Network Working Group
Request for Comment: 4803
Category: Standards Track

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

February 2007

Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR) 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 to configure and/or monitor a Generalized Multiprotocol Label Switching (GMPLS) Label Switching Router (LSR).

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 a Generalized Multiprotocol Label Switching (GMPLS) [RFC3945] Label Switching Router (LSR).

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

1.1. Migration Strategy

MPLS LSRs may be modeled and managed using the MPLS-LSR-STD-MIB module [RFC3813].

LSRs may be migrated to be modeled and managed 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 LSR MIB module (GMPLS-LSR-STD-MIB), defined in this document, extends the MPLS-LSR-STD-MIB module [RFC3813] through a series of 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-LSR-STD-MIB support to GMPLS-LSR-STD-MIB support, an implementation needs only to add support for the additional tables and objects defined in GMPLS-LSR-STD-MIB. The gmplsInterfaceSignalingCaps object allows an implementation to use the objects and tables of GMPLS-LSR-STD-MIB without supporting the GMPLS protocols.

The GMPLS Label MIB module (GMPLS-LABEL-STD-MIB), also defined in this document, allows labels to be configured and examined, and it supports more varieties of labels as appropriate for GMPLS. Labels may be referenced using a row pointer from objects within the GMPLS-LSR-STD-MIB module. MPLS implementations (MPLS-LSR-STD-MIB) may also reference labels held in the GMPLS-LABEL-STD-MIB module through the various label pointer objects in the MPLS-LSR-STD-MIB module (such as mplsInSegmentLabelPtr), and may do so without implementing the GMPLS-LSR-STD-MIB module.

The companion document modeling and managing GMPLS-based traffic engineering [RFC4802] extends the MPLS-TE-STD-MIB module [RFC3812] with the same intentions.

Textual conventions are defined in [RFC4801], which extends the set of textual conventions originally defined in [RFC3811].

2. Terminology

This document uses terminology from the document describing the MPLS architecture [RFC3031] and the GMPLS architecture [RFC3945].

A Label Switched Path (LSP) is modeled as a connection consisting of one or more incoming segments (in-segments) and/or one or more outgoing segments (out-segments) at an LSR. The association or interconnection of the in-segments and out-segments is accomplished by using a cross-connect. We use the terminology "connection" and "LSP" interchangeably where the meaning is clear from the context.

in-segment This is analogous to a GMPLS Label on an interface.

out-segment This is analogous to a GMPLS Label on an interface.

cross-connect This describes the conceptual connection between a set of in-segments and out-segments. Note that either set may be empty; for example, a cross-connect may connect only out-segments together with no in-segments in the case where an LSP originates on an LSR.

The terms 'ingress' and 'head-end' (or 'head') are used in this document to indicate the signaling source of an LSP. This is sometimes also referred to as the 'sender'.

The terms 'egress' and 'tail-end' (or 'tail') are used in this document to indicate the signaling destination of an LSP.

The term 'upstream' is used in this document to refer to the part of an LSP that is closer to the ingress than the current point of reference.

The term 'downstream' is used in this document to refer to the part of an LSP that is closer to the egress than the current point of reference.

The term 'forward' is used in this document to indicate the direction of data flow from the ingress toward the egress.

The term 'reverse' is used in this document to indicate the direction of data flow from the egress toward the ingress.

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

4.1. MIB Modules

There are two MIB modules defined in this document.

The GMPLS-LSR-STD-MIB module contains tables that sparse augment tables defined in the MPLS-LSR-STD-MIB module [RFC3813]. This MIB module is used in conjunction with the MPLS-LSR-STD-MIB module [RFC3813] in systems that support GMPLS.

The GMPLS-LABEL-STD-MIB module contains objects for managing GMPLS Labels when they cannot be represented using the textual conventions of the MPLS-TC-STD-MIB module [RFC3811], or when more detailed access to the sub-fields of the labels is required.

4.1.1. Summary of the GMPLS-LSR-STD-MIB Module

The MIB tables in the GMPLS-LSR-STD-MIB module are as follows:

- The interface configuration table (gmplsInterfaceTable) sparse augments the mplsInterfaceTable [RFC3813] to enable the GMPLS protocol on MPLS-capable interfaces.

- The in-segment (gmplsInSegmentTable) and out-segment (gmplsOutSegmentTable) tables sparse augment mplsInSegmentTable and mplsOutSegmentTable [RFC3813] to enable configuration of GMPLS-specific parameters for LSP segments at an LSR.

These tables are described in the subsequent sections.

4.1.2. Summary of the GMPLS-LABEL-STD-MIB Module

There is one MIB table in the GMPLS-LABEL-STD-MIB module as follows:

- The gmplsLabelTable allows Generalized Labels to be defined and managed in a central location. Generalized Labels can be of variable length and have distinct bit-by-bit interpretations depending upon how they are defined for the specific technology in which they are used. For example, labels used for MPLS packet switching are different in length and content from labels used in Time Division Multiplexer (TDM) timeslot switching.

4.2. Configuring Statically Provisioned LSPs

Configuring statically provisioned GMPLS LSPs through an LSR involves the following steps:

- Configuring an interface using the MPLS-LSR-STD-MIB module [RFC3813].

- Enabling GMPLS on GMPLS-capable interfaces using the GMPLS-LSR-STD-MIB module in this document.

- Configuring in-segments and out-segments using the MPLS-LSR-STD-MIB module [RFC3813].

- Configuring GMPLS extensions to the in-segments and out-segments using the GMPLS-LSR-STD-MIB module in this document.

- Setting up the cross-connect table in the MPLS-LSR-STD-MIB module [RFC3813] to associate segments and/or to indicate connection origination and termination.

- Optionally setting up labels in the label table in the GMPLS-LABEL-STD-MIB module in this document if the textual convention MplsLabel [RFC3811] is not capable of holding the required label (for example, if the label requires more than 32 bits to encode it), or if the operator wishes to disambiguate GMPLS Label types.

- Optionally specifying label stack actions in the MPLS-LSR-STD-MIB module [RFC3813].

- Optionally specifying segment traffic parameters in the MPLS-LSR-STD-MIB module [RFC3813].

5. Bidirectional LSPs

The GMPLS-LSR-STD-MIB module supports bidirectional LSPs as required for GMPLS. A single value of mplsXCIndex is shared by all of the segments for the entire bidirectional LSP. This facilitates a simple reference from [RFC3812] and [RFC4802] and makes fate-sharing more obvious.

It is, however, important that the direction of segments is understood to avoid connecting all in-segments to all out-segments. This is achieved by an object in each segment that indicates the direction of the segment with respect to data flow.

A segment that is marked as 'forward' carries data from the 'head' of the LSP to the 'tail'. A segment marked as 'reverse' carries data in the reverse direction.

Where an LSP is signaled using a conventional signaling protocol, the 'head' of the LSP is the source of the signaling (also known as the ingress) and the 'tail' is the destination (also known as the egress). For manually configured LSPs, an arbitrary decision must be made about which segments are 'forward' and which 'reverse'. For consistency, this decision should be made across all LSRs that participate in the LSP by assigning 'head' and 'tail' ends to the LSP.

6. Example of LSP Setup

In this section, we provide a brief example of using the MIB objects described in sections 7 and 8 to set up an LSP. While this example is not meant to illustrate every nuance of the MIB modules, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB modules themselves. A prerequisite is an understanding of the MPLS-LSR-STD-MIB module [RFC3813].

Suppose that one would like to manually create a best-effort, bidirectional LSP. Assume that, in the forward direction, the LSP enters the LSR via MPLS interface A with ifIndex 12 and exits the LSR via MPLS interface B with ifIndex 13. For the reverse direction, we assume that the LSP enters via interface B and leaves via interface A (i.e., the forward and reverse directions use the same bidirectional interfaces). Let us also assume that we do not wish to have a label stack beneath the top label on the outgoing labeled packets. The following example illustrates which rows and corresponding objects might be created to accomplish this.

We must first create rows in the gmplsLabelTable corresponding to the labels required for each of the forward- and reverse-direction in- and out-segments. For the purpose of this example, the forward and reverse labels on each interface will be the same, hence we need to create just two rows in the gmplsLabelTable - one for each interface.

```
In gmplsLabelTable:
{
  gmplsLabelInterface          = 12,
  gmplsLabelIndex              = 1,
  gmplsLabelSubindex           = 0,
  gmplsLabelType               = gmplsFreeformLabel(3),
  gmplsLabelFreeform           = 0x123456789ABCDEF0
  gmplsLabelRowStatus          = createAndGo(4)
}
```

```
In gmplsLabelTable:
{
  gmplsLabelInterface          = 13,
  gmplsLabelIndex              = 1,
  gmplsLabelSubindex           = 0,
  gmplsLabelType               = gmplsFreeformLabel(3),
  gmplsLabelFreeform           = 0xFEDCBA9876543210
  gmplsLabelRowStatus          = createAndGo(4)
}
```

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the mplsInSegmentTable and mplsOutSegmentTable. Note that we use a row pointer to the two rows in the gmplsLabelTable rather than specify the labels explicitly in the in- and out-segment tables. Also note that the row status for each row is set to createAndWait(5) to allow corresponding entries in the gmplsInSegmentTable and gmplsOutSegmentTable to be created.

For the forward direction.

```
In mplsInSegmentTable:
{
   mplsInSegmentIndex          = 0x00000015
   mplsInSegmentLabel          = 0, -- incoming label in label table
   mplsInSegmentNPop           = 1,
   mplsInSegmentInterface      = 12, -- incoming interface

   -- RowPointer MUST point to the first accessible column.
   mplsInSegmentTrafficParamPtr    = 0.0,
   mplsInSegmentLabelPtr           = gmplsLabelTable(12,1,0)
   mplsInSegmentRowStatus          = createAndWait(5)
}

In mplsOutSegmentTable:
{
   mplsOutSegmentIndex         = 0x00000012,
   mplsOutSegmentInterface     = 13, -- outgoing interface
   mplsOutSegmentPushTopLabel  = true(1),
   mplsOutSegmentTopLabel      = 0, -- outgoing label in label table

   -- RowPointer MUST point to the first accessible column.
   mplsOutSegmentTrafficParamPtr   = 0.0,
   mplsOutSegmentLabelPtr          = gmplsLabelTable(13,1,0)
   mplsOutSegmentRowStatus         = createAndWait(5)
}
```

```
   For the reverse direction.

   In mplsInSegmentTable:
   {
      mplsInSegmentIndex           = 0x00000016
      mplsInSegmentLabel           = 0, -- incoming label in label table
      mplsInSegmentNPop            = 1,
      mplsInSegmentInterface       = 13, -- incoming interface

      -- RowPointer MUST point to the first accessible column.
      mplsInSegmentTrafficParamPtr    = 0.0,
      mplsInSegmentLabelPtr           = gmplsLabelTable(13,1,0)

      mplsInSegmentRowStatus          = createAndWait(5)
   }

   In mplsOutSegmentTable:
   {
      mplsOutSegmentIndex          = 0x00000013,
      mplsOutSegmentInterface      = 12, -- outgoing interface
      mplsOutSegmentPushTopLabel   = true(1),
      mplsOutSegmentTopLabel       = 0, -- outgoing label in label table

      -- RowPointer MUST point to the first accessible column.
      mplsOutSegmentTrafficParamPtr   = 0.0,
      mplsOutSegmentLabelPtr          = gmplsLabelTable(12,1,0)
      mplsOutSegmentRowStatus         = createAndWait(5)
   }

   These table entries are extended by entries in the
   gmplsInSegmentTable and gmplsOutSegmentTable.  Note that the nature
   of the 'extends' relationship is a sparse augmentation so that the
   entry in the gmplsInSegmentTable has the same index values as the
   entry in the mplsInSegmentTable.  Similarly, the entry in the
   gmplsOutSegmentTable has the same index values as the entry in the
   mplsOutSegmentTable.

   First for the forward direction:

   In gmplsInSegmentTable(0x00000015)
   {
     gmplsInSegmentDirection          = forward(1)
   }

   In gmplsOutSegmentTable(0x00000012)
   {
     gmplsOutSegmentDirection         = forward(1)
   }
```

Next for the reverse direction:

```
In gmplsInSegmentTable(0x00000016)
{
  gmplsInSegmentDirection          = reverse(2)
}

In gmplsOutSegmentTable(0x00000013)
{
  gmplsOutSegmentDirection         = reverse(2)
}
```

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

```
In mplsXCTable:
{
   mplsXCIndex                = 0x01,
   mplsXCInSegmentIndex       = 0x00000015,
   mplsXCOutSegmentIndex      = 0x00000012,
   mplsXCLspId                = 0x0102 -- unique ID
   mplsXCLabelStackIndex      = 0x00, -- only a single outgoing label
   mplsXCRowStatus            = createAndGo(4)
}

In mplsXCTable:
{
   mplsXCIndex                = 0x02,
   mplsXCInSegmentIndex       = 0x00000016,
   mplsXCOutSegmentIndex      = 0x00000013,
   mplsXCLspId                = 0x0102 -- unique ID
   mplsXCLabelStackIndex      = 0x00, -- only a single outgoing label
   mplsXCRowStatus            = createAndGo(4)
}
```

Finally, the in-segments and out-segments are activated.

```
In mplsInSegmentTable(0x00000015):
{
   mplsInSegmentRowStatus          = active(1)
}
In mplsInSegmentTable(0x00000016):
{
   mplsInSegmentRowStatus          = active(1)
}
```

```
   In mplsOutSegmentTable(0x00000012):
   {
      mplsOutSegmentRowStatus         = active(1)
   }

   In mplsOutSegmentTable(0x00000013):
   {
      mplsOutSegmentRowStatus         = active(1)
   }
```

7. GMPLS Label Switching Router MIB Definitions

This MIB module makes reference to the following documents: [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3209], [RFC3443], [RFC3468], [RFC3472], [RFC3473], [RFC3811], [RFC3813], and [RFC4801].

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

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, zeroDotZero
    FROM SNMPv2-SMI                                 -- RFC 2578
  MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF                                -- RFC 2580
  RowPointer
    FROM SNMPv2-TC                                  -- RFC 2579
  GmplsSegmentDirectionTC
    FROM GMPLS-TC-STD-MIB                           -- RFC 4801
  mplsInterfaceIndex, mplsInSegmentIndex, mplsOutSegmentIndex,
  mplsInterfaceGroup, mplsInSegmentGroup, mplsOutSegmentGroup,
  mplsXCGroup, mplsPerfGroup, mplsLsrNotificationGroup
    FROM MPLS-LSR-STD-MIB                           -- RFC 3813
  ifGeneralInformationGroup, ifCounterDiscontinuityGroup
    FROM IF-MIB                                     -- RFC 2863
  mplsStdMIB
    FROM MPLS-TC-STD-MIB                            -- RFC 3811
;

gmplsLsrStdMIB 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 4803; see the RFC itself for
     full legal notices.

     This MIB module contains managed object definitions for the
     Generalized Multiprotocol (GMPLS) Label Switching Router as
     defined in Generalized Multi-Protocol Label Switching (GMPLS)
     Architecture, Mannie et al., RFC 3945, October 2004."
  REVISION
    "200702270000Z" -- 27 February 2007 00:00:00 GMT
  DESCRIPTION
    "Initial version issued as part of RFC 4803."
  ::= { mplsStdMIB 15 }

-- no notifications are currently defined.
gmplsLsrObjects       OBJECT IDENTIFIER ::= { gmplsLsrStdMIB 1 }
gmplsLsrConformance   OBJECT IDENTIFIER ::= { gmplsLsrStdMIB 2 }

gmplsInterfaceTable OBJECT-TYPE
  SYNTAX        SEQUENCE OF GmplsInterfaceEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "This table specifies per-interface GMPLS capability and
     associated information.  It extends the information in the
     mplsInterfaceTable of MPLS-LSR-STD-MIB through a
     sparse augmentation relationship."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  ::= { gmplsLsrObjects 1 }

gmplsInterfaceEntry OBJECT-TYPE
  SYNTAX        GmplsInterfaceEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "A conceptual row in this table is created automatically by an
     LSR for each interface that is both capable of supporting
     GMPLS and configured to support GMPLS.  Note that
     support of GMPLS is not limited to control plane signaling,
     but may include data-plane-only function configured through
     SNMP SET commands performed on this MIB module.
```

```
     A conceptual row in this table may also be created via SNMP
     SET commands or automatically by the LSR to supplement a
     conceptual row in the mplsInterfaceTable where the interface
     is not capable of GMPLS but where the other objects carried
     in this row provide useful additional information for an
     MPLS interface.

     A conceptual row in this table will exist if and only if a
     corresponding entry in the mplsInterfaceTable exists, and a
     corresponding entry in the ifTable exists with ifType = mpls(166).
     If the associated entry in the ifTable is operationally disabled
     (thus removing the GMPLS capabilities on the interface) or the
     entry in the mplsInterfaceTable is deleted, the corresponding entry
     in this table MUST be deleted shortly thereafter.

     The indexes are the same as for the mplsInterfaceTable.  Thus, the
     entry with index 0 represents the per-platform label space and
     contains parameters that apply to all interfaces that
     participate in the per-platform label space."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  INDEX { mplsInterfaceIndex }
::= { gmplsInterfaceTable 1 }

GmplsInterfaceEntry ::= SEQUENCE {
  gmplsInterfaceSignalingCaps      BITS,
  gmplsInterfaceRsvpHelloPeriod    Unsigned32
}

gmplsInterfaceSignalingCaps OBJECT-TYPE
  SYNTAX  BITS {
    unknown(0),
    rsvpGmpls(1),
    crldpGmpls(2), -- note the use of CR-LDP is deprecated
    otherGmpls(3)
  }
  MAX-ACCESS   read-create
  STATUS       current
  DESCRIPTION
    "Defines the signaling capabilities on this interface.  Multiple
     bits may legitimately be set at once, but if 'unknown' is set
     then no other bit may be set.  Setting no bits implies that GMPLS
     signaling cannot be performed on this interface and all LSPs
     must be manually provisioned or that this table entry is only
     present to supplement an entry in the mplsInterfaceTable by
     providing the information carried in other objects in this row."
  REFERENCE
```

```
    "1. Generalized MPLS Signaling - CR-LDP Extensions, RFC 3472.
     2. The Multiprotocol Label Switching (MPLS) Working Group
        decision on MPLS signaling protocols, RFC 3468.
     3. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473."
  DEFVAL { { rsvpGmpls } }
::= { gmplsInterfaceEntry 1 }

gmplsInterfaceRsvpHelloPeriod OBJECT-TYPE
  SYNTAX       Unsigned32
  UNITS        "milliseconds"
  MAX-ACCESS   read-create
  STATUS       current
  DESCRIPTION
    "Period, in milliseconds, between sending Resource Reservation
     Protocol (RSVP) Hello messages on this interface.  A value of 0
     indicates that no Hello messages should be sent on this
     interface.

     This object is only valid if gmplsInterfaceSignalingCaps has no
     bits set or includes the rsvpGmpls bit."
  REFERENCE
    "1. RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209,
        section 5.
     2. Generalized MPLS Signaling - RSVP-TE Extensions, RFC 3473,
        section 9.3."
  DEFVAL { 3000 }
::= { gmplsInterfaceEntry 2 }

gmplsInSegmentTable  OBJECT-TYPE
  SYNTAX        SEQUENCE OF GmplsInSegmentEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "This table sparse augments the mplsInSegmentTable of
     MPLS-LSR-STD-MIB to provide GMPLS-specific information about
     incoming segments to an LSR."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
::= { gmplsLsrObjects 2 }

gmplsInSegmentEntry  OBJECT-TYPE
  SYNTAX        GmplsInSegmentEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "An entry in this table extends the representation of an incoming
     segment represented by an entry in the mplsInSegmentTable in
```

```
     MPLS-LSR-STD-MIB through a sparse augmentation.  An entry can be
     created by a network administrator via SNMP SET commands, or in
     response to signaling protocol events.

     Note that the storage type for this entry is given by the value
     of mplsInSegmentStorageType in the corresponding entry of the
     mplsInSegmentTable."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  INDEX { mplsInSegmentIndex }
::= { gmplsInSegmentTable 1 }

GmplsInSegmentEntry ::= SEQUENCE {
  gmplsInSegmentDirection        GmplsSegmentDirectionTC,
  gmplsInSegmentExtraParamsPtr   RowPointer
}

gmplsInSegmentDirection OBJECT-TYPE
  SYNTAX        GmplsSegmentDirectionTC
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "This object indicates the direction of data flow on this
     segment.  This object cannot be modified if
     mplsInSegmentRowStatus for the corresponding entry in the
     mplsInSegmentTable is active(1)."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  DEFVAL        { forward }
::= { gmplsInSegmentEntry 1 }

gmplsInSegmentExtraParamsPtr  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 can be
     supplied from an external table and referenced from here.  A value
     of zeroDotZero in this attribute indicates that there is no such
     additional information."
  DEFVAL      { zeroDotZero }
  ::= { gmplsInSegmentEntry 2 }

gmplsOutSegmentTable  OBJECT-TYPE
```

```
  SYNTAX        SEQUENCE OF GmplsOutSegmentEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "This table sparse augments the mplsOutSegmentTable of
     MPLS-LSR-STD-MIB to provide GMPLS-specific information about
     outgoing segments from an LSR."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
::= { gmplsLsrObjects 3 }

gmplsOutSegmentEntry  OBJECT-TYPE
  SYNTAX        GmplsOutSegmentEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "An entry in this table extends the representation of an outgoing
     segment represented by an entry in the mplsOutSegmentTable of
     MPLS-LSR-STD-MIB through a sparse augmentation.  An entry can be
     created by a network administrator via SNMP SET commands, or in
     response to signaling protocol events.

     Note that the storage type for this entry is given by the value
     of mplsOutSegmentStorageType in the corresponding entry of the
     mplsOutSegmentTable."
  REFERENCE
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  INDEX  { mplsOutSegmentIndex }
::= { gmplsOutSegmentTable 1 }

GmplsOutSegmentEntry ::= SEQUENCE {
  gmplsOutSegmentDirection       GmplsSegmentDirectionTC,
  gmplsOutSegmentTTLDecrement    Unsigned32,
  gmplsOutSegmentExtraParamsPtr  RowPointer
}

gmplsOutSegmentDirection OBJECT-TYPE
  SYNTAX        GmplsSegmentDirectionTC
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "This object indicates the direction of data flow on this
     segment.  This object cannot be modified if
     mplsOutSegmentRowStatus for the corresponding entry in the
     mplsOutSegmentTable is active(1)."
  REFERENCE
```

```
    "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
  DEFVAL  { forward }
::= { gmplsOutSegmentEntry 1 }

gmplsOutSegmentTTLDecrement OBJECT-TYPE
  SYNTAX        Unsigned32
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "This object indicates the amount by which to decrement the Time
     to Live (TTL) of any payload packets forwarded on this segment if
     per-hop decrementing is being done.

     A value of zero indicates that no decrement should be made or
     that per-hop decrementing is not in use.

     See the gmplsTunnelTTLDecrement object in the gmplsTunnelTable
     of GMPLS-TE-STD-MIB for a value by which to decrement the TTL
     for the whole of a tunnel.

     This object cannot be modified if mplsOutSegmentRowStatus for
     the associated entry in the mplsOutSegmentTable is active(1)."
  REFERENCE
    "1. Time To Live (TTL) Processing in Multi-Protocol Label
        Switching (MPLS) Networks, RFC 3443.
     2. Generalized Multiprotocol Label Switching (GMPLS) Traffic
        Engineering Management Information Base, RFC 4802."
  DEFVAL  { 0 }
::= { gmplsOutSegmentEntry 2 }

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

     A value of zeroDotZero in this attribute indicates that there is
     no such additional information."
  DEFVAL      { zeroDotZero }
  ::= { gmplsOutSegmentEntry 3 }

gmplsLsrGroups
  OBJECT IDENTIFIER ::= { gmplsLsrConformance 1 }
```

```
gmplsLsrCompliances
  OBJECT IDENTIFIER ::= { gmplsLsrConformance 2 }

-- Compliance requirement for fully compliant implementations.

gmplsLsrModuleFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that provide full support for
     GMPLS-LSR-STD-MIB.

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

  MODULE IF-MIB -- The Interfaces Group MIB, RFC 2863.

  MANDATORY-GROUPS {
    ifGeneralInformationGroup,
    ifCounterDiscontinuityGroup
  }

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

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

  MODULE -- this module

  MANDATORY-GROUPS    {
    gmplsInterfaceGroup,
    gmplsInSegmentGroup,
    gmplsOutSegmentGroup
  }

  OBJECT      gmplsInSegmentDirection
  SYNTAX      GmplsSegmentDirectionTC
  MIN-ACCESS  read-only
  DESCRIPTION
    "The only valid value for unidirectional LSPs is forward(1)."
```

```
  OBJECT      gmplsOutSegmentDirection
  SYNTAX      GmplsSegmentDirectionTC
  MIN-ACCESS  read-only
  DESCRIPTION
    "The only valid value for unidirectional LSPs is forward(1)."

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

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

::= { gmplsLsrCompliances 1 }

-- Compliance requirement for implementations that provide read-only
-- access.

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

  MODULE IF-MIB -- The interfaces Group MIB, RFC 2863

  MANDATORY-GROUPS {
    ifGeneralInformationGroup,
    ifCounterDiscontinuityGroup
  }

  MODULE MPLS-LSR-STD-MIB

  MANDATORY-GROUPS {
    mplsInterfaceGroup,
    mplsInSegmentGroup,
    mplsOutSegmentGroup,
    mplsXCGroup,
    mplsPerfGroup
  }
```

```
  MODULE -- this module

  MANDATORY-GROUPS {
    gmplsInterfaceGroup,
    gmplsInSegmentGroup,
    gmplsOutSegmentGroup
  }

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

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

  OBJECT      gmplsInSegmentDirection
  SYNTAX      GmplsSegmentDirectionTC
  MIN-ACCESS  read-only
  DESCRIPTION
    "The only valid value for unidirectional LSPs is forward(1)."

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

  OBJECT      gmplsOutSegmentDirection
  MIN-ACCESS  read-only
  DESCRIPTION
    "The only valid value for unidirectional LSPs is forward(1)."

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

::= { gmplsLsrCompliances 2 }

gmplsInterfaceGroup OBJECT-GROUP
  OBJECTS {
    gmplsInterfaceSignalingCaps,
```

```
    gmplsInterfaceRsvpHelloPeriod
  }
  STATUS  current
  DESCRIPTION
   "Collection of objects that provide additional
    information for an MPLS interface and are needed
    for GMPLS interface configuration and performance
    information."
::= { gmplsLsrGroups 1 }

gmplsInSegmentGroup  OBJECT-GROUP
  OBJECTS {
    gmplsInSegmentDirection,
    gmplsInSegmentExtraParamsPtr
  }
  STATUS  current
  DESCRIPTION
    "Collection of objects that provide additional
     information for an MPLS in-segment and are needed
     for GMPLS in-segment configuration and performance
     information."
::= { gmplsLsrGroups 2 }

gmplsOutSegmentGroup  OBJECT-GROUP
  OBJECTS {
    gmplsOutSegmentDirection,
    gmplsOutSegmentTTLDecrement,
    gmplsOutSegmentExtraParamsPtr
  }
  STATUS  current
  DESCRIPTION
    "Collection of objects that provide additional
     information for an MPLS out-segment and are needed
     for GMPLS out-segment configuration and performance
     information."
::= { gmplsLsrGroups 3 }
END
```

8. GMPLS Label MIB Definitions

This MIB module makes reference to the following documents: [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3032], [RFC3289], [RFC3471], [RFC3811], and [RFC4801].

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

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, Integer32
    FROM SNMPv2-SMI                                 -- RFC 2578
  MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF                                -- RFC 2580
  RowStatus, StorageType
    FROM SNMPv2-TC                                  -- RFC 2579
  InterfaceIndexOrZero
    FROM IF-MIB                                     -- RFC 2863
  IndexIntegerNextFree
    FROM DIFFSERV-MIB                               -- RFC 3289
  MplsLabel, mplsStdMIB
    FROM MPLS-TC-STD-MIB                            -- RFC 3811
  GmplsLabelTypeTC, GmplsFreeformLabelTC
    FROM GMPLS-TC-STD-MIB                           -- RFC 4801
;

gmplsLabelStdMIB 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 4803; see the RFC itself for
     full legal notices.
```

```
     This MIB module contains managed object definitions for labels
     within GMPLS systems as defined in
     Generalized Multi-Protocol Label Switching (GMPLS) Signaling
     Functional Description, Berger, L. (Editor), RFC 3471,
     January 2003."
  REVISION
    "200702270000Z" -- 27 February 2007 00:00:00 GMT
  DESCRIPTION
    "Initial version issued as part of RFC 4803."
  ::= { mplsStdMIB 16 }

-- no notifications are currently defined.

gmplsLabelObjects        OBJECT IDENTIFIER ::= { gmplsLabelStdMIB 1 }
gmplsLabelConformance    OBJECT IDENTIFIER ::= { gmplsLabelStdMIB 2 }

gmplsLabelIndexNext OBJECT-TYPE
  SYNTAX        IndexIntegerNextFree
  MAX-ACCESS    read-only
  STATUS        current
  DESCRIPTION
      "This object contains an unused value for gmplsLabelIndex,
       or a zero to indicate that no unused value exists or is
       available.

       A management application wishing to create a row in the
       gmplsLabelTable may read this object and then attempt to
       create a row in the table.  If row creation fails (because
       another application has already created a row with the
       supplied index), the management application should read this
       object again to get a new index value.

       When a row is created in the gmplsLabelTable with the
       gmplsLabelIndex value held by this object, an implementation
       MUST change the value in this object."
  ::= { gmplsLabelObjects 1 }

gmplsLabelTable OBJECT-TYPE
  SYNTAX        SEQUENCE OF GmplsLabelEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "Table of GMPLS Labels.  This table allows the representation
     of the more complex label forms required for GMPLS that cannot
     be held within the TEXTUAL-CONVENTION MplsLabel; that is, labels
     that cannot be encoded within 32 bits.  It is, nevertheless, also
     capable of holding 32-bit labels or regular MPLS Labels if
     desired.
```

```
     Each entry in this table represents an individual GMPLS Label
     value.  The representation of Labels in tables in other MIB
     modules may be achieved by a referrence to an entry in this
     table by means of a row pointer into this table.  The indexing
     of this table provides for arbitrary indexing and also for
     concatenation of labels.

     For an example of label concatenation, see RFC 3945, section 7.1.
     In essence, a GMPLS Label may be composite in order to identify
     a set of resources in the data plane.  Practical examples are
     timeslots and wavelength sets (which are not contiguous like
     wavebands).

     The indexing mechanism allows multiple entries in this table to
     be seen as a sequence of labels that should be concatenated.
     Ordering is potentially very sensitive for concatenation."
   REFERENCE
     "1. Generalized Multiprotocol Label Switching (GMPLS)
         Architecture, RFC 3945, section 7.1."
::= { gmplsLabelObjects 2 }

gmplsLabelEntry OBJECT-TYPE
  SYNTAX        GmplsLabelEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "An entry in this table represents a single label value.  There
     are three indexes into the table.

     -  The interface index may be helpful to distinguish which
        labels are in use on which interfaces or to handle cases
        where there are a very large number of labels in use in the
        system.  When label representation is desired to apply to the
        whole system or when it is not important to distinguish
        labels by their interfaces, this index MAY be set to zero.

     -  The label index provides a way of identifying the label.

     -  The label sub-index is only used for concatenated labels.  It
        identifies each component label.  When non-concatenated labels
        are used, this index SHOULD be set to zero.

     A storage type object is supplied to control the storage type
     for each entry, but implementations should note that the storage
     type of conceptual rows in other tables that include row
     pointers to an entry in this table SHOULD dictate the storage
     type of the rows in this table where the row in the other table
     is more persistent."
```

```
  INDEX {
    gmplsLabelInterface,
    gmplsLabelIndex,
    gmplsLabelSubindex }
::= { gmplsLabelTable 1 }

GmplsLabelEntry ::= SEQUENCE {
  gmplsLabelInterface           InterfaceIndexOrZero,
  gmplsLabelIndex               Unsigned32,
  gmplsLabelSubindex            Unsigned32,
  gmplsLabelType                GmplsLabelTypeTC,
  gmplsLabelMplsLabel           MplsLabel,
  gmplsLabelPortWavelength      Unsigned32,
  gmplsLabelFreeform            GmplsFreeformLabelTC,
  gmplsLabelSonetSdhSignalIndex Integer32,
  gmplsLabelSdhVc               Integer32,
  gmplsLabelSdhVcBranch         Integer32,
  gmplsLabelSonetSdhBranch      Integer32,
  gmplsLabelSonetSdhGroupBranch Integer32,
  gmplsLabelWavebandId          Unsigned32,
  gmplsLabelWavebandStart       Unsigned32,
  gmplsLabelWavebandEnd         Unsigned32,
  gmplsLabelStorageType         StorageType,
  gmplsLabelRowStatus           RowStatus
}

gmplsLabelInterface OBJECT-TYPE
  SYNTAX        InterfaceIndexOrZero
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "The interface on which this label is used.  If this object is set
     to zero, the label MUST have applicability across the
     whole system and not be limited to a single interface."
::= { gmplsLabelEntry 1 }

gmplsLabelIndex OBJECT-TYPE
  SYNTAX        Unsigned32 (0..4294967295)
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "An arbitrary index into the table to identify a label.

     Note that implementations that are representing 32-bit labels
     within this table MAY choose to align this index with the value
     of the label, and this may result in the use of the value zero
     since it represents a valid label value.  Such implementation
     should be aware of the implications of sparsely populated
```

```
     tables.

     A management application may read the gmplsLabelIndexNext
     object to find a suitable value for this object."
::= { gmplsLabelEntry 2 }

gmplsLabelSubindex OBJECT-TYPE
  SYNTAX        Unsigned32 (0..4294967295)
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "In conjunction with gmplsLabelInterface and gmplsLabelIndex,
     this object uniquely identifies this row.  This sub-index allows
     a single GMPLS Label to be defined as a concatenation of labels.
     This is particularly useful in TDM.

     The ordering of sub-labels is strict with the sub-label with
     the lowest gmplsLabelSubindex appearing first.  Note that all
     sub-labels of a single GMPLS Label must share the same
     gmplsLabelInterface and gmplsLabelIndex values.  For labels that
     are not composed of concatenated sub-labels, this value SHOULD
     be set to zero."
::= { gmplsLabelEntry 3 }

gmplsLabelType OBJECT-TYPE
  SYNTAX        GmplsLabelTypeTC
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "Identifies the type of this label.  Note that this object does
     not determine whether MPLS or GMPLS signaling is in use: a value
     of gmplsMplsLabel(1) denotes that an MPLS Packet Label is
     present in the gmplsLabelMplsLabel object and encoded using the
     MplsLabel TEXTUAL-CONVENTION (may be a 20-bit MPLS Label, a 10-
     or 23-bit Frame Relay Label, or an Asynchronous Transfer Mode
     (ATM) Label), but does not describe whether this is signaled
     using MPLS or GMPLS.

     The value of this object helps determine which of the following
     objects are valid.  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
   REFERENCE
     "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
         Functional Description, RFC 3471, section 3."
::= { gmplsLabelEntry 4 }

gmplsLabelMplsLabel OBJECT-TYPE
  SYNTAX         MplsLabel
```

```
  MAX-ACCESS     read-create
  STATUS         current
  DESCRIPTION
    "The value of an MPLS Label (that is a Packet Label) if this
     table is used to store it.  This may be used in MPLS systems even
     though the label values can be adequately stored in the MPLS MIB
     modules (MPLS-LSR-STD-MIB and MPLS-TE-STD-MIB).  Furthermore, in
     mixed MPLS and GMPLS systems, it may be advantageous to store all
     labels in a single label table.  Lastly, in GMPLS systems where
     Packet Labels are used (that is in systems that use GMPLS
     signaling and GMPLS Labels for packet switching), it may be
     desirable to use this table.

     This object is only valid if gmplsLabelType is set
     to gmplsMplsLabel(1).  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. MPLS Label Stack Encoding, RFC 3032."
  DEFVAL { 0 }
::= { gmplsLabelEntry 5 }

gmplsLabelPortWavelength OBJECT-TYPE
  SYNTAX        Unsigned32
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The value of a Port or Wavelength Label when carried as a
     Generalized Label.  Only valid if gmplsLabelType is set to
     gmplsPortWavelengthLabel(2).  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 3.2.1.1."
  DEFVAL { 0 }
::= { gmplsLabelEntry 6 }

gmplsLabelFreeform OBJECT-TYPE
  SYNTAX        GmplsFreeformLabelTC
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The value of a Freeform Generalized Label that does not conform
     to one of the standardized label encodings or that an
     implementation chooses to represent as an octet string without
     further decoding.  Only valid if gmplsLabelType is set to
     gmplsFreeformLabel(3).  This object cannot be modified
     if gmplsLabelRowStatus is active(1)."
  REFERENCE
```

```
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 3.2."
  DEFVAL { '00'h }
::= { gmplsLabelEntry 7 }

gmplsLabelSonetSdhSignalIndex OBJECT-TYPE
  SYNTAX        Integer32 (0..4095)
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The Signal Index value (S) of a SONET or SDH Generalized Label.
     Zero indicates that this field is non-significant.  Only valid if
     gmplsLabelType is set to gmplsSonetLabel(4) or gmplsSdhLabel(5).
     This object cannot be modified if gmplsLabelRowStatus is
     active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Extensions
        for Synchronous Optical Network (SONET) and Synchronous
        Digital Hierarchy (SDH) Control, RFC 4606, section 3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 8 }

gmplsLabelSdhVc OBJECT-TYPE
  SYNTAX        Integer32 (0..15)
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The VC Indicator (U) of an SDH Generalized Label.  Zero indicates
     that this field is non-significant.  Only valid if gmplsLabelType
     is set to gmplsSdhLabel(5).  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Extensions
        for Synchronous Optical Network (SONET) and Synchronous
        Digital Hierarchy (SDH) Control, RFC 4606, section 3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 9 }

gmplsLabelSdhVcBranch OBJECT-TYPE
  SYNTAX        Integer32 (0..15)
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The VC Branch Indicator (K) of an SDH Generalized Label.  Zero
     indicates that this field is non-significant.  Only valid if
     gmplsLabelType is set to gmplsSdhLabel(5).  This
     object cannot be modified if gmplsLabelRowStatus is active(1)."
  REFERENCE
```

```
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Extensions
        for Synchronous Optical Network (SONET) and Synchronous
        Digital Hierarchy (SDH) Control, RFC 4606, section 3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 10 }

gmplsLabelSonetSdhBranch OBJECT-TYPE
  SYNTAX        Integer32 (0..15)
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The Branch Indicator (L) of a SONET or SDH Generalized Label.
     Zero indicates that this field is non-significant.  Only valid
     gmplsLabelType is set to gmplsSonetLabel(4) or
     gmplsSdhLabel(5).  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Extensions
        for Synchronous Optical Network (SONET) and Synchronous
        Digital Hierarchy (SDH) Control, RFC 4606, section 3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 11 }

gmplsLabelSonetSdhGroupBranch OBJECT-TYPE
  SYNTAX        Integer32 (0..15)
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The Group Branch Indicator (M) of a SONET or SDH Generalized
     Label.  Zero indicates that this field is non-significant.
     Only valid if gmplsLabelType is set to gmplsSonetLabel(4) or
     gmplsSdhLabel(5).  This object cannot be modified if
     gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Extensions
        for Synchronous Optical Network (SONET) and Synchronous
        Digital Hierarchy (SDH) Control, RFC 4606, section 3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 12 }

gmplsLabelWavebandId OBJECT-TYPE
  SYNTAX        Unsigned32
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The waveband identifier component of a Waveband Label.  Only
     valid if gmplsLabelType is set to gmplsWavebandLabel(6).  This
     object cannot be modified if gmplsLabelRowStatus is active(1)."
```

```
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 3.3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 13 }

gmplsLabelWavebandStart OBJECT-TYPE
  SYNTAX        Unsigned32
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The starting label component of a Waveband Label.  Only valid if
     gmplsLabelType is set to gmplsWavebandLabel(6).  This object
     cannot be modified if gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 3.3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 14 }

gmplsLabelWavebandEnd OBJECT-TYPE
  SYNTAX        Unsigned32
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "The end label component of a Waveband Label.  Only valid if
     gmplsLabelType is set to gmplsWavebandLabel(6).  This object
     cannot be modified if gmplsLabelRowStatus is active(1)."
  REFERENCE
    "1. Generalized Multi-Protocol Label Switching (GMPLS) Signaling
        Functional Description, RFC 3471, section 3.3."
  DEFVAL { 0 }
::= { gmplsLabelEntry 15 }

gmplsLabelStorageType OBJECT-TYPE
  SYNTAX        StorageType
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "This variable indicates the storage type for this row.  The
     agent MUST ensure that this object's value remains consistent
     with the storage type of any rows in other tables that contain
     pointers to this row.  In particular, the storage type of this
     row must be at least as permanent as that of any row that points
     to it.
     Conceptual rows having the value 'permanent' need not
     allow write-access to any columnar objects in the row."
  REFERENCE
```

```
    "1. Textual Conventions for SMIv2, STD 58, RFC 2579, section 2."
  DEFVAL { volatile }
::= { gmplsLabelEntry 16 }

gmplsLabelRowStatus OBJECT-TYPE
  SYNTAX        RowStatus
  MAX-ACCESS    read-create
  STATUS        current
  DESCRIPTION
    "This variable is used to create, modify, and/or delete a row in
     this table.  When a row in this table has a row in the active(1)
     state, no objects in this row can be modified except the
     gmplsLabelRowStatus and gmplsLabelStorageType.

     The gmplsLabelType object does not have a default and must be
     set before a row can become active.  The corresponding label
     objects (dependent on the value of gmplsLabelType) should also
     be set unless they happen to need to use the specified default
     values as follows:

     gmplsLabelType setting             objects to be set
     ------------------------------------------------------------
     gmplsMplsLabel(1)                  gmplsLabelMplsLabel

     gmplsPortWavelengthLabel(2)        gmplsLabelPortWavelength

     gmplsFreeformLabel(3)              gmplsLabelFreeform

     gmplsSonetLabel(4)                 gmplsLabelSonetSdhSignalIndex
                                        gmplsLabelSdhVc
                                        gmplsLabelSdhVcBranch
                                        gmplsLabelSonetSdhBranch
                                        gmplsLabelSonetSdhGroupBranch

     gmplsSdhLabel(5)                   gmplsLabelSonetSdhSignalIndex
                                        gmplsLabelSdhVc
                                        gmplsLabelSdhVcBranch
                                        gmplsLabelSonetSdhBranch
                                        gmplsLabelSonetSdhGroupBranch

     gmplsWavebandLabel(6)              gmplsLabelWavebandId
                                        gmplsLabelWavebandStart
                                        gmplsLabelWavebandEnd"
::= { gmplsLabelEntry 17 }

gmplsLabelGroups
  OBJECT IDENTIFIER ::= { gmplsLabelConformance 1 }
```

```
gmplsLabelCompliances
  OBJECT IDENTIFIER ::= { gmplsLabelConformance 2 }

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

  MODULE -- this module

  -- The mandatory groups have to be implemented by LSRs claiming
  -- support for this MIB module.  This MIB module is, however, not
  -- mandatory for a working implementation of a GMPLS LSR with full
  -- MIB support if the GMPLS Labels in use can be represented within
  -- a 32-bit quantity.

  MANDATORY-GROUPS {
    gmplsLabelTableGroup
  }

  GROUP gmplsLabelPacketGroup
  DESCRIPTION
    "This group extends gmplsLabelTableGroup for implementations that
     support Packet Labels.  It is optional for implementations that
     do not support Packet Labels."

  GROUP gmplsLabelPortWavelengthGroup
  DESCRIPTION
    "This group extends gmplsLabelTableGroup for implementations that
     support Port and Wavelength Labels.  It is optional for
     implementations that do not support Wavelength Labels."

  GROUP gmplsLabelFreeformGroup
  DESCRIPTION
    "This group extends gmplsLabelTableGroup for implementations that
     support Freeform Labels.  It is optional for implementations that
     do not support Freeform Labels."

  GROUP gmplsLabelSonetSdhGroup
  DESCRIPTION
    "This group extends gmplsLabelTableGroup for implementations that
     support SONET or SDH Labels.  It is optional for implementations
     that do not support SONET or SDH Labels."

  GROUP gmplsLabelWavebandGroup
  DESCRIPTION
```

```
    "This group extends gmplsLabelTableGroup for implementations that
     support Waveband Labels.  It is optional for implementations that
     do not support Waveband Labels."
  OBJECT      gmplsLabelType
  MIN-ACCESS  read-only
  DESCRIPTION
    "Write access is not required."

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

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

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

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

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

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

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

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

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

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

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

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

::= { gmplsLabelCompliances 1 }

gmplsLabelModuleFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that support the complete
     GMPLS-LABEL-STD-MIB module.

     The mandatory groups have to be implemented by GMPLS LSRs
     claiming support for this MIB module.  This MIB module is,
     however, not mandatory for a working implementation of a GMPLS
     LSR with full MIB support if the GMPLS Labels in use can be
     represented within a 32-bit quantity."

  MODULE -- this module

  MANDATORY-GROUPS {
    gmplsLabelTableGroup
  }

::= { gmplsLabelCompliances 2 }
```

```
gmplsLabelTableGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelIndexNext,
    gmplsLabelType,
    gmplsLabelStorageType,
    gmplsLabelRowStatus
  }

  STATUS  current
  DESCRIPTION
    "Necessary, but not sufficient, set of objects to implement label
     table support.  In addition, depending on the type of labels
     supported, the following other groups defined below are
     mandatory:

       gmplsLabelWavebandGroup and/or
       gmplsLabelPacketGroup and/or
       gmplsLabelPortWavelengthGroup and/or
       gmplsLabelFreeformGroup and/or
       gmplsLabelSonetSdhGroup."
::= { gmplsLabelGroups 1 }

gmplsLabelPacketGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelMplsLabel
  }
  STATUS  current
  DESCRIPTION
    "Object needed to implement Packet (MPLS) Labels."
::= { gmplsLabelGroups 2 }

gmplsLabelPortWavelengthGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelPortWavelength
  }
  STATUS  current
  DESCRIPTION
    "Object needed to implement Port and Wavelength Labels."
::= { gmplsLabelGroups 3 }

gmplsLabelFreeformGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelFreeform
  }
  STATUS  current
  DESCRIPTION
    "Object needed to implement Freeform Labels."
::= { gmplsLabelGroups 4 }
```

```
gmplsLabelSonetSdhGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelSonetSdhSignalIndex,
    gmplsLabelSdhVc,
    gmplsLabelSdhVcBranch,
    gmplsLabelSonetSdhBranch,
    gmplsLabelSonetSdhGroupBranch
  }
  STATUS  current
  DESCRIPTION
    "Objects needed to implement SONET and SDH Labels."
::= { gmplsLabelGroups 5 }

gmplsLabelWavebandGroup OBJECT-GROUP
  OBJECTS {
    gmplsLabelWavebandId,
    gmplsLabelWavebandStart,
    gmplsLabelWavebandEnd
  }
  STATUS  current
  DESCRIPTION
    "Objects needed to implement Waveband Labels."
::= { gmplsLabelGroups 6 }

END
```

9. Security Considerations

It is clear that the MIB modules described in this document in association with MPLS-LSR-STD-MIB [RFC3813] are potentially useful for monitoring of GMPLS LSRs. These MIB modules can also be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

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

- the gmplsInterfaceTable, gmplsInSegmentTable, gmplsOutSegmentTable, and gmplsLabelTable collectively contain objects to provision GMPLS interfaces, LSPs, and their associated parameters on a Label Switching Router (LSR). Unauthorized write access to objects in these tables could result in disruption of

traffic on the network. This is especially true if an LSP 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:

- the gmplsInterfaceTable, gmplsInSegmentTable, gmplsOutSegmentTable, and gmplsLabelTable collectively show the LSP network topology and its capabilities. 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 the MIB tables in [RFC3813]. The authors would like to express their gratitude to all those who worked on that earlier MIB document.

The authors would like to express their thanks to Dan Joyle for his careful review and comments on early versions of the label table. Special thanks to Joan Cucchiara and Len Nieman for their help with

compilation issues. Lars Eggert, Tom Petch, Dan Romascanu, and Bert Wijnen provided useful input in the final stages of review.

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

11. IANA Considerations

IANA has rooted MIB objects in the two MIB modules 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
15	GMPLS-LSR-STD-MIB	[RFC4803]
16	GMPLS-LABEL-STD-MIB	[RFC4803]

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].

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.

[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.

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

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

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

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

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

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, January 2003.

[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.

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

[RFC4606] Mannie, E. and D. Papadimitriou, "Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control", RFC 4606, August 2006.

[RFC4801] Nadeau, T., Ed. and A. Farrel, Ed., "Definitions of Textual Conventions for Multiprotocol Label Switching (MPLS) Management", RFC 4801, February 2007.

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

12.2. Informative References

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

[RFC3468] Andersson, L. and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols", RFC 3468, February 2003.

[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.

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

Contact Information

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

EMail: tnadeau@cisco.com

Adrian Farrel
Old Dog Consulting

Phone: +44-(0)-1978-860944
EMail: adrian@olddog.co.uk

Cheenu Srinivasan
Bloomberg L.P.
731 Lexington Ave.
New York, NY 10022

Phone: +1-212-617-3682
EMail: cheenu@bloomberg.net

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.