmplsTunnelHopExpRvrsLabel      Unsigned32,
     gmplsTunnelHopExpRvrsLabelPtr   RowPointer
   }

   gmplsTunnelHopLabelStatuses OBJECT-TYPE
     SYNTAX  BITS {
       forwardPresent (0),
       reversePresent (1)
     }
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "This bitmask indicates the presence of labels indicated by the
        gmplsTunnelHopExpLabel or gmplsTunnelHopExpLabelPtr, and
        gmplsTunnelHopExpRvrsLabel or gmplsTunnelHopExpRvrsLabel
        objects.

        For the Present bits, a set bit indicates that a label is
        present for this hop in the route. This allows zero to be a
        valid label value."
     DEFVAL  { { } }
   ::= { gmplsTunnelHopEntry 1 }
```

```
   gmplsTunnelHopExpLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "If gmplsTunnelHopLabelStatuses object indicates that a forward
        label is present and gmplsTunnelHopExpLabelPtr contains the
        value zeroDotZero, then the label to use on this hop is found in
        object encoded within a 32-bit integer."
   ::= { gmplsTunnelHopEntry 2 }

   gmplsTunnelHopExpLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelHopLabelStatuses object indicates that a
        forward label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label to use on this hop in the forward direction.
        If the gmplsTunnelHopLabelStatuses object indicates that a
        forward label is present and this object contains the value
        zeroDotZero, then the label to use on this hop is found in the
        gmplsTunnelHopExpLabel object."
     DEFVAL  { zeroDotZero }
   ::= { gmplsTunnelHopEntry 3 }

   gmplsTunnelHopExpRvrsLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelHopLabelStatuses object indicates that a
        reverse label is present and gmplsTunnelHopExpRvrsLabelPtr
        contains the value zeroDotZero, then the label to use on this
        this hop is found in this object encoded as a 32-bit integer."
   ::= { gmplsTunnelHopEntry 4 }
```

```
   gmplsTunnelHopExpRvrsLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-create
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelHopLabelStatuses object indicates that a
        reverse label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label to use on this hop in the reverse direction.

        If the gmplsTunnelHopLabelStatuses object indicates that a
        reverse label is present and this object contains the value
        zeroDotZero, then the label to use on this hop is found in the
        gmplsTunnelHopExpRvrsLabel object."
     DEFVAL  { zeroDotZero }
   ::= { gmplsTunnelHopEntry 5 }

   -- End of gmplsTunnelHopTable

   -- Tunnel Actual Route Hop table.

   gmplsTunnelARHopTable  OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelARHopEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "The gmplsTunnelARHopTable 'extends' the mplsTunnelARHopTable. It
        is used to indicate the labels currently in use for a GMPLS
        tunnel defined in mplsTunnelTable and gmplsTunnelTable, as
        reported by the signaling protocol. It does not insert new hops,
        but does define new values for hops defined in
        mplsTunnelARHopTable.

        Each row in this table is indexed by the same indexes as
        mplsTunnelARHopTable. It is acceptable for some rows in
        mplsTunnelARHopTable to have corresponding entries in this table
        and some to have no corresponding entry in this table.

        Note that since the information necessary to build entries
        within this table is not provided by some signaling protocols
        and might not be returned in all cases of other signaling
        protocols, implementation of this table and mplsTunnelARHopTable
        is optional. Furthermore, since the information in this table is
        actually provided by the signaling protocol after the path has
        been set-up, the entries in this table are provided only for
        observation, and hence, all variables in this table are
        accessible exclusively as read-only."
```

```
      REFERENCE
        "1. Extensions to RSVP for LSP Tunnels, Awduche et al, RFC 3209,
            December 2001
         2. Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
            (Editor), RFC 3473, January 2003."
    ::= { gmplsTeObjects 3 }

    gmplsTunnelARHopEntry  OBJECT-TYPE
      SYNTAX  GmplsTunnelARHopEntry
      MAX-ACCESS not-accessible
      STATUS  current
      DESCRIPTION
        "An entry in this table represents additions to a tunnel hop
         visible in mplsTunnelARHopEntry. An entry is created by the
         signaling protocol for a signaled ERLSP set up by the signaling
         protocol.

         At any node on the LSP (ingress, transit or egress) Thus at this
         table and mplsTunnelARHopTable (if the tables are supported and
         if the signaling protocol is recording actual route information)
         contains the actual route of the whole tunnel. If the signaling
         protocol is not recording the actual route, this table MAY
         report the information from the gmplsTunnelHopTable or the
         gmplsTunnelCHopTable.

         Note that the recording of actual labels is distinct from the
         recording of the actual route in some signaling protocols. This
         feature is enabled using the gmplsTunnelAttributes object."
      INDEX {
        mplsTunnelARHopListIndex,
        mplsTunnelARHopIndex
      }
    ::= { gmplsTunnelARHopTable 1 }

    GmplsTunnelARHopEntry ::= SEQUENCE {
      gmplsTunnelARHopLabelStatuses     BITS,
      gmplsTunnelARHopExpLabel          Unsigned32,
      gmplsTunnelARHopExpLabelPtr       RowPointer,
      gmplsTunnelARHopExpRvrsLabel      Unsigned32,
      gmplsTunnelARHopExpRvrsLabelPtr   RowPointer,
      gmplsTunnelARHopProtection        BITS
    }
```

```
   gmplsTunnelARHopLabelStatuses OBJECT-TYPE
     SYNTAX  BITS {
       forwardPresent (0),
       reversePresent (1),
       forwardGlobal (2),
       reverseGlobal (3)
     }
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "This bitmask indicates the presence and status of labels
        indicated by the gmplsTunnelARHopExpLabel or
        gmplsTunnelARHopExpLabelPtr, and gmplsTunnelARHopExpRvrsLabel or
        gmplsTunnelARHopExpRvrsLabelPtr objects.

        For the Present bits, a set bit indicates that a label is
        present for this hop in the route. For the Global bits, a set
        bit indicates that the label comes from the Global Label Space.
        A clear bit indicates that this is a Per-Interface label. A
        Global bit only has meaning if the corresponding Present bit is
        set."
   ::= { gmplsTunnelARHopEntry 1 }

   gmplsTunnelARHopExpLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelARHopLabelStatuses object indicates that a
        forward label is present and gmplsTunnelARHopExpLabelPtr
        contains the value zeroDotZero, then the label in use on this
        hop is found in this object encoded within a 32-bit integer."
   ::= { gmplsTunnelARHopEntry 2 }

   gmplsTunnelARHopExpLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelARHopLabelStatuses object indicates that a
        forward label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label in use on this hop in the forward direction.
        If the gmplsTunnelARHopLabelStatuses object indicates that a
        forward label is present and this object contains the value
        zeroDotZero, then the label in use on this hop is found in the
        gmplsTunnelARHopExpLabel object."
   ::= { gmplsTunnelARHopEntry 3 }
```

```
   gmplsTunnelARHopExpRvrsLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelARHopLabelStatuses object indicates that a
        reverse label is present and gmplsTunnelARHopExpRvrsLabelPtr
        contains the value zeroDotZero, then the label in use on this
        hop is found in this object encoded as a 32-bit integer."
   ::= { gmplsTunnelARHopEntry 4 }

   gmplsTunnelARHopExpRvrsLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelARHopLabelStatuses object indicates that a
        reverse label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label in use on this hop in the reverse direction.
        If the gmplsTunnelARHopLabelStatuses object indicates that a
        reverse label is present and this object contains the value
        zeroDotZero, then the label in use on this hop is found in the
        gmplsTunnelARHopExpRvrsLabel object."
   ::= { gmplsTunnelARHopEntry 5 }

   gmplsTunnelARHopProtection  OBJECT-TYPE
     SYNTAX  BITS {
       localAvailable (0),
       localInUse (1)
     }
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
      "Availability and usage of protection on the reported link.

       localAvailable
          Indicates that the link downstream of this node is protected
          via a local repair mechanism.

       localInUse
          Indicates that a local repair mechanism is in use to maintain
          this tunnel (usually in the face of an outage of the link it
          was previously routed over)."
   ::= { gmplsTunnelARHopEntry 6 }

   -- End of mplsTunnelARHopTable
```

```
   -- Tunnel Computed Hop table.

   gmplsTunnelCHopTable  OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelCHopEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "The gmplsTunnelCHopTable 'extends' the mplsTunnelCHopTable. It
        is used to indicate additional information about the hops of a
        GMPLS tunnel defined in mplsTunnelTable and gmplsTunnelTable, as
        computed by a constraint-based routing protocol, based on the
        mplsTunnelHopTable and the gmplsTunnelHopTable.

        Each row in this table is indexed by the same indexes as
        mplsTunnelCHopTable. It is acceptable for some rows in
        mplsTunnelCHopTable to have corresponding entries in this table
        and some to have no corresponding entry in this table.

        Please note that since the information necessary to build
        entries within this table may not be supported by some LSRs,
        implementation of this table is optional.

        Furthermore, since the information in this table is actually
        provided by a path computation component after the path has been
        computed, the entries in this table are provided only for
        observation, and hence, all objects in this table are accessible
        exclusively as read-only."
     REFERENCE
       "Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
        (Editor), RFC 3473, January 2003."
   ::= { gmplsTeObjects 4 }

   gmplsTunnelCHopEntry  OBJECT-TYPE
     SYNTAX  GmplsTunnelCHopEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "An entry in this table represents additions to a computed tunnel
        hop visible in mplsTunnelCHopEntry. An entry is created by a
        path computation component based on the hops specified in the
        corresponding mplsTunnelHopTable and gmplsTunnelHopTable.

        At a transit LSR this table (if the table is supported) MAY
        contain the path computed by path computation engine on (or on
        behalf of) the transit LSR."
     INDEX {
       mplsTunnelCHopListIndex,
       mplsTunnelCHopIndex
     }
   ::= { gmplsTunnelCHopTable 1 }
   GmplsTunnelCHopEntry ::= SEQUENCE {
     gmplsTunnelCHopLabelStatuses     BITS,
     gmplsTunnelCHopExpLabel          Unsigned32,
     gmplsTunnelCHopExpLabelPtr       RowPointer,
     gmplsTunnelCHopExpRvrsLabel      Unsigned32,
     gmplsTunnelCHopExpRvrsLabelPtr   RowPointer
   }

   gmplsTunnelCHopLabelStatuses OBJECT-TYPE
     SYNTAX  BITS {
       forwardPresent (0),
       reversePresent (1)
     }
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "This bitmask indicates the presence of labels indicated by the
        gmplsTunnelCHopExpLabel or gmplsTunnelCHopExpLabelPtr and
        gmplsTunnelCHopExpRvrsLabel or gmplsTunnelCHopExpRvrsLabelPtr
        objects.
        A set bit indicates that a label is present for this hop in the
        route thus allowing zero to be a valid label value."
   ::= { gmplsTunnelCHopEntry 1 }

   gmplsTunnelCHopExpLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelCHopLabelStatuses object indicates that a
        forward label is present and gmplsTunnelCHopExpLabelPtr contains
        the value zeroDotZero, then the label to use on this hop is
        found in this object encoded within a 32-bit integer."
   ::= { gmplsTunnelCHopEntry 2 }

   gmplsTunnelCHopExpLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelCHopLabelStatuses object indicates that a
        forward label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label to use on this hop in the forward direction.
        If the gmplsTunnelCHopLabelStatuses object indicates that a
        forward label is present and this object contains the value
        zeroDotZero, then the label to use on this hop is found in the
        gmplsTunnelCHopExpLabel object."
   ::= { gmplsTunnelCHopEntry 3 }
```

```
   gmplsTunnelCHopExpRvrsLabel OBJECT-TYPE
     SYNTAX  Unsigned32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelCHopLabelStatuses object indicates that a
        reverse label is present and gmplsTunnelCHopExpRvrsLabelPtr
        contains the value zeroDotZero, then the label to use on this
        hop is found in this object encoded as a 32-bit integer."
   ::= { gmplsTunnelCHopEntry 4 }

   gmplsTunnelCHopExpRvrsLabelPtr OBJECT-TYPE
     SYNTAX  RowPointer
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "If the gmplsTunnelCHopLabelStatuses object indicates that a
        reverse label is present, this object contains a pointer to a
        row in another MIB table (such as the gmplsLabelTable) that
        contains the label to use on this hop in the reverse direction.

        If the gmplsTunnelCHopLabelStatuses object indicates that a
        reverse label is present and this object contains the value
        zeroDotZero, then the label to use on this hop is found in the
        gmplsTunnelCHopExpRvrsLabel object."
   ::= { gmplsTunnelCHopEntry 5 }

   -- End of gmplsTunnelCHopTable

   -- GMPLS Tunnel Reverse Direction Performance Table.

   gmplsTunnelReversePerfTable  OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelReversePerfEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "This table 'augments' the gmplsTunnelTable to provides
        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."
     REFERENCE
       "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), Srinivasan, C., Viswanathan,
        A., Nadeau, T., RFC 3812, June 2004."
   ::= { gmplsTeObjects 5 }
```

```
   gmplsTunnelReversePerfEntry OBJECT-TYPE
     SYNTAX  GmplsTunnelReversePerfEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "An entry in this table is created by the LSR for every
        bidirectional GMPLS tunnel where packets are visible to the
        LSR."
     AUGMENTS { gmplsTunnelEntry }
   ::= { gmplsTunnelReversePerfTable 1 }

   GmplsTunnelReversePerfEntry ::= SEQUENCE {
     gmplsTunnelReversePerfPackets     Counter32,
     gmplsTunnelReversePerfHCPackets   Counter64,
     gmplsTunnelReversePerfErrors      Counter32,
     gmplsTunnelReversePerfBytes       Counter32,
     gmplsTunnelReversePerfHCBytes     Counter64
   }

   gmplsTunnelReversePerfPackets OBJECT-TYPE
     SYNTAX  Counter32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "Number of packets forwarded on the tunnel in the reverse
        direction if it is bidirectional."
   ::= { gmplsTunnelReversePerfEntry 1 }

   gmplsTunnelReversePerfHCPackets OBJECT-TYPE
     SYNTAX  Counter64
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "High capacity counter for number of packets forwarded on the
        tunnel in the reverse direction if it is bidirectional."
   ::= { gmplsTunnelReversePerfEntry 2 }

   gmplsTunnelReversePerfErrors OBJECT-TYPE
     SYNTAX  Counter32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "Number of errored packets received on the tunnel in the reverse
        direction if it is bidirectional."
   ::= { gmplsTunnelReversePerfEntry 3 }
```

```
   gmplsTunnelReversePerfBytes OBJECT-TYPE
     SYNTAX  Counter32
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "Number of bytes forwarded on the tunnel in the reverse direction
        if it is bidirectional."
   ::= { gmplsTunnelReversePerfEntry 4 }

   gmplsTunnelReversePerfHCBytes OBJECT-TYPE
     SYNTAX  Counter64
     MAX-ACCESS read-only
     STATUS  current

     DESCRIPTION
       "High capacity counter for number of bytes forwarded on the
        tunnel in the reverse direction if it is bidirectional."
   ::= { gmplsTunnelReversePerfEntry 5 }

   -- End of gmplsTunnelReversePerfTable

   -- GMPLS Tunnel Error Table.

   gmplsTunnelErrorTable  OBJECT-TYPE
     SYNTAX  SEQUENCE OF GmplsTunnelErrorEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "This table 'augments' the mplsTunnelTable
        This table provides per-tunnel information about errors. Errors
        may be detected locally or reported through the signaling
        protocol. Error reporting is not exclusive to GMPLS and this
        table may be applied in MPLS systems."
     REFERENCE
       "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), Srinivasan, C., Viswanathan,
        A., Nadeau, T., RFC 3812, June 2004."
   ::= { gmplsTeObjects 6 }
```

```
   gmplsTunnelErrorEntry OBJECT-TYPE
     SYNTAX  GmplsTunnelErrorEntry
     MAX-ACCESS not-accessible
     STATUS  current
     DESCRIPTION
       "An entry in this table is created by the LSR for every tunnel
        where error information is visible to the LSR.
        Note that systems which read the objects in this table one at a
        time may experience a discontinuity as the result of a new error
        occurring in between object reads. Systems that are vulnerable
        to this should read gmplsTunnelErrorLastTime before and after
        reading the other objects."
     AUGMENTS { mplsTunnelEntry }
   ::= { gmplsTunnelErrorTable 1 }

   GmplsTunnelErrorEntry ::= SEQUENCE {
     gmplsTunnelErrorLastErrorType      INTEGER,
     gmplsTunnelErrorLastTime           TimeStamp,
     gmplsTunnelErrorReporterType       InetAddressType,
     gmplsTunnelErrorReporter           InetAddress,
     gmplsTunnelErrorCode               Unsigned32,
     gmplsTunnelErrorSubcode            Unsigned32,
     gmplsTunnelErrorTLVs               OCTET STRING,
     gmplsTunnelErrorHelpString         DisplayString
   }

   gmplsTunnelErrorLastErrorType OBJECT-TYPE
     SYNTAX  INTEGER {
       noError (0),
       unknown (1),
       protocol (2),
       pathComputation (3),
       localConfiguration (4),
       localResources (5),
       localOther (6)
     }
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "The nature of the last error. Provides interpretation context
        for gmplsTunnelErrorProtocolCode and
        gmplsTunnelErrorProtocolSubcode.

        A value of noError (0) shows that there is no error associated
        with this tunnel and means that the other objects in this table
        entry have no meaning.

        A value of unknown (1) shows that there is an error but that no
        additional information about the cause is known. The error may
```

have been received in a signaled message or generated locally.

```
        A value of protocol (2) or pathComputation (3) indicates that
        the cause of an error and identifies an error that has been
        received through signaling or will itself be signaled.

        A value of localConfiguration (4), localResources (5) or
        localOther (6) identifies an error which has been detected
        by the local node, but which will not be reported through
        signaling."
   ::= { gmplsTunnelErrorEntry 1 }

   gmplsTunnelErrorLastTime OBJECT-TYPE
     SYNTAX  TimeStamp
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "The time at which the last error occurred. This is presented as
        the value of SysUpTime when the error occurred or was reported
        to this node.

        If gmplsTunnelErrorLastErrorType has the value noError (0), then
        this object is ignored."
   ::= { gmplsTunnelErrorEntry 2 }

   gmplsTunnelErrorReporterType OBJECT-TYPE
      SYNTAX  InetAddressType
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
        "The address type of the error reported.

         This object is used to aid in interpretation of
         gmplsTunnelErrorReporter."
   ::= { gmplsTunnelErrorEntry 3 }

   gmplsTunnelErrorReporter OBJECT-TYPE
     SYNTAX  InetAddress
     MAX-ACCESS read-only
     STATUS  current
     DESCRIPTION
       "The address of the node reporting the last error, or the address
        of the resource (such as an interface) associated with the
        error.

        If gmplsTunnelErrorLastErrorType has the value noError (0), then
        this object is ignored.

        If gmplsTunnelErrorLastErrorType has the value unknown (1),
        localConfiguration (4), localResources (5), or localOther (6)
        this object MAY contain a zero value.
```

```
           This object should be interpreted in the context of the value of
           the object gmplsTunnelErrorReporterType."
        REFERENCE
          "RFC4001, Textual Conventions for Internet Network Addresses,
           Section 4. Usage Hints."
      ::= { gmplsTunnelErrorEntry 4 }

      gmplsTunnelErrorCode OBJECT-TYPE
        SYNTAX  Unsigned32
        MAX-ACCESS read-only
        STATUS  current
        DESCRIPTION
          "The primary error code associated with the last error.
           The interpretation of this error code depends on the value of
           gmplsTunnelErrorLastErrorType. If the value of
           gmplsTunnelErrorLastErrorType is noError (0) the value of this
           object should be 0 and should be ignored. If the value of
           gmplsTunnelErrorLastErrorType is protocol (2) the error should
           be interpreted in the context of the signling protocol
           identified by the mplsTunnelSignallingProto object.

           Values in excess 32767 of are not used by signaling protocols
           and may safely be used as implementation-specific error codes."
        REFERENCE
          "1. Braden, R. (Ed.) et al., Resource ReserVation Protocol --
              Version 1 Functional Specification, RFC 2205, September 1997.
           2. RSVP-TE: Extensions to RSVP for LSP Tunnels, Awduche et al.,
              RFC 3209, December 2001.
           3. Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
              (Editor), RFC 3473, January 2003."
      ::= { gmplsTunnelErrorEntry 5 }

      gmplsTunnelErrorSubcode OBJECT-TYPE
        SYNTAX  Unsigned32
        MAX-ACCESS read-only
        STATUS  current
        DESCRIPTION
          "The secondary error code associated with the last error and the
           protocol used to signal this tunnel. This value is interpreted
           in the context of the value of gmplsTunnelErrorCode.
           If the value of gmplsTunnelErrorLastErrorType is noError (0) the
           value of this object should be 0 and should be ignored."
        REFERENCE
          "1. Braden, R. (Ed.) et al., Resource ReserVation Protocol --
              Version 1 Functional Specification, RFC 2205, September 1997.
           2. RSVP-TE: Extensions to RSVP for LSP Tunnels, Awduche et al.,
              RFC 3209, December 2001.
           3. Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
              (Editor), RFC 3473, January 2003."
      ::= { gmplsTunnelErrorEntry 6 }
      gmplsTunnelErrorTLVs OBJECT-TYPE
        SYNTAX  OCTET STRING
        MAX-ACCESS read-only
        STATUS  current
        DESCRIPTION
          "The sequence of interface identifier TLVs reported with the
           error by the protocol code. The interpretation of the TLVs and
           the encoding within the protocol are described in the
           references. A value of zero in the first octet indicates that no
           TLVs are present."
         REFERENCE
          "Generalized MPLS Signaling - RSVP-TE Extensions, Berger, L.
           (Editor), RFC 3473, January 2003."
      ::= { gmplsTunnelErrorEntry 7 }

      gmplsTunnelErrorHelpString OBJECT-TYPE
        SYNTAX  DisplayString
        MAX-ACCESS read-only
        STATUS  current
        DESCRIPTION
          "A textual string containing information about the last error,
           recovery actions and support advice. If there is no help string
           this object contains a zero length string.
           If the value of gmplsTunnelErrorLastErrorType is noError (0)
           this object should contain a zero length string, but may contain
           a help string indicating that there is no error."
      ::= { gmplsTunnelErrorEntry 8 }

      -- GMPLS Notifications.

      gmplsTunnelDown NOTIFICATION-TYPE
      OBJECTS  {
        mplsTunnelAdminStatus,
        mplsTunnelOperStatus,
        gmplsTunnelErrorLastErrorType,
        gmplsTunnelErrorReporterType,
        gmplsTunnelErrorReporter,
        gmplsTunnelErrorCode,
        gmplsTunnelErrorSubcode
      }
      STATUS      current
      DESCRIPTION
           "This notification is generated when a mplsTunnelOperStatus
            object for one of the configured tunnels is about to enter the
            down state from some other state (but not from the notPresent
            state). This other state is indicated by the included value of
            mplsTunnelOperStatus.

            The objects in this notification provide additional error
```

information that indicates the reason why the tunnel has transitioned down.

```
         Note that an implementation SHOULD only issue one of
         mplsTunnelDown and gmplsTunnelDown for a single event on a
         single tunnel."
   ::= { gmplsTeNotifications 1 }

   -- End of notifications.

   -- Module compliance.

   gmplsTeGroups
     OBJECT IDENTIFIER ::= { gmplsTeConformance 1 }

   gmplsTeCompliances
     OBJECT IDENTIFIER ::= { gmplsTeConformance 2 }

   -- Compliance requirement for fully compliant implementations.
   -- 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.

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

   MODULE -- this module
   MANDATORY-GROUPS {
     gmplsTunnelGroup,
     gmplsTunnelScalarGroup,
     gmplsTunnelSignaledGroup
   }
   ::= { gmplsTeCompliances 1 }

   -- Compliance requirement for read-only compliant implementations.

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

     MODULE -- this module

   -- 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.

```
   MANDATORY-GROUPS {
     gmplsTunnelGroup,
     gmplsTunnelScalarGroup
   }

   GROUP gmplsTunnelSignaledGroup
     DESCRIPTION
       "This group is mandatory for devices which support signaled
        tunnel set up, in addition to gmplsTunnelGroup. The following
        constraints apply:

        mplsTunnelSignallingProto should be at least read-only returning
        a value of ldp(2), or rsvp(3)."

   GROUP gmplsTunnelIsNotIntfcGroup
     DESCRIPTION
       "This group is mandatory for devices which support tunnels that
        are not interfaces, in addition to gmplsTunnelGroup. The
        following constraints apply:
        gmplsTunnelIsIf must at least be read-only returning no(0)."

   GROUP gmplsTunnelIsIntfcGroup
     DESCRIPTION
       "This group is mandatory for devices which support tunnels that
        are interfaces, in addition to gmplsTunnelGroup."

   GROUP gmplsTunnelOptionalGroup
     DESCRIPTION
       "Objects in this group are optional."

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

   -- GMPLS Tunnel scalars.
   -- All scalars have max access read-only

   -- gmplsTunnelTable

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

   OBJECT gmplsTunnelLSPEncoding
     SYNTAX IANAGmplsLSPEncoding
     MIN-ACCESS  read-only
     DESCRIPTION
       "Write access is not required."
```

```
   OBJECT gmplsTunnelSwitchingType
     SYNTAX IANAGmplsSwitchingType
     MIN-ACCESS  read-only
     DESCRIPTION
       "Write access is not required."

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

   OBJECT gmplsTunnelGPid
     SYNTAX IANAGmplsGPid
     MIN-ACCESS  read-only
     DESCRIPTION
       "Write access is not required."

   OBJECT gmplsTunnelSecondary
     SYNTAX TruthValue
     MIN-ACCESS  read-only
     DESCRIPTION
       "Write access is not required."

   OBJECT gmplsTunnelDirection
     SYNTAX INTEGER {
       forward (0),
       bidirectional (1)
     }
     MIN-ACCESS  read-only
     DESCRIPTION
       "Only forward (0) is required."

   OBJECT gmplsTunnelPathComp
     SYNTAX INTEGER {
       dynamicFull(1),   -- CSPF fully computed
       explicit(2),      -- fully specified path
       dynamicPartial(3) -- CSPF partially computed
     }

     MIN-ACCESS  read-only
     DESCRIPTION
       "Only explicit (2) is required."

   OBJECT gmplsTunnelUpNotRecip
     SYNTAX  IpAddress
     MIN-ACCESS read-only
     DESCRIPTION
       "Write access is not required."
```

```
   OBJECT gmplsTunnelDownNotRecip
     SYNTAX  IpAddress
     MIN-ACCESS read-only
     DESCRIPTION
       "Write access is not required."

   OBJECT gmplsTunnelAdminStatusFlags
      SYNTAX  IANAGmplsAdminStatusFlags
      MIN-ACCESS read-only
      DESCRIPTION
       "Write access is not required."

   OBJECT gmplsTunnelExtraParamsPtr
     SYNTAX  RowPointer
     MIN-ACCESS read-only
     DESCRIPTION
       "Write access is not required."

   -- gmplsTunnelHopTable

   -- gmplsTunnelHopLabelStatuses has max access read-only

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

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

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

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

   -- gmplsTunnelARHopTable
   -- all objects have max access read-only

   -- glmpsTunnelCHopTable
   -- all objects have max access read-only

   -- gmplsTunnelReversePerfTable
   -- all objects have max access read-only
```

```
   -- gmplsTunnelErrorTable
   -- all objects have max access read-only

   ::= { gmplsTeCompliances 2 }

   -- Units of conformance.

   gmplsTunnelGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelDirection,
       gmplsTunnelReversePerfPackets,
       gmplsTunnelReversePerfHCPackets,
       gmplsTunnelReversePerfErrors,
       gmplsTunnelReversePerfBytes,
       gmplsTunnelReversePerfHCBytes,
       gmplsTunnelErrorLastErrorType,
       gmplsTunnelErrorLastTime,
       gmplsTunnelErrorReporterType,
       gmplsTunnelErrorReporter,
       gmplsTunnelErrorCode,
       gmplsTunnelErrorSubcode,
       gmplsTunnelErrorTLVs,
       gmplsTunnelErrorHelpString
     }
     STATUS  current
     DESCRIPTION
       "Necessary, but not sufficient, set of objects to implement
        tunnels. In addition, depending on the type of the tunnels
        supported (for example, manually configured or signaled,
        persistent or non-persistent, etc.), the following other
        groups defined below are mandatory:

        gmplsTunnelSignaledGroup, gmplsTunnelIsNotIntfcGroup
        and/or gmplsTunnelIsIntfcGroup."
   ::= { gmplsTeGroups 1 }
```

```
   gmplsTunnelSignaledGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelAttributes,
       gmplsTunnelLSPEncoding,
       gmplsTunnelSwitchingType,
       gmplsTunnelLinkProtection,
       gmplsTunnelGPid,
       gmplsTunnelSecondary,
       gmplsTunnelPathComp,
       gmplsTunnelUpNotRecip,
       gmplsTunnelDownNotRecip,
       gmplsTunnelAdminStatusFlags,
       gmplsTunnelHopLabelStatuses,
       gmplsTunnelHopExpLabel,
       gmplsTunnelHopExpLabelPtr,
       gmplsTunnelHopExpRvrsLabel,
       gmplsTunnelHopExpRvrsLabelPtr
     }

     STATUS  current
     DESCRIPTION
       "Objects needed to implement signaled tunnels."
   ::= { gmplsTeGroups 3 }

   gmplsTunnelScalarGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelsConfigured,
       gmplsTunnelsActive
     }
     STATUS  current
     DESCRIPTION
       "Scalar objects needed to implement MPLS tunnels."
   ::= { gmplsTeGroups 4 }

   gmplsTunnelIsIntfcGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelUnnumIf
     }
     STATUS  current
     DESCRIPTION
       "Objects needed to implement tunnels that are interfaces."
   ::= { gmplsTeGroups 5 }

   gmplsTunnelIsNotIntfcGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelUnnumIf
     }
     STATUS  current
     DESCRIPTION
```

```
        "Objects needed to implement tunnels that are not interfaces."
    ::= { gmplsTeGroups 6 }
```

```
   gmplsTunnelOptionalGroup OBJECT-GROUP
     OBJECTS {
       gmplsTunnelExtraParamsPtr,
       gmplsTunnelARHopLabelStatuses,
       gmplsTunnelARHopExpLabel,
       gmplsTunnelARHopExpLabelPtr,
       gmplsTunnelARHopExpRvrsLabel,
       gmplsTunnelARHopExpRvrsLabelPtr,
       gmplsTunnelARHopProtection,
       gmplsTunnelCHopLabelStatuses,
       gmplsTunnelCHopExpLabel,
       gmplsTunnelCHopExpLabelPtr,
       gmplsTunnelCHopExpRvrsLabel,
       gmplsTunnelCHopExpRvrsLabelPtr
     }
     STATUS  current
     DESCRIPTION
       "The objects in this group are optional."
   ::= { gmplsTeGroups 7 }

   gmplsTeNotificationGroup NOTIFICATION-GROUP
     NOTIFICATIONS {
        gmplsTunnelDown
     }
     STATUS  current
     DESCRIPTION
       "Set of notifications implemented in this module. None is
        mandatory."
   ::= { gmplsTeGroups 8 }

   END
```

9. Security Considerations

It is clear that the MIB modules described in this document in association with the MPLS-TE-STD-MIB are potentially useful for monitoring of MPLS and GMPLS tunnels. These MIB modules can also be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

There are a number of management objects defined in these MIB modules with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- the gmplsTunnelTable and gmplsTunnelHopTable collectively contain objects to provision GMPLS tunnels interfaces at their ingress LSRs. Unauthorized write access to objects in these tables, could result in disruption of traffic on the network. This is especially true if a tunnel has already been established. The use of stronger mechanisms such as SNMPv3 security should be considered where possible. Specifically, SNMPv3 VACM and USM MUST be used with any SNMPv3 agent which implements these MIB modules.

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

- the gmplsTunnelTable, gmplsTunnelHopTable, gmplsTunnelARHopTable, gmplsTunnelCHopTable, gmplsTunnelReversePerfTable, gmplsTunnelErrorTable collectively show the tunnel network topology and status. If an Administrator does not want to reveal this information, then these tables should be considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure "for example by using IPSec", even then, there is no control as to who on the secure network is allowed to access and GET/SET "read/change/create/delete" the objects in these MIB modules. It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework "see [RFC3410], section 8", including full support for the SNMPv3 cryptographic mechanisms "for authentication and privacy".

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

10. Acknowledgments

This draft is the work of the five authors listed in the Authors' Addresses section.

This document extends [RFC3812]. The authors would like to express their gratitude to all those who worked on that earlier MIB document. Thanks also to Tony Zinicola and Jeremy Crossen for their valuable contributions during an early implementation, and to Baktha Muralidharan and Tom Petch for their review comments.

Special thanks to Joan Cucchiara and Len Nieman for their help with compilation issues.

11. IANA Considerations

As requested in the GMPLS-TC-STD-MIB [GMPLSTCMIB], GMPLS related standards track MIB modules should be rooted under the mplsStdMIB subtree. There is one GMPLS MIB Module contained in this document, and the following "IANA Considerations" subsection requests IANA for a new assignment under the mplsStdMIB subtree. New assignments in the mplsStdMIB subtree can only be made via a Standards Action as specified in [RFC2434].

11.1. IANA Considerations for GMPLS-TE-STD-MIB

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

11.2. Dependence on IANA MIB Modules

Three MIB objects in this MIB module (gmplsTunnelLSPEncoding, gmplsTunnelSwitchingType, and gmplsTunnelGPid) use textual conventions imported from the IANA-GMPLS-MIB. The purpose of defining these textual conventions in a separate MIB module is to allow additional values to be defined without having to issue a new version of this document. The Internet Assigned Numbers Authority (IANA) is responsible for the assignment of all Internet numbers; it will administer the values associated with these textual conventions.

The rules for additions or changes to the IANA-GMPLS-MIB are outlined in the DESCRIPTION clause associated with its MODULE-IDENTITY statement.

The current versions of the IANA-GMPLS-MIB can be accessed from the IANA home page at: "http://www.iana.org/".

11.2.1. IANA-GMPLS-MIB Definition

This is a temporary section intended to supply the base definition of an IANA MIB module. The normal procedure is that this MIB module is moved into the direct control of IANA, at which time this section should be deleted from this document.

IANA is requested to assign an OID to the IANA-GMPLS-MIB module specified in this document as { transmission YYY }.

```
   IANA-GMPLS-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, transmission       FROM SNMPv2-SMI  -- RFC2578
       TEXTUAL-CONVENTION                  FROM SNMPv2-TC;  -- RFC2579

   ianaGmpls MODULE-IDENTITY
       LAST-UPDATED "200505200001Z" -- 20 May 2005 00:00:01 GMT
       ORGANIZATION "IANA"
       CONTACT-INFO
                  "        Internet Assigned Numbers Authority
                   Postal: USC/Information Sciences Institute
                           4676 Admiralty Way, Marina del Rey, CA 90292
                   Tel:    +1 310 822 1511
                   E-Mail: iana@isi.edu"
       DESCRIPTION
            "The MIB module which defines the GMPLS textual
             conventions for use as enumerations within GMPLS MIB
             modules, thus protecting those MIB modules from changes
             to the enumerations."
       -- Revision history.
          REVISION
               "200505200001Z" -- 20 May 2005 00:00:01 GMT
         DESCRIPTION
           -- RFC Editor: Please see the IANA Considerations Section.
           -- RFC-editor please fill in XXXX
           "Initial version issued as part of RFC XXXX."
       ::= { transmission YYY }
```

```
   IANAGmplsLSPEncoding ::= TEXTUAL-CONVENTION
       STATUS        current
       DESCRIPTION
            "This data type is used as the syntax of the
             gmplsTunnelLSPEncoding object in the definition of
             GMPLS-TE-MIB's gmplsTunnelTable.

             The definition of this textual convention with the
             addition of newly assigned values is published
             periodically by the IANA, in either the Assigned
             Numbers RFC, or some derivative of it specific to
             Internet Network Management number assignments. (The
             latest arrangements can be obtained by contacting the
             IANA.)

             Requests for new values should be made to IANA via
             email (iana@isi.edu).

             gmplsTunnelLSPEncoding is used to represent and control
             the LSP encoding type of an LSP signaled by a GMPLS
             signaling protocol. The relationship between the
             assignment of gmplsTunnelLSPEncoding values and of the
             values used to represent LSP encoding types within the
             GMPLS signaling protocols is solely the purview of IANA
             and is subject to change without notice."
       REFERENCE
            "1. Berger, L., et al., Generalized Multi-Protocol
                Label Switching (GMPLS) Signaling Functional
                Description, RFC 3471, January 2003.
             2. D. Papadimitriou (Editor), Generalized MPLS
                Signalling Extensions for G.709 Optical Transport
                Networks Control, draft-ietf-ccamp-gmpls-g709,
                work in progress."
       SYNTAX  INTEGER {
                  tunnelLspNotGmpls (0),          -- GMPLS is not in use
                  tunnelLspPacket (1),            -- Packet
                  tunnelLspEthernet (2),          -- Ethernet
                  tunnelLspAnsiEtsiPdh (3),       -- PDH
                  -- the value 4 is deprecated
                  tunnelLspSdhSonet (5),          -- SDH or SONET
                  -- the value 6 is deprecated
                  tunnelLspDigitalWrapper (7),    -- Digital Wrapper
                  tunnelLspLambda (8),            -- Lambda
                  tunnelLspFiber (9),             -- Fiber
                  -- the value 10 is deprecated
                  tunnelLspFiberChannel (11),     -- Fiber Channel
                  tunnelDigitalPath (12),         -- Digital Path
                  tunnelOpticalChannel (13)       -- Optical Channel
                }
```

```
   IANAGmplsSwitchingType ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
            "This data type is used as the syntax of the
             gmplsTunnelSwitchingType object in the definition of
             GMPLS-TE-MIB's gmplsTunnelTable.

             The definition of this textual convention with the
             addition of newly assigned values is published
             periodically by the IANA, in either the Assigned
             Numbers RFC, or some derivative of it specific to
             Internet Network Management number assignments. (The
             latest arrangements can be obtained by contacting the
             IANA.)

             Requests for new values should be made to IANA via
             email (iana@isi.edu).

             gmplsTunnelSwitchingType is used to represent and
             control the LSP switching type of an LSP signaled by a
             GMPLS signaling protocol. The relationship between the
             assignment of gmplsTunnelSwitchingType values and of the
             values used to represent LSP switching types within the
             GMPLS signaling protocols is solely the purview of IANA
             and is subject to change without notice."
       REFERENCE
            "1. Kompella, K., Rekhter, Y. (Editors), Routing Extensions
                in Support of Generalized Multi-Protocol Label Switching
                draft-ietf-ccamp-gmpls-routing, work in progress.
             2. Berger, L., et al., Generalized Multi-Protocol
                Label Switching (GMPLS) Signaling Functional
                Description, RFC 3471, January 2003."
       SYNTAX  INTEGER {
                  unknown (0),   -- none of the following, or not known
                  psc1 (1),      -- Packet-Switch-Capable 1
                  psc2 (2),      -- Packet-Switch-Capable 2
                  psc3 (3),      -- Packet-Switch-Capable 3
                  psc4 (4),      -- Packet-Switch-Capable 4
                  l2sc (51),     -- Layer-2-Switch-Capable
                  tdm (100),     -- Time-Division-Multiplex
                  lsc (150),     -- Lambda-Switch-Capable
                  fsc (200)      -- Fiber-Switch-Capable
                }
```

```
   IANAGmplsGPid ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
            "This data type is used as the syntax of the
             gmplsTunnelGPid object in the definition of
             GMPLS-TE-MIB's gmplsTunnelTable.
             The definition of this textual convention with the
             addition of newly assigned values is published
             periodically by the IANA, in either the Assigned
             Numbers RFC, or some derivative of it specific to
             Internet Network Management number assignments. (The
             latest arrangements can be obtained by contacting the
             IANA.)

             Requests for new values should be made to IANA via
             email (iana@isi.edu).

             gmplsTunnelGPid is used to represent and control the LSP
             Generalized Protocol Identifier (G-PID) of an LSP
             signaled by a GMPLS signaling protocol. The relationship
             between the assignment of gmplsTunnelGPid values and of
             the values used to represent G-PIDs within the GMPLS
             signaling protocols is solely the purview of IANA and is
             subject to change without notice."
        REFERENCE
            "1. Berger, L., et al., Generalized Multi-Protocol
                Label Switching (GMPLS) Signaling Functional
                Description, RFC 3471, January 2003.
             2. D. Papadimitriou (Editor), Generalized MPLS
                Signalling Extensions for G.709 Optical Transport
                Networks Control, draft-ietf-ccamp-gmpls-g709,
                work in progress."
        SYNTAX  INTEGER {
                  unknown(0),      -- unknown or none of the following
                  asynchE4(5),
                  asynchDS3T3(6),
                  asynchE3(7),
                  bitsynchE3(8),
                  bytesynchE3(9),
                  asynchDS2T2(10),
                  bitsynchDS2T2(11),
                  asynchE1(13),
                  bytesynchE1(14),
                  bytesynch31ByDS0(15),
                  asynchDS1T1(16),
                  bitsynchDS1T1(17),
                  bytesynchDS1T1(18),
                  vc1vc12(19),
                  ds1SFAsynch(22),
                  ds1ESFAsynch(23),
                  ds3M23Asynch(24),
                  ds3CBitParityAsynch(25),
                  vtLovc(26),
                  stsSpeHovc(27),
                  posNoScramble16BitCrc(28),
                  posNoScramble32BitCrc(29),
                  posScramble16BitCrc(30),
                  posScramble32BitCrc(31),
                  atm(32),
                  ethernet(33),
                  sdhSonet(34),
                  digitalwrapper(36),
                  lambda(37),
                  ansiEtsiPdh (38),
                  lapsSdh (40),
                  fddi (41),
                  dqdb (42),
                  fiberChannel3 (43),
                  hdlc (44),
                  ethernetV2DixOnly (45),
                  ethernet802dot3Only (46),
                  g709ODUj (47),
                  g709OTUk (48),
                  g709CBRorCBRa (49),
                  g709CBRb (50),
                  g709BSOT (51),
                  g709BSNT (52),
                  gfpIPorPPP (53),
                  gfpEthernetMAC (54),
                  gfpEthernetPHY (55),
                  g709ESCON (56),
                  g709FICON (57),
                  g709FiberChannel (58)
                }
```

```
IANAGmplsAdminStatusFlags ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
         "This data type is used as the syntax of the
          gmplsTunnelAdminStatusFlags object in the definition of
          GMPLS-TE-MIB's gmplsTunnelTable.

          The definition of this textual convention with the
          addition of newly assigned values is published
          periodically by the IANA, in either the Assigned
          Numbers RFC, or some derivative of it specific to
          Internet Network Management number assignments. (The
          latest arrangements can be obtained by contacting the
          IANA.)

          Requests for new values should be made to IANA via
          email (iana@isi.edu).

          gmplsTunnelAdminStatusFlags determines the setting of the
          Admin Status flags in the Admin Status object or TLV, as
          described in RFC 3471. Setting this object to a non-zero
          value will result in the inclusion of the Admin Status
          object or TLV on signaling messages.

          The relationship between the assignment of
          gmplsTunnelAdminStatusFlags values and of the bit flags
          in the Admin Status object or TLV within the GMPLS
          signaling protocols is solely the purview of IANA and is
          subject to change without notice."
   REFERENCE
         "Berger, L., et al., Generalized Multi-Protocol Label
          Switching (GMPLS) Signaling Functional Description,
          RFC 3471, January 2003."
   SYNTAX  BITS {
                  delInProgress (0),
                  adminDown (1),
                  testing (2),
                  reflect (31)
                }

END
```

12. References

12.1. 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., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

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

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

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

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

[RFC3212] Jamoussi, B., Aboul-Magd, O., Andersson, L., Ashwood-Smith, P., Hellstrand, F., Sundell, K., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Halpern, J., Heinanen, J., Kilty, T., Malis, A., and P. Vaananen, "Constraint-Based LSP Setup using LDP", RFC 3212, December 2001.

[RFC3471] Berger, L. (Editor), "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[RFC3472] Ashwood-Smith, P., Berger, L. (Editors), "Generalized MPLS Signaling - CR-LDP Extensions", RFC 3472, January 2003.

[RFC3473] Berger, L. (Editor), "Generalized MPLS Signaling - RSVP-TE Extensions", RFC 3473, January 2003.

[RFC3477] Kompella, K. and Rekhter, Y., "Signalling Unnumbered

Links in RSVP-TE", RFC 3477, January 2003.

[RFC3480]   Kompella, K., Rekhter, Y. and Kullberg, A., "Signalling Unnumbered Links in CR-LDP", RFC 3480, February 2003.

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

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

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

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

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

[GMPLSLSRMIB]   Nadeau, T., Farrel, A., (Editors) "Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) Management Information Base", draft-ietf-ccamp-gmpls-lsr-mib, work in progress.

[GMPLSOSPF]   Kompella, K., et al., "OSPF Extensions in Support of Generalized MPLS", draft-ietf-ccamp-ospf-gmpls-extensions, work in progress.

[GMPLSTCMIB]   Nadeau, T., Farrel, A., (Editors) "Definitions of Textual Conventions for Multiprotocol Label Switching (MPLS) Management", draft-ietf-ccamp-gmpls-te-mib, work in progress.

12.2. Informational References

[RFC2026]   S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[RFC2434]   Narten, T. and H. Alvestrand., "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC3413] Levi, D., Meyer, P., Stewart, B., "SNMP Applications", RFC 3413, December 2002.

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

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", RFC 3411, December 2002.

[GMPLS-G709] D. Papadimitriou (Editor), "Generalized MPLS Signalling Extensions for G.709 Optical Transport Networks Control", draft-ietf-ccamp-gmpls-g709, work in progress.

13. Authors' Addresses

Thomas D. Nadeau
Cisco Systems, Inc.
300 Apollo Drive
Chelmsford, MA 01824
Phone: +1-978-244-3051
Email: tnadeau@cisco.com

Cheenu Srinivasan
Bloomberg L.P.
731 Lexington Ave.
New York, NY 10022
Phone: +1-212-617-3682
Email: cheenu@bloomberg.net

Adrian Farrel
Old Dog Consulting
Phone: +44-(0)-1978-860944
Email: adrian@olddog.co.uk

Tim Hall
Data Connection Ltd.
100 Church Street
Enfield, Middlesex
EN2 6BQ, UK
Phone: +44 20 8366 1177
Email: tim.hall@dataconnection.com

Ed Harrison
Data Connection Ltd.
100 Church Street
Enfield, Middlesex
EN2 6BQ, UK
Phone: +44 20 8366 1177
Email: ed.harrison@dataconnection.com

14. Intellectual Property Considerations

15. Full Copyright Statement