Network Working Group
Request for Comment: 4802
Category: Standards Track

T. Nadeau, Ed.
Cisco Systems, Inc.
A. Farrel, Ed.
Old Dog Consulting

February 2007

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

Status of This Memo

Copyright Notice

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 (TE). 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.

The MIB modules in this document should be used in conjunction with the companion document [RFC4803] for GMPLS-based traffic engineering configuration and management.

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

1.1. Migration Strategy

MPLS-TE Label Switched paths (LSPs) may be modeled and managed using the MPLS-TE-STD-MIB module [RFC3812].

Label Switching Routers (LSRs) may be migrated to model and manage their TE LSPs using the MIB modules in this document in order to migrate the LSRs to GMPLS support, or to take advantage of additional MIB objects defined in these MIB modules that are applicable to MPLS-TE.

The GMPLS TE MIB module (GMPLS-TE-STD-MIB) defined in this document extends the MPLS-TE-STD-MIB module [RFC3812] through a series of augmentations and sparse augmentations of the MIB tables. The only additions are for support of GMPLS or to support the increased complexity of MPLS and GMPLS systems.

In order to migrate from MPLS-TE-STD-MIB support to GMPLS-TE-STD-MIB support, an implementation needs only to add support for the additional tables and objects defined in GMPLS-TE-STD-MIB. The gmplsTunnelLSPEncoding may be set to tunnelLspNotGmpls to allow an MPLS-TE LSP tunnel to benefit from the additional objects and tables of GMPLS-LSR-STD-MIB without supporting the GMPLS protocols.

The companion document for modeling and managing GMPLS-based LSRs [RFC4803] extends the MPLS-LSR-STD-MIB module [RFC3813] with the same intentions.

Textual conventions are defined in [RFC3811] and the IANA-GMPLS-TC-MIB module.

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 is performed using a cross-connect.

These segment and cross-connect objects are defined in the MPLS Label Switching Router MIB (MPLS-LSR-STD-MIB) [RFC3813], but see also the GMPLS Label Switching Router MIB (GMPLS-LSR-STD-MIB) [RFC4803] for the GMPLS-specific extensions to these objects.

3. The Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. Outline

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

- Setting up tunnels with appropriate MPLS configuration parameters using [RFC3812].

- Extending the tunnel definitions with GMPLS configuration parameters.

- Configuring loose and strict source routed tunnel hops.

These actions may need to be accompanied with corresponding actions using [RFC3813] and [RFC4803] 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 following tables contain MIB objects for performing the actions listed above when they cannot be performed solely using MIB objects defined in MPLS-TE-STD-MIB [RFC3812].

- Tunnel table (gmplsTunnelTable) for providing GMPLS-specific tunnel configuration parameters.

- Tunnel hop, actual tunnel hop, and computed tunnel 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, the GMPLS-TE-STD-MIB module contains a new notification.

- The GMPLS Tunnel Down Notification (gmplsTunnelDown) should be used for all GMPLS tunnels in place of the mplsTunnelDown notification defined in [RFC3812]. An implementation must not issue both the gmplsTunnelDown and the mplsTunnelDown notifications for the same event. As well as indicating that a tunnel has transitioned to operational down state, this new notification 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 (MPLS-TE-STD-MIB [RFC3812]) 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 multipoint-to-point and point-to-multipoint 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 the GMPLS-TE-STD-MIB module from changes (in particular, extensions) to the range of enumerations supported by the signaling

protocols, these MIB objects use textual conventions with values maintained 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 the 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 set up. 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 and when and how it was reported, and it can give recovery advice through an admin string.

5.6. gmplsTunnelReversePerfTable

The 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.

5.7. Use of 32-bit and 64-bit Counters

64-bit counters are provided in the GMPLS-TE-STD-MIB module for high-speed interfaces where the use of 32-bit counters might be impractical. The requirements on the use of 32-bit and 64-bit counters (copied verbatim from [RFC2863]) are as follows:

> For interfaces that operate at 20,000,000 (20 million) bits per second or less, 32-bit byte and packet counters MUST be supported. For interfaces that operate faster than 20,000,000 bits/second, and slower than 650,000,000 bits/second, 32-bit packet counters MUST be supported and 64-bit octet counters MUST be supported. For interfaces that operate at 650,000,000 bits/second or faster, 64-bit packet counters AND 64-bit octet counters MUST be supported.

6. Cross-referencing to the gmplsLabelTable

The gmplsLabelTable is found in the GMPLS-LABEL-STD-MIB module in [RFC4803] 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 (gmplsTunnelHopTable, gmplsTunnelCHopTable, and gmplsTunnelARHopTable) and the segment tables in [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. Furthermore, 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 be made available to represent the labels. This is achieved through the gmplsLabelTable contained in the GMPLS-LABEL-STD-MIB [RFC4803].

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 an 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 MPLS-TE-STD-MIB [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        = 192.0.2.1,
  mplsTunnelEgressLSRId         = 192.0.2.2,
  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(1,1,192.0.2.1,192.0.2.2):
{
  gmplsTunnelUnnumIf             = true(1),
  gmplsTunnelAttributes          = labelRecordingRequired(1),
  gmplsTunnelLSPEncoding         = tunnelLspLambda,
  gmplsTunnelSwitchingType       = lsc,
  gmplsTunnelLinkProtection      = shared(2),
  gmplsTunnelGPid                = lambda,
  gmplsTunnelSecondary           = false(2),
  gmplsTunnelDirection           = bidirectional(1)
  gmplsTunnelPathComp            = explicit(2),
  gmplsTunnelSendPathNotifyRecipientType = ipv4(1),
  gmplsTunnelSendPathNotifyRecipient     = 'C0000201'H,
  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 in our example. We have used the fictitious LSR identified by "192.0.2.1" as our head-end router.

```
In mplsTunnelHopTable:
{
  mplsTunnelHopListIndex         = 1,
  mplsTunnelPathOptionIndex      = 1,
  mplsTunnelHopIndex             = 1,
  mplsTunnelHopAddrType          = ipv4(1),
  mplsTunnelHopIpv4Addr          = 192.0.2.1,
  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 "192.0.2.2" as our tail-end router.

```
In mplsTunnelHopTable:
{
  mplsTunnelHopListIndex         = 1,
  mplsTunnelPathOptionIndex      = 1,
  mplsTunnelHopIndex             = 2,
  mplsTunnelHopAddrType          = ipv4(1),
  mplsTunnelHopIpv4Addr          = 192.0.2.2,
  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                 = gmplsFreeformLabel(3),
  gmplsLabelFreeform             = 0xFEDCBA9876543210
  gmplsLabelRowStatus            = createAndGo(4)
}

In gmplsTunnelHopTable(1,1,1):
{
  gmplsTunnelHopLabelStatuses           = forwardPresent(0)
                                             +reversePresent(1),
  gmplsTunnelHopExplicitForwardLabelPtr = gmplsLabelTable(2,1,0)
  gmplsTunnelHopExplicitReverseLabelPtr = 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(1,1,192.0.2.1,192.0.2.2)
{
  mplsTunnelRowStatus           = active(1)
}
```

8. GMPLS Traffic Engineering MIB Module

This MIB module makes reference to the following documents: [RFC2205], [RFC2578], [RFC2579], [RFC2580], [RFC3209], [RFC3411], [RFC3471], [RFC3473], [RFC3477], [RFC3812], [RFC4001], and [RFC4202].

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

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
  Unsigned32, Counter32, Counter64, zeroDotZero, Gauge32
    FROM SNMPv2-SMI                                 -- RFC 2578
  MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF                                -- RFC 2580
  TruthValue, TimeStamp, RowPointer
    FROM SNMPv2-TC                                  -- RFC 2579
  InetAddress, InetAddressType
    FROM INET-ADDRESS-MIB                           -- RFC 4001
  SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB                         -- RFC 3411
  mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
  mplsTunnelEgressLSRId, mplsTunnelHopListIndex,
  mplsTunnelHopPathOptionIndex, mplsTunnelHopIndex,
  mplsTunnelARHopListIndex, mplsTunnelARHopIndex,
  mplsTunnelCHopListIndex, mplsTunnelCHopIndex,
  mplsTunnelEntry,
  mplsTunnelAdminStatus, mplsTunnelOperStatus,
  mplsTunnelGroup, mplsTunnelScalarGroup
    FROM MPLS-TE-STD-MIB                            -- RFC3812
  IANAGmplsLSPEncodingTypeTC, IANAGmplsSwitchingTypeTC,
  IANAGmplsGeneralizedPidTC, IANAGmplsAdminStatusInformationTC
    FROM IANA-GMPLS-TC-MIB
  mplsStdMIB
    FROM MPLS-TC-STD-MIB                            -- RFC 3811
;
```

```
gmplsTeStdMIB MODULE-IDENTITY
      LAST-UPDATED
         "200702270000Z" -- 27 February 2007 00:00:00 GMT
      ORGANIZATION
        "IETF 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 directly
         to the CCAMP working group mailing list at
         ccamp@ops.ietf.org."

      DESCRIPTION
        "Copyright (C) The IETF Trust (2007).  This version of
         this MIB module is part of RFC 4802; see the RFC itself for
         full legal notices.

         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
        "200702270000Z" -- 27 February 2007 00:00:00 GMT
      DESCRIPTION
        "Initial version issued as part of RFC 4802."
::= { mplsStdMIB 13 }

gmplsTeNotifications OBJECT IDENTIFIER ::= { gmplsTeStdMIB 0 }
gmplsTeScalars OBJECT IDENTIFIER ::= { gmplsTeStdMIB 1 }
gmplsTeObjects OBJECT IDENTIFIER ::= { gmplsTeStdMIB 2 }
gmplsTeConformance OBJECT IDENTIFIER ::= { gmplsTeStdMIB 3 }

gmplsTunnelsConfigured OBJECT-TYPE
  SYNTAX  Gauge32
  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  Gauge32
  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 }

gmplsTunnelTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF GmplsTunnelEntry
  MAX-ACCESS not-accessible
  STATUS  current
  DESCRIPTION
    "The gmplsTunnelTable sparsely extends the mplsTunnelTable of
     MPLS-TE-STD-MIB.  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 multipoint-to-point and point-to-multipoint
     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 the
     mplsTunnelRowStatus in the corresponding entry in the
     mplsTunnelTable.  When the corresponding mplsTunnelRowStatus has
     value active(1), a row in this table may not be created or
     modified.

     The exception to this rule is the
     gmplsTunnelAdminStatusInformation object, which can be modified
     while the tunnel is active."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812."
::= { 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 via SNMP SET
     commands, or in response to signaling protocol events."
  INDEX {
    mplsTunnelIndex,
    mplsTunnelInstance,
    mplsTunnelIngressLSRId,
    mplsTunnelEgressLSRId
  }
::= { gmplsTunnelTable 1 }

  GmplsTunnelEntry ::= SEQUENCE {
   gmplsTunnelUnnumIf                       TruthValue,
   gmplsTunnelAttributes                    BITS,
   gmplsTunnelLSPEncoding                   IANAGmplsLSPEncodingTypeTC,
   gmplsTunnelSwitchingType                 IANAGmplsSwitchingTypeTC,
   gmplsTunnelLinkProtection                BITS,
   gmplsTunnelGPid                          IANAGmplsGeneralizedPidTC,
   gmplsTunnelSecondary                     TruthValue,
   gmplsTunnelDirection                     INTEGER,
   gmplsTunnelPathComp                      INTEGER,
   gmplsTunnelUpstreamNotifyRecipientType   InetAddressType,
   gmplsTunnelUpstreamNotifyRecipient       InetAddress,
   gmplsTunnelSendResvNotifyRecipientType   InetAddressType,
   gmplsTunnelSendResvNotifyRecipient       InetAddress,
   gmplsTunnelDownstreamNotifyRecipientType InetAddressType,
   gmplsTunnelDownstreamNotifyRecipient     InetAddress,
   gmplsTunnelSendPathNotifyRecipientType   InetAddressType,
   gmplsTunnelSendPathNotifyRecipient       InetAddress,
   gmplsTunnelAdminStatusFlags        IANAGmplsAdminStatusInformationTC,
   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 by an entry in the interfaces group table
     (the ifTable) with ifType set to mpls(166).
```

```
     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
    "1. Signalling Unnumbered Links in RSVP-TE, RFC 3477."
  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 is set to indicate that label information should be
       included when doing a route record.  This bit is not valid
       unless the recordRoute bit is set."
  REFERENCE
    "1. RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209,
        sections 4.4.3, 4.7.1, and 4.7.2."
  DEFVAL  { { } }
::= { gmplsTunnelEntry 2 }

gmplsTunnelLSPEncoding OBJECT-TYPE
  SYNTAX  IANAGmplsLSPEncodingTypeTC
  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
     IANAGmplsLSPEncodingTypeTC found in the IANA-GMPLS-TC-MIB
     module."
  DEFVAL  { tunnelLspNotGmpls }
::= { gmplsTunnelEntry 3 }

gmplsTunnelSwitchingType OBJECT-TYPE
  SYNTAX  IANAGmplsSwitchingTypeTC
  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
     IANAGmplsSwitchingTypeTC found in the IANA-GMPLS-TC-MIB module.

     This object is only meaningful if gmplsTunnelLSPEncodingType
     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
       This flag is set to indicate 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
       This flag is set to indicate that the LSP should not use any
       link layer protection.

     shared
       This flag is set to indicate that a shared link layer
       protection scheme, such as 1:N protection, should be used to
       support the LSP.

     dedicatedOneToOne
       This flag is set to indicate that a dedicated link layer
       protection scheme, i.e., 1:1 protection, should be used to
       support the LSP.

     dedicatedOnePlusOne
       This flag is set to indicate that a dedicated link layer
       protection scheme, i.e., 1+1 protection, should be used to
       support the LSP.

     enhanced
       This flag is set to indicate 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
     "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
         Functional Description, RFC 3471, section 7.1."
  DEFVAL  { { } }
::= { gmplsTunnelEntry 5 }

gmplsTunnelGPid OBJECT-TYPE
  SYNTAX  IANAGmplsGeneralizedPidTC
  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
     IANAGmplsGeneralizedPidTC found in the IANA-GMPLS-TC-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
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 7.1."
  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 Constrained
```

```
       Shortest Path First (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.

     When an entry is present in the gmplsTunnelTable for a
     tunnel, gmplsTunnelPathComp MUST be used and any
     corresponding mplsTunnelHopEntryPathComp object in the
     mplsTunnelHopTable MUST be ignored and SHOULD not be set.

     mplsTunnelHopTable and mplsTunnelHopEntryPathComp are part of
     MPLS-TE-STD-MIB.

     This object should be ignored if the value of
     gmplsTunnelLSPEncoding is 'tunnelLspNotGmpls'."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812."
  DEFVAL { dynamicFull }
::= { gmplsTunnelEntry 9 }

gmplsTunnelUpstreamNotifyRecipientType OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
   "This object is used to aid in interpretation of
    gmplsTunnelUpstreamNotifyRecipient."
  DEFVAL { unknown }
::= { gmplsTunnelEntry 10 }

gmplsTunnelUpstreamNotifyRecipient OBJECT-TYPE
  SYNTAX  InetAddress
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
```

```
    "Indicates the address of the upstream recipient for Notify
     messages relating to this tunnel and issued by this LSR.  This
     information is typically received from an upstream LSR in a Path
     message.

     This object is only valid when signaling a tunnel using RSVP.

     It is also not valid at the head end of a tunnel since there are
     no upstream LSRs to which to send a Notify message.

     This object is interpreted in the context of the value of
     gmplsTunnelUpstreamNotifyRecipientType. If this object is set to
     0, the value of gmplsTunnelUpstreamNotifyRecipientType MUST be
     set to unknown(0)."
  REFERENCE
    "1. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 4.2. "
  DEFVAL { '00000000'H } -- 0.0.0.0
::= { gmplsTunnelEntry 11 }

gmplsTunnelSendResvNotifyRecipientType OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
   "This object is used to aid in interpretation of
    gmplsTunnelSendResvNotifyRecipient."
  DEFVAL { unknown }
::= { gmplsTunnelEntry 12 }

gmplsTunnelSendResvNotifyRecipient OBJECT-TYPE
  SYNTAX  InetAddress
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
    "Indicates to an upstream LSR the address to which it should send
     downstream 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 since no Resv
     messages are sent from that LSR for this tunnel.

     If set to 0, no Notify Request object will be included in the
     outgoing Resv messages.

     This object is interpreted in the context of the value of
     gmplsTunnelSendResvNotifyRecipientType. If this object is set to
```

```
     0, the value of gmplsTunnelSendResvNotifyRecipientType MUST be
     set to unknown(0)."
  REFERENCE
    "1. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 4.2. "
  DEFVAL { '00000000'H } -- 0.0.0.0
::= { gmplsTunnelEntry 13 }

gmplsTunnelDownstreamNotifyRecipientType OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
   "This object is used to aid in interpretation of
    gmplsTunnelDownstreamNotifyRecipient."
  DEFVAL { unknown }
::= { gmplsTunnelEntry 14 }

gmplsTunnelDownstreamNotifyRecipient OBJECT-TYPE
  SYNTAX  InetAddress
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
    "Indicates the address of the downstream recipient for Notify
     messages relating to this tunnel and issued by this LSR.  This
     information is typically received from an upstream LSR in a Resv
     message.  This object is only valid when signaling a tunnel using
     RSVP.

     It is also not valid at the tail end of a tunnel since there are
     no downstream LSRs to which to send a Notify message.

     This object is interpreted in the context of the value of
     gmplsTunnelDownstreamNotifyRecipientType. If this object is set
     to 0, the value of gmplsTunnelDownstreamNotifyRecipientType MUST
     be set to unknown(0)."
  REFERENCE
    "1. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 4.2.
    "
  DEFVAL { '00000000'H } -- 0.0.0.0
::= { gmplsTunnelEntry 15 }

gmplsTunnelSendPathNotifyRecipientType OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
```

```
   "This object is used to aid in interpretation of
    gmplsTunnelSendPathNotifyRecipient."
  DEFVAL { unknown }
::= { gmplsTunnelEntry 16 }

gmplsTunnelSendPathNotifyRecipient OBJECT-TYPE
  SYNTAX  InetAddress
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
    "Indicates to a downstream LSR the address to which it should
     send upstream 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 since no Path
     messages are sent from that LSR for this tunnel.

     If set to 0, no Notify Request object will be included in the
     outgoing Path messages.

     This object is interpreted in the context of the value of
     gmplsTunnelSendPathNotifyRecipientType.  If this object is set to
     0, the value of gmplsTunnelSendPathNotifyRecipientType MUST be
     set to unknown(0)."
  REFERENCE
    "1. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 4.2. "
  DEFVAL { '00000000'H } -- 0.0.0.0
::= { gmplsTunnelEntry 17 }

gmplsTunnelAdminStatusFlags OBJECT-TYPE
   SYNTAX   IANAGmplsAdminStatusInformationTC
   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
      IANAGmplsAdminStatusInformationTC found in the
      IANA-GMPLS-TC-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
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 8."
  DEFVAL  { { } }
  ::= { gmplsTunnelEntry 18 }

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,
     Synchronous Optical Network (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 19 }

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

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

     The storage type for this entry is given by the value
     of mplsTunnelHopStorageType in the corresponding entry in the
     mplsTunnelHopTable.

     The row status of an entry in this table is controlled by
     mplsTunnelHopRowStatus in the corresponding entry in the
     mplsTunnelHopTable.  That is, it is not permitted to create a row
```

```
     in this table, or to modify an existing row, when the
     corresponding mplsTunnelHopRowStatus has the value active(1)."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812.
     2. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473.
    "
::= { 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,
  gmplsTunnelHopExplicitForwardLabel    Unsigned32,
  gmplsTunnelHopExplicitForwardLabelPtr RowPointer,
  gmplsTunnelHopExplicitReverseLabel    Unsigned32,
  gmplsTunnelHopExplicitReverseLabelPtr 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
     gmplsTunnelHopExplicitForwardLabel or
     gmplsTunnelHopExplicitForwardLabelPtr, and
     gmplsTunnelHopExplicitReverseLabel or
```

```
     gmplsTunnelHopExplicitReverseLabelPtr 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 }

gmplsTunnelHopExplicitForwardLabel OBJECT-TYPE
  SYNTAX  Unsigned32
  MAX-ACCESS read-create
  STATUS  current
  DESCRIPTION
    "If gmplsTunnelHopLabelStatuses object indicates that a Forward
     Label is present and gmplsTunnelHopExplicitForwardLabelPtr
     contains the value zeroDotZero, then the label to use on this
     hop is represented by the value of this object."
::= { gmplsTunnelHopEntry 2 }

gmplsTunnelHopExplicitForwardLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelHopExplicitForwardLabel object."
  DEFVAL  { zeroDotZero }
::= { gmplsTunnelHopEntry 3 }

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

```
gmplsTunnelHopExplicitReverseLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelHopExplicitReverseLabel object."
  DEFVAL  { zeroDotZero }
::= { gmplsTunnelHopEntry 5 }

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

     Each row in this table is indexed by the same indexes as in the
     mplsTunnelARHopTable.  It is acceptable for some rows in the
     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 the
     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, RFC 3209.
```

```
     2. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473.
     3. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812."
::= { 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), this table
     and the mplsTunnelARHopTable (if the tables are supported and if
     the signaling protocol is recording actual route information)
     contain 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,
  gmplsTunnelARHopExplicitForwardLabel    Unsigned32,
  gmplsTunnelARHopExplicitForwardLabelPtr RowPointer,
  gmplsTunnelARHopExplicitReverseLabel    Unsigned32,
  gmplsTunnelARHopExplicitReverseLabelPtr 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 gmplsTunnelARHopExplicitForwardLabel or
     gmplsTunnelARHopExplicitForwardLabelPtr, and
     gmplsTunnelARHopExplicitReverseLabel or
     gmplsTunnelARHopExplicitReverseLabelPtr 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 }

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

gmplsTunnelARHopExplicitForwardLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelARHopExplicitForwardLabel object."
::= { gmplsTunnelARHopEntry 3 }
```

```
gmplsTunnelARHopExplicitReverseLabel OBJECT-TYPE
  SYNTAX  Unsigned32
  MAX-ACCESS read-only
  STATUS  current
  DESCRIPTION
    "If the gmplsTunnelARHopLabelStatuses object indicates that a
     Reverse Label is present and
     gmplsTunnelARHopExplicitReverseLabelPtr 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 }

gmplsTunnelARHopExplicitReverseLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelARHopExplicitReverseLabel 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
       This flag is set to indicate that the link downstream of this
       node is protected via a local repair mechanism.

     localInUse
       This flag is set to indicate 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)."
  REFERENCE
```

```
    "1. RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209,
        section 4.4.1."
::= { gmplsTunnelARHopEntry 6 }

gmplsTunnelCHopTable  OBJECT-TYPE
  SYNTAX  SEQUENCE OF GmplsTunnelCHopEntry
  MAX-ACCESS not-accessible
  STATUS  current
  DESCRIPTION
    "The gmplsTunnelCHopTable sparsely extends the
     mplsTunnelCHopTable of MPLS-TE-STD-MIB.  It is used to indicate
     additional information about the hops of a GMPLS tunnel defined
     in the 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 in the
     mplsTunnelCHopTable.  It is acceptable for some rows in the
     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
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812.
     2. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473."
::= { 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 a path computation engine on (or on
```

```
     behalf of) the transit LSR."
  INDEX {
    mplsTunnelCHopListIndex,
    mplsTunnelCHopIndex
  }
::= { gmplsTunnelCHopTable 1 }

GmplsTunnelCHopEntry ::= SEQUENCE {
  gmplsTunnelCHopLabelStatuses           BITS,
  gmplsTunnelCHopExplicitForwardLabel    Unsigned32,
  gmplsTunnelCHopExplicitForwardLabelPtr RowPointer,
  gmplsTunnelCHopExplicitReverseLabel    Unsigned32,
  gmplsTunnelCHopExplicitReverseLabelPtr 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
     gmplsTunnelCHopExplicitForwardLabel or
     gmplsTunnelCHopExplicitForwardLabelPtr and
     gmplsTunnelCHopExplicitReverseLabel or
     gmplsTunnelCHopExplicitReverseLabelPtr 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 }

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

gmplsTunnelCHopExplicitForwardLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelCHopExplicitForwardLabel object."
::= { gmplsTunnelCHopEntry 3 }

gmplsTunnelCHopExplicitReverseLabel OBJECT-TYPE
  SYNTAX  Unsigned32
  MAX-ACCESS read-only
  STATUS  current
  DESCRIPTION
    "If the gmplsTunnelCHopLabelStatuses object indicates that a
     Reverse Label is present and
     gmplsTunnelCHopExplicitReverseLabelPtr 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 }

gmplsTunnelCHopExplicitReverseLabelPtr 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 of
     GMPLS-LABEL-STD-MIB) 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
     gmplsTunnelCHopExplicitReverseLabel object."
::= { gmplsTunnelCHopEntry 5 }

gmplsTunnelReversePerfTable  OBJECT-TYPE
  SYNTAX  SEQUENCE OF GmplsTunnelReversePerfEntry
  MAX-ACCESS not-accessible
  STATUS  current
  DESCRIPTION
```

```
    "This table augments the gmplsTunnelTable to provide
     per-tunnel packet performance information for the reverse
     direction of a bidirectional tunnel.  It can be seen as
     supplementing the mplsTunnelPerfTable, which augments the
     mplsTunnelTable.

     For links that do not transport packets, these packet counters
     cannot be maintained.  For such links, attempts to read the
     objects in this table will return noSuchInstance.

     A tunnel can be known to be bidirectional by inspecting the
     gmplsTunnelDirection object."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812."
::= { 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.

     This object represents the 32-bit value of the least
     significant part of the 64-bit value if both
     gmplsTunnelReversePerfHCPackets and this object are returned.
```

```
     For links that do not transport packets, this packet counter
     cannot be maintained.  For such links, this value will return
     noSuchInstance."
::= { 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.

     For links that do not transport packets, this packet counter
     cannot be maintained.  For such links, this value will return
     noSuchInstance."
::= { 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.  For links that do not
     transport packets, this packet counter cannot be maintained.  For
     such links, this value will return noSuchInstance."
::= { 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.

     This object represents the 32-bit value of the least
     significant part of the 64-bit value if both
     gmplsTunnelReversePerfHCBytes and this object are returned.

     For links that do not transport packets, this packet counter
     cannot be maintained.  For such links, this value will return
     noSuchInstance."
::= { 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.

     For links that do not transport packets, this packet counter
     cannot be maintained.  For such links, this value will return
     noSuchInstance."
::= { gmplsTunnelReversePerfEntry 5 }

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.

     Entries in this table are not persistent over system resets
     or re-initializations of the management system."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Traffic Engineering (TE)
        Management Information Base (MIB), RFC 3812."
::= { 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 that read the objects in this table one at
     a time and do not perform atomic operations to read entire
     instantiated table rows at once, should, for each conceptual
     column with valid data, read gmplsTunnelErrorLastTime
     prior to the other objects in the row and again subsequent to
     reading the last object of the row.  They should verify that
     the value of gmplsTunnelErrorLastTime did not change and
     thereby ensure that all data read belongs to the same error
     event."
```

```
  AUGMENTS { mplsTunnelEntry }
::= { gmplsTunnelErrorTable 1 }

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

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 (conceptual row) 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 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 that has been detected
     by the local node but that 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 not valid and should be ignored.

     Note that entries in this table are not persistent over system
     resets or re-initializations of the management system."
::= { 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 not valid and should be 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
    "1. Textual Conventions for Internet Network Addresses, RFC 4001,
        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 signaling protocol
     identified by the mplsTunnelSignallingProto object."
  REFERENCE
    "1. Resource ReserVation Protocol -- Version 1 Functional
        Specification, RFC 2205, section B.
     2. RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209,
        section 7.3.
     3. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 13.1."
::= { 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. Resource ReserVation Protocol -- Version 1 Functional
        Specification, RFC 2205, section B.
     2. RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209,
        section 7.3.
     3. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 13.1. "
::= { gmplsTunnelErrorEntry 6 }

gmplsTunnelErrorTLVs OBJECT-TYPE
  SYNTAX  OCTET STRING (SIZE(0..65535))
  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
    "1. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 8.2."
::= { gmplsTunnelErrorEntry 7 }

gmplsTunnelErrorHelpString OBJECT-TYPE
  SYNTAX  SnmpAdminString
  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 }

--
-- Notifications
--

gmplsTunnelDown NOTIFICATION-TYPE
OBJECTS  {
  mplsTunnelAdminStatus,
  mplsTunnelOperStatus,
  gmplsTunnelErrorLastErrorType,
  gmplsTunnelErrorReporterType,
  gmplsTunnelErrorReporter,
  gmplsTunnelErrorCode,
  gmplsTunnelErrorSubcode
}
STATUS      current
DESCRIPTION
     "This notification is generated when an mplsTunnelOperStatus
      object for a tunnel in the gmplsTunnelTable 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 to down(2).

      Note that an implementation MUST only issue one of
      mplsTunnelDown and gmplsTunnelDown for any single event on a
      single tunnel.  If the tunnel has an entry in the
      gmplsTunnelTable, an implementation SHOULD use gmplsTunnelDown
      for all tunnel-down events and SHOULD NOT use mplsTunnelDown.

      This notification is subject to the control of
      mplsTunnelNotificationEnable.  When that object is set
      to false(2), then the notification must not be issued.

      Further, this notification is also subject to
      mplsTunnelNotificationMaxRate.  That object indicates the
      maximum number of notifications issued per second.  If events
      occur more rapidly, the implementation may simply fail to emit
      some notifications during that period, or may queue them until
      an appropriate time.  The notification rate applies to the sum
      of all notifications in the MPLS-TE-STD-MIB and
      GMPLS-TE-STD-MIB modules applied across the whole of the
      reporting device.

      mplsTunnelOperStatus, mplsTunnelAdminStatus, mplsTunnelDown,
      mplsTunnelNotificationEnable, and mplsTunnelNotificationMaxRate
      objects are found in MPLS-TE-STD-MIB."
    REFERENCE
      "1. Multiprotocol Label Switching (MPLS) Traffic Engineering
          (TE) Management Information Base (MIB), RFC 3812."
::= { gmplsTeNotifications 1 }

gmplsTeGroups
  OBJECT IDENTIFIER ::= { gmplsTeConformance 1 }

gmplsTeCompliances
  OBJECT IDENTIFIER ::= { gmplsTeConformance 2 }

-- Compliance requirement for fully compliant implementations.

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.

      The mandatory group has to be implemented by all LSRs that
      originate, terminate, or act as transit for TE-LSPs/tunnels.
      In addition, depending on the type of tunnels supported, other
```

```
      groups become mandatory as explained below."

  MODULE MPLS-TE-STD-MIB -- The MPLS-TE-STD-MIB, RFC 3812

  MANDATORY-GROUPS {
     mplsTunnelGroup,
     mplsTunnelScalarGroup
  }

MODULE -- this module

MANDATORY-GROUPS {
  gmplsTunnelGroup,
  gmplsTunnelScalarGroup
}

GROUP gmplsTunnelSignaledGroup
  DESCRIPTION
    "This group is mandatory for devices that 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 gmplsTunnelOptionalGroup
  DESCRIPTION
    "Objects in this group are optional."

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

::= { 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 that 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 gmplsTunnelOptionalGroup
  DESCRIPTION
    "Objects in this group are optional."

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

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

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

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

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

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

```
    "Write access is not required."

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

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

OBJECT gmplsTunnelDirection
  MIN-ACCESS  read-only
  DESCRIPTION
    "Only forward(0) is required."

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

OBJECT gmplsTunnelUpstreamNotifyRecipientType
  SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
  MIN-ACCESS   read-only
  DESCRIPTION  "Only unknown(0), ipv4(1), and ipv6(2) support
                 is required."

OBJECT gmplsTunnelUpstreamNotifyRecipient
  SYNTAX      InetAddress (SIZE(0|4|16))
  MIN-ACCESS  read-only
  DESCRIPTION "An implementation is only required to support
               unknown(0), ipv4(1), and ipv6(2) sizes."

OBJECT gmplsTunnelSendResvNotifyRecipientType
  SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
  MIN-ACCESS read-only
  DESCRIPTION "Only unknown(0), ipv4(1), and ipv6(2) support
               is required."

OBJECT gmplsTunnelSendResvNotifyRecipient
  SYNTAX      InetAddress (SIZE(0|4|16))
  MIN-ACCESS read-only
  DESCRIPTION "An implementation is only required to support
               unknown(0), ipv4(1), and ipv6(2) sizes."

OBJECT gmplsTunnelDownstreamNotifyRecipientType
  SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
```

```
  MIN-ACCESS read-only
  DESCRIPTION "Only unknown(0), ipv4(1), and ipv6(2) support
               is required."

OBJECT gmplsTunnelDownstreamNotifyRecipient
  SYNTAX      InetAddress (SIZE(0|4|16))
  MIN-ACCESS read-only
  DESCRIPTION "An implementation is only required to support
               unknown(0), ipv4(1), and ipv6(2) sizes."

OBJECT gmplsTunnelSendPathNotifyRecipientType
  SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
  MIN-ACCESS read-only
  DESCRIPTION "Only unknown(0), ipv4(1), and ipv6(2) support
               is required."

OBJECT gmplsTunnelSendPathNotifyRecipient
  SYNTAX      InetAddress (SIZE(0|4|16))
  MIN-ACCESS read-only
  DESCRIPTION "An implementation is only required to support
               unknown(0), ipv4(1), and ipv6(2) sizes."

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

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

-- gmplsTunnelHopLabelStatuses has max access read-only

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

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

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

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

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

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

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

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

OBJECT gmplsTunnelErrorReporterType
  SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
  DESCRIPTION "Only unknown(0), ipv4(1), and ipv6(2) support
               is required."

OBJECT gmplsTunnelErrorReporter
  SYNTAX      InetAddress (SIZE(0|4|16))
  DESCRIPTION "An implementation is only required to support
               unknown(0), ipv4(1), and ipv6(2)."
::= { gmplsTeCompliances 2 }

gmplsTunnelGroup OBJECT-GROUP
  OBJECTS {
    gmplsTunnelDirection,
    gmplsTunnelReversePerfPackets,
    gmplsTunnelReversePerfHCPackets,
    gmplsTunnelReversePerfErrors,
    gmplsTunnelReversePerfBytes,
    gmplsTunnelReversePerfHCBytes,
    gmplsTunnelErrorLastErrorType,
    gmplsTunnelErrorLastTime,
    gmplsTunnelErrorReporterType,
    gmplsTunnelErrorReporter,
    gmplsTunnelErrorCode,
    gmplsTunnelErrorSubcode,
    gmplsTunnelErrorTLVs,
    gmplsTunnelErrorHelpString,
    gmplsTunnelUnnumIf
  }
  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
     gmplsTunnelSignaledGroup group is mandatory."
::= { gmplsTeGroups 1 }

gmplsTunnelSignaledGroup OBJECT-GROUP
  OBJECTS {
    gmplsTunnelAttributes,
    gmplsTunnelLSPEncoding,
    gmplsTunnelSwitchingType,
    gmplsTunnelLinkProtection,
    gmplsTunnelGPid,
    gmplsTunnelSecondary,
    gmplsTunnelPathComp,
    gmplsTunnelUpstreamNotifyRecipientType,
    gmplsTunnelUpstreamNotifyRecipient,
    gmplsTunnelSendResvNotifyRecipientType,
    gmplsTunnelSendResvNotifyRecipient,
    gmplsTunnelDownstreamNotifyRecipientType,
    gmplsTunnelDownstreamNotifyRecipient,
    gmplsTunnelSendPathNotifyRecipientType,
    gmplsTunnelSendPathNotifyRecipient,
    gmplsTunnelAdminStatusFlags,
    gmplsTunnelHopLabelStatuses,
    gmplsTunnelHopExplicitForwardLabel,
    gmplsTunnelHopExplicitForwardLabelPtr,
    gmplsTunnelHopExplicitReverseLabel,
    gmplsTunnelHopExplicitReverseLabelPtr
  }
  STATUS  current
  DESCRIPTION
    "Objects needed to implement signaled tunnels."
::= { gmplsTeGroups 2 }

gmplsTunnelScalarGroup OBJECT-GROUP
  OBJECTS {
    gmplsTunnelsConfigured,
    gmplsTunnelsActive
  }
  STATUS  current
  DESCRIPTION
    "Scalar objects needed to implement MPLS tunnels."
::= { gmplsTeGroups 3 }

gmplsTunnelOptionalGroup OBJECT-GROUP
  OBJECTS {
```

```
    gmplsTunnelExtraParamsPtr,
    gmplsTunnelARHopLabelStatuses,
    gmplsTunnelARHopExplicitForwardLabel,
    gmplsTunnelARHopExplicitForwardLabelPtr,
    gmplsTunnelARHopExplicitReverseLabel,
    gmplsTunnelARHopExplicitReverseLabelPtr,
    gmplsTunnelARHopProtection,
    gmplsTunnelCHopLabelStatuses,
    gmplsTunnelCHopExplicitForwardLabel,
    gmplsTunnelCHopExplicitForwardLabelPtr,
    gmplsTunnelCHopExplicitReverseLabel,
    gmplsTunnelCHopExplicitReverseLabelPtr
  }
  STATUS  current
  DESCRIPTION
    "The objects in this group are optional."
::= { gmplsTeGroups 4 }

gmplsTeNotificationGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
     gmplsTunnelDown
  }
  STATUS  current
  DESCRIPTION
    "Set of notifications implemented in this module.  None is
     mandatory."
::= { gmplsTeGroups 5 }

END
```

9. Security Considerations

It is clear that the MIB modules described in this document in association with MPLS-TE-STD-MIB [RFC3812] 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:

- o 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.

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:

- o the gmplsTunnelTable, gmplsTunnelHopTable, gmplsTunnelARHopTable, gmplsTunnelCHopTable, gmplsTunnelReversePerfTable, and 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 document is a product of the CCAMP Working Group.

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 Lars Eggert,

Baktha Muralidharan, Tom Petch, Dan Romascanu, Dave Thaler, and Bert Wijnen for their review comments.

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

Joan Cucchiara provided a helpful and very thorough MIB Doctor review.

11. IANA Considerations

IANA has rooted MIB objects in the MIB modules contained in this document according to the sections below.

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

IANA has rooted MIB objects in the GMPLS-TE-STD-MIB module contained in this document under the mplsStdMIB subtree.

IANA has made the following assignments in the "NETWORK MANAGEMENT PARAMETERS" registry located at http://www.iana.org/assignments/smi-numbers in table:

...mib-2.transmission.mplsStdMIB (1.3.6.1.2.1.10.166)

Decimal	Name	References
13	GMPLS-TE-STD-MIB	[RFC4802]

In the future, GMPLS-related standards-track MIB modules should be rooted under the mplsStdMIB (sic) subtree. IANA has been requested to manage that namespace in the SMI Numbers registry [RFC3811]. New assignments can only be made via a Standards Action as specified in [RFC2434].

11.2. Dependence on IANA MIB Modules

Three MIB objects in the GMPLS-TE-STD-MIB module defined in this document (gmplsTunnelLSPEncoding, gmplsTunnelSwitchingType, and gmplsTunnelGPid) use textual conventions imported from the IANA-GMPLS-TC-MIB module. 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 IANA-GMPLS-TC-MIB are outlined in the DESCRIPTION clause associated with its MODULE-IDENTITY statement.

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

11.2.1. IANA-GMPLS-TC-MIB Definition

This section provides the base definition of the IANA GMPLS TC MIB module. This MIB module is under the direct control of IANA. Please see the most updated version of this MIB at <http://www.iana.org/assignments/ianagmplstc-mib>.

This MIB makes reference to the following documents: [RFC2578], [RFC2579], [RFC3471], [RFC3473], [RFC4202], [RFC4328], and [RFC4783].

IANA assigned an OID to the IANA-GMPLS-TC-MIB module specified in this document as { mib-2 152 }.

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

IMPORTS
    MODULE-IDENTITY, mib-2              FROM SNMPv2-SMI  -- RFC 2578
    TEXTUAL-CONVENTION                  FROM SNMPv2-TC;  -- RFC 2579

ianaGmpls MODULE-IDENTITY
    LAST-UPDATED
                "200702270000Z" -- 27 February 2007 00:00:00 GMT
    ORGANIZATION
                "IANA"
    CONTACT-INFO
                "Internet Assigned Numbers Authority
                 Postal: 4676 Admiralty Way, Suite 330
                         Marina del Rey, CA 90292
                 Tel:    +1 310 823 9358
                 E-Mail: iana@iana.org"
    DESCRIPTION
      "Copyright (C) The IETF Trust (2007).  The initial version
       of this MIB module was published in RFC 4802.  For full legal
       notices see the RFC itself.  Supplementary information
       may be available on:
       http://www.ietf.org/copyrights/ianamib.html"

       REVISION
        "200702270000Z" -- 27 February 2007 00:00:00 GMT
      DESCRIPTION
        "Initial version issued as part of RFC 4802."
```

```
       ::= { mib-2 152 }

   IANAGmplsLSPEncodingTypeTC ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
            "This type is used to represent and control
             the LSP encoding type of an LSP signaled by a GMPLS
             signaling protocol.

             This textual convention is strongly tied to the LSP
             Encoding Types sub-registry of the GMPLS Signaling
             Parameters registry managed by IANA.  Values should be
             assigned by IANA in step with the LSP Encoding Types
             sub-registry and using the same registry management rules.
             However, the actual values used in this textual convention
             are solely within the purview of IANA and do not
             necessarily match the values in the LSP Encoding Types
             sub-registry.

             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@iana.org)."
       REFERENCE
            "1. Generalized Multi-Protocol Label Switching (GMPLS)
                Signaling Functional Description, RFC 3471, section
                3.1.1.
             2. Generalized MPLS Signalling Extensions for G.709 Optical
                Transport Networks Control, RFC 4328, section 3.1.1."
       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
              }

   IANAGmplsSwitchingTypeTC ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
            "This type is used to represent and
             control the LSP switching type of an LSP signaled by a
             GMPLS signaling protocol.

             This textual convention is strongly tied to the Switching
             Types sub-registry of the GMPLS Signaling Parameters
             registry managed by IANA.  Values should be assigned by
             IANA in step with the Switching Types sub-registry and
             using the same registry management rules.  However, the
             actual values used in this textual convention are solely
             within the purview of IANA and do not necessarily match
             the values in the Switching Types sub-registry.

             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@iana.org)."
       REFERENCE
            "1. Routing Extensions in Support of Generalized
                Multi-Protocol Label Switching, RFC 4202, section 2.4.
             2. Generalized Multi-Protocol Label Switching (GMPLS)
                Signaling Functional Description, RFC 3471, section
                3.1.1."
       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
                }
```

```
   IANAGmplsGeneralizedPidTC ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
            "This data type is used to represent and control the LSP
             Generalized Protocol Identifier (G-PID) of an LSP
             signaled by a GMPLS signaling protocol.

             This textual convention is strongly tied to the Generalized
             PIDs (G-PID) sub-registry of the GMPLS Signaling Parameters
             registry managed by IANA.  Values should be assigned by
             IANA in step with the Generalized PIDs (G-PID) sub-registry
             and using the same registry management rules.  However, the
             actual values used in this textual convention are solely
             within the purview of IANA and do not necessarily match the
             values in the Generalized PIDs (G-PID) sub-registry.

             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@iana.org)."
        REFERENCE
            "1. Generalized Multi-Protocol Label Switching (GMPLS)
                Signaling Functional Description, RFC 3471, section
                3.1.1.
             2. Generalized MPLS Signalling Extensions for G.709 Optical
                Transport Networks Control, RFC 4328, section 3.1.3."
        SYNTAX  INTEGER {
                  unknown(0),      -- unknown or none of the following
                  -- the values 1, 2, 3 and 4 are reserved in RFC 3471
                  asynchE4(5),
                  asynchDS3T3(6),
                  asynchE3(7),
                  bitsynchE3(8),
                  bytesynchE3(9),
                  asynchDS2T2(10),
                  bitsynchDS2T2(11),
                  reservedByRFC3471first(12),
                  asynchE1(13),
                  bytesynchE1(14),
                  bytesynch31ByDS0(15),
                  asynchDS1T1(16),
                  bitsynchDS1T1(17),
```

```
                bytesynchDS1T1(18),
                vc1vc12(19),
                reservedByRFC3471second(20),
                reservedByRFC3471third(21),
                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)
              }

   IANAGmplsAdminStatusInformationTC ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
            "This data type 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.

            This textual convention is strongly tied to the
            Administrative Status Information Flags sub-registry of
            the GMPLS Signaling Parameters registry managed by IANA.
            Values should be assigned by IANA in step with the
            Administrative Status Flags sub-registry and using the
            same registry management rules.  However, the actual
            values used in this textual convention are solely
            within the purview of IANA and do not necessarily match
            the values in the Administrative Status Information
            Flags sub-registry.

            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@iana.org)."
        REFERENCE
            "1. Generalized Multi-Protocol Label Switching (GMPLS)
                Signaling Functional Description, RFC 3471, section 8.
             2. Generalized MPLS Signaling - RSVP-TE Extensions,
                RFC 3473, section 7.
             3. GMPLS - Communication of Alarm Information,
                RFC 4783, section 3.2.1."
        SYNTAX BITS {
                  reflect(0), -- Reflect bit (RFC 3471)
                  reserved1(1), -- reserved
                  reserved2(2), -- reserved
                  reserved3(3), -- reserved
                  reserved4(4), -- reserved
                  reserved5(5), -- reserved
                  reserved6(6), -- reserved
                  reserved7(7), -- reserved
                  reserved8(8), -- reserved
                  reserved9(9), -- reserved
                  reserved10(10), -- reserved
                  reserved11(11), -- reserved
                  reserved12(12), -- reserved
                  reserved13(13), -- reserved
                  reserved14(14), -- reserved
                  reserved15(15), -- reserved
                  reserved16(16), -- reserved
```

```
                reserved17(17), -- reserved
                reserved18(18), -- reserved
                reserved19(19), -- reserved
                reserved20(20), -- reserved
                reserved21(21), -- reserved
                reserved22(22), -- reserved
                reserved23(23), -- reserved
                reserved24(24), -- reserved
                reserved25(25), -- reserved
                reserved26(26), -- reserved
                reserved27(27), -- Inhibit Alarm bit (RFC 4783)
                reserved28(28), -- reserved
                testing(29), -- Testing bit (RFC 3473)
                administrativelyDown(30), -- Admin down (RFC 3473)
                deleteInProgress(31) -- Delete bit (RFC 3473)
              }
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.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

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

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

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

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

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

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

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

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

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

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

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

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

[RFC3945] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

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

[RFC4202] Kompella, K. and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

[RFC4328] Papadimitriou, D., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Extensions for G.709 Optical Transport Networks Control", RFC 4328, January 2006.

[RFC4783] Berger, L., "GMPLS - Communication of Alarm Information", RFC 4783, December 2006.

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

12.2. Informative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

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

[RFC3472] Ashwood-Smith, P. and L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions", RFC 3472, January 2003.

Contact Information

Thomas D. Nadeau
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719

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

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.