

Network Working Group
Internet Draft
Expires: July 2002

Thomas D. Nadeau
Cisco Systems, Inc.

Cheenu Srinivasan
Parama Networks, Inc.

Adrian Farrel
Movaz Networks, Inc.

Tim Hall
Edward Harrison
Data Connection Ltd.

January 2002

**Generalized Multiprotocol Label Switching (GMPLS)
Label Switching Router Management Information Base**

[draft-nadeau-ccamp-gmpls-lsr-mib-01.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#) [[RFC2026](#)].

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

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

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

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

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 Multiprotocol Label Switching (MPLS)

and Generalized Multiprotocol Label Switching (GMPLS)
Label Switched Routers (LSRs).

Table of Contents

1. Changes and Pending Work	2
1.1. Changes Since the Last Version	2
1.2. Pending Work	3
2. Introduction	3
2.1. Migration Strategy	4
3. The SNMP Management Framework	4
4. Terminology	6
5. Feature List	6
6. Outline	7
6.1. Summary of LSR MIB	7
7. Brief Description of MIB Objects	8
7.1. gmplsInterfaceConfTable	8
7.2. gmplsInterfacePerfTable	9
7.3. gmplsInSegmentTable	9
7.4. gmplsInSegmentPerfTable	9
7.5. gmplsOutSegmentTable	9
7.6. gmplsOutSegmentPerfTable	10
7.7. gmplsXCTable	10
7.8. gmplsLabelStackTable	10
7.9. gmplsTrafficParamTable	10
8. Bidirectional LSPs	10
9. Example of LSP Setup	11
10. Application of the Interface Group to MPLS	14
10.1. Support of the MPLS Layer by ifTable	15
11. The Use of RowPointer and Other Cross-References	16
11.1. RowPointer	16
11.2. Cross-referencing to the gmplsLabelTable	17
12. GMPLS Label Switching Router MIB Definitions	17
13. Security Considerations	60
14. References	62
14.1. Normative References	64
14.2. Informational References	66
15. Authors' Addresses	66
16. Full Copyright Statement	67

[1. Changes and Pending Work](#)

This section must be removed before the draft progresses
to RFC.

1.1. Changes Since the Last Version

Nadeau et al.

Expires July 2002

[Page 2]

- Clarify which objects can be modified when row-status and admin-status are set to active.
- Apply bug fixes in line with updates to [[LSRMIB](#)].
- Update examples and provide more detail.

[1.2.](#) Pending Work

The following work items have been identified for this draft. They will be addressed in a future version.

- Expand conformance statements to give one for monitoring only, and one for monitoring and control.
- Bring references up to date, include all drafts referenced from this document, and exclude those that are not referenced.
- Provide objects or tables to support getNextIndex for all arbitrary indexes.
- Provide support for monitoring tunnel resources in GMPLS systems. For example, SONET/SDH or G.709. This might be done through an arbitrary RowPointer to an external MIB.
- Enhance gmplsTrafficParamsTable to support GMPLS traffic parameters for signaled and static LSPs.
- Enhance performance tables for technology-specific GMPLS LSPs.

[2.](#) 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 Multiprotocol Label Switching (MPLS) [[RFC3031](#)] and Generalized Multiprotocol Label Switching (GMPLS) [[GMPLSArch](#)] Label Switching Routers (LSRs).

Comments should be made directly to the CCAMP mailing

Nadeau et al.

Expires July 2002

[Page 3]

list at ccamp@ops.ietf.org.

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

[2.1.](#) Migration Strategy

This MIB is built upon the Label Switching Router MIB defined for use with MPLS [[LSRMIB](#)]. The only changes made are additions for support of GMPLS or changes that are necessary to support the increased complexity of a GMPLS system.

In all cases, these changes have been made such that migration from [[LSRMIB](#)] to this MIB will be as simple as possible.

Note that this MIB may be used in systems that support MPLS, GMPLS or both.

This MIB may be seen as a replacement for the MPLS LSR MIB [[LSRMIB](#)] in systems which support GMPLS, but it is not a requirement that it replace the MPLS LSR MIB in systems that only support MPLS.

The companion document modeling and managing GMPLS based traffic engineering [[GMPLSTEMIB](#)] is based on the MPLS TE MIB [[TEMIB](#)] with the same intentions. It is not expected that a system would mix MPLS and GMPLS MIBs.

Textual conventions and OBJECT-IDENTIFIERS are defined in [[GMPLSTCMIB](#)] which extends the set of textual conventions originally defined in [[TCMIB](#)].

See [section 11.1](#) for a description of how the `gmplsLabelTable` may be omitted in systems that support MPLS only.

[3.](#) The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in [RFC 2571](#) [[RFC2571](#)].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, [RFC 1155](#) [[RFC1155](#)], STD 16, [RFC 1212](#) [[RFC1212](#)] and STD 16, [RFC 1215](#) [[RFC1215](#)]. The second version, called SMIV2, is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901](#) [[RFC1901](#)] and [RFC 1906](#) [[RFC1906](#)]. The third version of the message protocol is called SNMPv3 and described in [RFC 1906](#) [[RFC1906](#)], [RFC 2572](#) [[RFC2572](#)] and [RFC 2574](#) [[RFC2574](#)].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second set of protocol operations and associated PDU formats is described in [RFC 1905](#) [[RFC1905](#)].
- A set of fundamental applications described in [RFC 2573](#) [[RFC2573](#)] and the view-based access control mechanism described in [RFC 2575](#) [[RFC2575](#)].

A more detailed introduction to the current SNMP Management Framework can be found in [RFC 2570](#) [[RFC2570](#)].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation

process. However, this loss of machine readable
information is not considered to change the semantics of

the MIB.

4. Terminology

This document uses terminology from the document describing the MPLS architecture [[RFC3031](#)].

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 a Label Switching Router (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.

5. Feature List

The GMPLS Label Switching Router MIB is designed to satisfy the following requirements and constraints:

- The MIB supports both manually configured LSPs as well as those configured via any MPLS or GMPLS signaling protocol.
- The MIB supports the enabling and disabling of MPLS capability on MPLS capable interfaces of an LSR.
- The MIB allows resource sharing between two or more LSPs.
- Both per-platform and per-interface label spaces are supported.
- Forwarding can be performed solely based on an incoming top label [[RFC3031](#), [RFC3032](#)].
- Support is provided for next-hop resolution when the outgoing interface is a shared media interface. In the point-to-multipoint case, each outgoing segment can reside on a different shared media interface.
- The MIB supports point-to-point, point-to-multipoint and multipoint-to-point connections at an LSR.
- For multipoint-to-point connections all outgoing

packets can have the same top label.

- For multipoint-to-point connections, the outgoing resources of the merged connections can be shared.
- For multipoint-to-point connections, packets from different incoming connections can have distinct outgoing label stacks beneath the (identical) top label.
- In the point-to-multipoint case each outgoing connection can have a distinct label stack including the top label.
- All the members of a point-to-multipoint connection can share the resources allocated for the ingress segments.
- The MIB provides cross-connect capability to "pop" an incoming label and forward the packet with the remainder of the label stack unchanged and without pushing any labels ("pop-and-go") [[RFC3032](#)].
- The MIB supports persistent as well as non-persistent LSPs.
- Performance counters are provided for in-segments and out-segments as well as for measuring performance on a per-interface basis.

6. Outline

Configuring LSPs through an LSR involves the following steps:

- Enabling MPLS on MPLS capable interfaces.
- Configuring in-segments and out-segments.
- Setting up the cross-connect table to associate segments and/or to indicate connection origination and termination.
- Optionally specifying label stack actions.
- Optionally specifying segment traffic parameters.

6.1. Summary of LSR MIB

The MIB objects for performing these actions consist of

Nadeau et al.

Expires July 2002

[Page 7]

the following tables:

- The interface configuration table (gmplsInterfaceConfTable), which is used for enabling the MPLS protocol on MPLS-capable interfaces.
- The in-segment (gmplsInSegmentTable) and out-segment (gmplsOutSegmentTable) tables, which are used for configuring LSP segments at an LSR.
- The cross-connect table (gmplsXCTable), which is used to associate in and out segments together, in order to form a cross-connect.
- The label stack table (gmplsLabelStackTable), which is used for specifying label stack operations.
- The Traffic Parameter table (gmplsTrafficParamTable), which is used for specifying LSP-related traffic parameters.

Further, the in-segment and out-segment performance tables, gmplsInSegmentPerfTable and gmplsOutSegmentPerfTable, contain the objects necessary to measure the performance of LSPs, and gmplsInterfacePerfTable has objects to measure performance on a per-interface basis.

These tables are described in the subsequent sections.

7. Brief Description of MIB Objects

Sections [7.1-7.2](#) describe objects pertaining to MPLS-capable interfaces of an LSR. The objects described in Sections [7.3-7.8](#), were derived from the Incoming Label Map (ILM) and Next Hop Label Forwarding Entry (NHLFE) as specified in the MPLS architecture document [[RFC3031](#)]. [Section 7.9](#) describes objects for specifying traffic parameters for in and out segments. It is appropriate to note that the in-segment, out-segment, and cross-connect tables were modeled after similar tables found in [[RFC2515](#)].

[7.1.](#) gmplsInterfaceConfTable

This table represents the interfaces that are MPLS or

GMPLS capable. An LSR creates an entry in this table for every such interface on that LSR.

7.2. gmplsInterfacePerfTable

This table contains objects to measure the performance of MPLS and GMPLS capable interfaces and is an AUGMENT to gmplsInterfaceConfTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

7.3. gmplsInSegmentTable

This table contains a description of the incoming segments to an LSR and their associated parameters.

The administrative and operational status objects for this table are used to control packet transmission on this segment. If either the administrative or operational status objects for this table is set to 'down', this implies that packets will not be forwarded. Likewise, if the values are set to 'up' this implies that packets are forwarded. These values are particularly useful in cases where multi-point connections utilize a single cross-connect and the administrator wishes to disable some, but not all of the streams. In these cases, the administrator may set the administrative status object to 'down' on some of the in-segments.

7.4. gmplsInSegmentPerfTable

The In-Segment Performance Table has objects to measure the performance of an incoming segment configured on an LSR. It is an AUGMENT to gmplsInSegmentTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

7.5. gmplsOutSegmentTable

The Out-Segment Table contains a description of the outgoing segments at an LSR and their associated parameters. The administrative and operational status objects for this table are used to control packet transmission on this segment. If either the administrative or operational status objects is set to 'down', this implies that packets will not be forwarded.

Likewise, if the values are set to 'up' this implies that packets are forwarded. These values are particularly

useful in cases where multicast connections utilize a single cross-connect and the administrator wishes to disable some, but not all of the streams. In these cases, the administrator may set the administrative status object to 'down' on some of the out-segments.

7.6. gmplsOutSegmentPerfTable

The Out-Segment Table contains objects to measure the performance of an outgoing segment configured on an LSR. It is an AUGMENT to gmplsOutSegmentTable. High capacity counters are provided for objects that are likely to wrap around quickly on high-speed interfaces.

7.7. gmplsXCTable

The gmplsXCTable specifies information for associating segments together in order to instruct the LSR to switch between the specified segments. It supports point-to-point, point-to-multi-point and multi-point-to-point connections.

The administrative and operational status objects for this table imply control of packet forwarding to and from a XCEntry. When the administrative and operational status objects are set to 'down' for example, this implies that the specified XCEntry will not forward packets. Likewise, when either is set to 'up' this implies that packets will be forwarded.

7.8. gmplsLabelStackTable

The gmplsLabelStackTable specifies the label stack to be pushed onto a packet, beneath the top label. Entries to this table are referred to from gmplsXCTable.

7.9. gmplsTrafficParamTable

The gmplsTrafficParamTable contains objects for specifying the traffic parameters of in-segments and out-segments. Entries in this table are referred to from gmplsInSegmentTable and gmplsOutSegmentTable.

8. Bidirectional LSPs

Nadeau et al.

Expires July 2002

[Page 10]

This MIB supports bidirectional LSPs as required for GMPLS.

A single entry in the `gmplsXCIndex` is shared by all of the segments for the entire bidirectional LSP. This facilitates a simple reference from [[GMPLSTEMIB](#)] 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.

9. Example of LSP Setup

In this section we provide a brief example of using the MIB objects described in [section 12](#). to set up an LSP. While this example is not meant to illustrate every nuance of the MIB, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB itself.

Suppose that one would like to manually create a best-effort, bi-directional 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 the LSP enters via interface B and leaves via interface A (i.e. the forward and reverse directions use the same bi-directional 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.

First, the traffic parameter entries must be set-up for both segments.

In gmplsTrafficParamTable for the incoming direction:

```
{
    gmplsTrafficParamIndex      = 5
    gmplsTrafficParamMaxRate    = 100000,
    gmplsTrafficParamMeanRate   = 100000,
    gmplsTrafficParamMaxBurstSize = 2000,
    gmplsTrafficParamRowStatus  = createAndGo(4)
}
```

In gmplsTrafficParamTable for the outgoing direction:

```
{
    gmplsTrafficParamIndex      = 6
    gmplsTrafficParamMaxRate    = 100000,
    gmplsTrafficParamMeanRate   = 100000,
    gmplsTrafficParamMaxBurstSize = 2000,
    gmplsTrafficParamRowStatus  = createAndGo(4)
}
```

Note that for a bi-directional LSP, the segments in the reverse direction can share the traffic parameter entries (and hence resources) with the segments in the forward direction.

We must next create the appropriate in-segment and out-segment entries with suitable traffic parameters by pointing to the appropriate traffic parameter entries that we have just created.

First for the forward direction:

In gmplsInSegmentTable

```
{
    gmplsInSegmentIndex      = 1,
    -- incoming interface
    gmplsInSegmentIfIndex    = 12,
    -- incoming label
    gmplsInSegmentLabel      = 21,
    gmplsInSegmentNPop       = 1,
    gmplsInSegmentOwner      = snmp (2),
    gmplsInSegmentTrafficParamPtr =
                                gmplsTrafficParamIndex.5,
    gmplsInSegmentDirection  = forward (1),
    gmplsInSegmentRowStatus  = createAndGo(4)
}
```

In gmpIsOutSegmentTable

Nadeau et al.

Expires July 2002

[Page 12]


```
{
  gmplsOutSegmentIndex      = 1,
  -- outgoing interface
  gmplsOutSegmentIfIndex    = 13,
  gmplsOutSegmentPushTopLabel = true(1),
  -- outgoing label
  gmplsOutSegmentTopLabel    = 22,
  gmplsOutSegmentOwner       = snmp (2),
  gmplsOutSegmentTrafficParamPtr =
                                gmplsTrafficParamIndex.6,
  gmplsOutSegmentDirection   = forward (1),
  gmplsOutSegmentRowStatus   = createAndGo(4)
}
```

Next for the reverse direction:

In gmplsInSegmentTable

```
{
  gmplsInSegmentIndex      = 2,
  -- incoming interface
  gmplsInSegmentIfIndex    = 13,
  -- incoming label
  gmplsInSegmentLabel      = 31,
  gmplsInSegmentNPop       = 1,
  gmplsInSegmentOwner       = snmp (2),
  gmplsInSegmentTrafficParamPtr =
                                gmplsTrafficParamIndex.6,
  gmplsInSegmentDirection   = reverse (2),
  gmplsInSegmentRowStatus   = createAndGo(4)
}
```

In gmplsOutSegmentTable

```
{
  gmplsOutSegmentIndex      = 2,
  -- outgoing interface
  gmplsOutSegmentIfIndex    = 12,
  gmplsOutSegmentPushTopLabel = true(1),
  -- outgoing label
  gmplsOutSegmentTopLabel    = 32,

  gmplsOutSegmentOwner       = snmp (2),
  gmplsOutSegmentTrafficParamPtr =
                                gmplsTrafficParamIndex.5,
  gmplsOutSegmentDirection   = reverse (2),
  gmplsOutSegmentRowStatus   = createAndGo(4)
}
```

Next, two cross-connect entries are created thereby associating the newly created segments together.

```
In gmplsXCTable:
{
    gmplsXCIndex          = 2,
    gmplsInSegmentIndex   = 1,
    gmplsOutSegmentIndex  = 1,
    -- 192.33.4.21.2
    gmplsXCLspId          = 'c021041502'H,
    gmplsXCIsPersistent   = false (1),
    gmplsXCOwner          = snmp (2),
    -- only a single outgoing label
    gmplsLabelStackIndex  = 0,
    gmplsXCRowStatus      = createAndGo(4)
}
```

```
In gmplsXCTable:
{
    gmplsXCIndex          = 3,
    gmplsInSegmentIndex   = 2,
    gmplsOutSegmentIndex  = 2,
    -- 192.33.4.21.2
    gmplsXCLspId          = 'c021041502'H,
    gmplsXCIsPersistent   = false (1),
    gmplsXCOwner          = snmp (2),
    -- only a single outgoing label
    gmplsLabelStackIndex  = 0,
    gmplsXCRowStatus      = createAndGo(4)
}
```

Note that the gmplsInSegmentXCIndex and gmplsOutSegmentXCIndex objects will automatically be populated with the values 2 and 3 (for the forward and reverse segments respectively) when these segments are referred to from the corresponding cross-connect entry.

10. Application of the Interface Group to MPLS

This memo contains media-specific extensions to the Interfaces Group for managing MPLS interfaces and assumes the interpretation of the Interfaces Group to be in accordance with [[RFC2863](#)] which states that the interfaces table (ifTable) contains information on the managed resource's interfaces and that each sub-layer below the internetwork layer of a network interface is considered an interface. Thus, the MPLS interface is represented as an entry in the ifTable. The inter-relation of entries in the ifTable is defined by

Interfaces Stack Group defined in [[RFC2863](#)].

When using MPLS interfaces, the interface stack table

might appear as follows:

```

+-----+
| MPLS-interface ifType = mpls(166)      +
+-----+
| Underlying Layer...                    +
+-----+

```

In the above diagram, "Underlying Layer..." refers to the ifIndex of any interface type, which has been defined for MPLS interworking. Examples include ATM, Frame Relay, Ethernet, etc.

10.1. Support of the MPLS Layer by ifTable

Some specific interpretations of ifTable for the MPLS layer follow.

Object	Use for the MPLS layerObject
ifIndex	Each MPLS interface is represented by an ifEntry.
ifDescr	Description of the MPLS interface.
ifType	The value that is allocated for MPLS is 166.
ifSpeed	The total bandwidth in bits per second for use by the MPLS layer.
ifPhysAddress	Unused.
ifAdminStatus	This variable indicates the administrator's intent as to whether MPLS should be enabled, disabled, or running in some diagnostic testing mode on this interface. Also see [RFC2863].
ifOperStatus	This value reflects the actual or operational status of MPLS on this interface.
ifLastChange	See [RFC2863].
ifInOctets	The number of received octets over the interface, i.e., the number of received,

octets received as labeled packets.

ifOutOctets The number of transmitted octets over the interface, i.e., the number of octets transmitted as labeled packets.

ifInErrors The number of labeled packets dropped due to uncorrectable errors.

ifInUnknownProtos The number of received packets discarded during packet header validation, including packets with unrecognized label values.

ifOutErrors See [[RFC2863](#)].

ifName Textual name (unique on this system) of the interface or an octet string of zero length.

ifLinkUpDownTrapEnable
 Default is disabled (2).

ifConnectorPresent
 Set to false (2).

ifHighSpeed See [[RFC2863](#)].

ifHCInOctets The 64-bit version of ifInOctets; supported if required by the compliance statements in [[RFC2863](#)].

ifHCOctets The 64-bit version of ifOutOctets; supported if required by the compliance statements in [[RFC2863](#)].

ifAlias The non-volatile 'alias' name for the interface as specified by a network manager.

ifCounterDiscontinuityTime
 See [[RFC2863](#)].

[11.](#) The Use of RowPointer and Other Cross-References

[11.1.](#) RowPointer

RowPointer is a textual convention used to identify a conceptual row in an SNMP Table by pointing to one of its

objects. In this MIB, it is used in `gmplsInSegmentTable` and `gmplsOutSegmentTable` for the following purposes. First, it indicates a particular traffic parameter table. An example of such a table is `gmplsTrafficParamTable`. Second, it is used to indicate a specific instance of a traffic parameter entry that is associated with a given in-segment or out-segment entry. In the in-segment and out-segment tables, the `trafficParamPtr` SHOULD point to the first column of the appropriate conceptual row.

11.2. Cross-referencing to the `gmplsLabelTable`

The `gmplsLabelTable` [GMPLSLABELMIB] provides a way to model labels in a GMPLS system where labels might not be simple 32 bit integers.

Several tables in this document (`gmplsInterfaceConfTable`, `gmplsInSegmentTable`, `gmplsOutSegmentTable`, `gmplsLabelStackTable`) use arbitrary indexes to point to entries in the `gmplsLabelTable` to indicate specific label values.

Since the primary index into `gmplsLabelTable` is a simple 32 bit integer (`gmplsLabelIndex`), in systems where the nature of a label is well-known, and where the label can safely be encoded as a 32 bit integer (for example a conventional MPLS system), the `gmplsLabelTable` does not need to be supported and the pointers to the `gmplsLabelTable` (`gmplsInterfaceLabelMinIn`, `gmplsInterfaceLabelMaxIn`, `gmplsInterfaceLabelMinOut`, `gmplsInterfaceLabelMaxOut`, `gmplsInSegmentLabel`, `gmplsOutSegmentTopLabel`, `gmplsLabelStackLabel`) may be replaced with the direct label values.

This provides both a good way to support legacy systems that implement the previous version of this MIB [[LSRMIB](#)], 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 sub-label index (`gmplsSublabelIndex`).

12. GMPLS Label Switching Router MIB Definitions

GMPLS-LSR-MIB DEFINITIONS ::= BEGIN

IMPORTS

Nadeau et al.

Expires July 2002

[Page 17]

```
MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
Integer32, Counter32, Unsigned32, Counter64,
Gauge32
    FROM SNMPv2-SMI
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF
TruthValue, RowStatus, StorageType,
RowPointer, TimeStamp
    FROM SNMPv2-TC
InterfaceIndexOrZero
    FROM IF-MIB
AddressFamilyNumbers
    FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB
InetAddressIPv4, InetAddressIPv6, InetAddressType
    FROM INET-ADDRESS-MIB
MplsLSPID, MplsBitRate, MplsBurstSize,
MplsInitialCreationSource
    FROM MPLS-TC-MIB
gmplsMIB, GmplsSegmentDirection, GmplsTrapEnable
    FROM GMPLS-TC-MIB
InetAddressIPv4, InetAddressIPv6
    FROM INET-ADDRESS-MIB
;
```

gmplsLsrMIB MODULE-IDENTITY

```
LAST-UPDATED
    "200201251200Z" -- 25 Jan 2002 12:00:00 GMT
ORGANIZATION
    "Common Control And Management Protocols
    (CCAMP) Working Group"
CONTACT-INFO
    "      Thomas D. Nadeau
    Postal: Cisco Systems, Inc.
           250 Apollo Drive
           Chelmsford, MA 01824
    Tel: +1-978-244-3051
    Email: tnadeau@cisco.com
```

```
      Cheenu Srinivasan
    Postal: Parama Networks, Inc.
           1030 Broad Street
           Shrewsbury, NJ 07702
    Tel: +1-732-544-9120 x731
    Email: cheenu@paramanet.com
```

```
      Adrian Farrel
    Postal: Movaz Networks, Inc.
```

7926 Jones Branch Drive
McLean, VA 22102
Tel: +1-703-847-1986

Nadeau et al.

Expires July 2002

[Page 18]

Email: afarrel@movaz.com

Edward Harrison
Postal: Data Connection Ltd.
100 Church Street
Enfield, Middlesex
EN2 6BQ, United Kingdom
Tel: +44-20-8366-1177
Email: eph@dataconnection.com

Tim Hall
Postal: Data Connection Ltd.
100 Church Street
Enfield, Middlesex
EN2 6BQ, United Kingdom
Tel: +44-20-8366-1177
Email: timhall@dataconnection.com

Email comments to the CCAMP WG Mailing List
at ccamp@ops.ietf.org."

DESCRIPTION

"This MIB contains managed object
definitions for the Generalized
Multiprotocol Label Switching (GMPLS)
Router."

-- Revision history.

REVISION

"200111111100Z" -- 11 Nov 2001 11:00:00 GMT

DESCRIPTION

"Initial draft version."

REVISION

"200201251200Z" -- 25 Jan 2002 12:00:00 GMT

DESCRIPTION

"Revision for compilation and work in
progress."

::= { gmplsMIB 2 }

-- Tables, Scalars

gmplsLsrObjects

OBJECT IDENTIFIER ::= { gmplsLsrMIB 1 }

-- Notifications

gmplsLsrNotifications

OBJECT IDENTIFIER ::= { gmplsLsrMIB 2 }

gmplsLsrNotifyPrefix

OBJECT IDENTIFIER ::= { gmplsLsrNotifications 0 }

-- Conformance
gmplsLsrConformance

Nadeau et al.

Expires July 2002

[Page 19]

OBJECT IDENTIFIER ::= { gmplsLsrMIB 3 }

-- GMPLS Interface Configuration Table.

gmplsInterfaceConfTable OBJECT-TYPE
SYNTAX SEQUENCE OF GmplsInterfaceConfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table specifies per-interface MPLS
capability and associated information."
::= { gmplsLsrObjects 1 }

gmplsInterfaceConfEntry OBJECT-TYPE
SYNTAX GmplsInterfaceConfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"An entry in this table is created by an LSR
for every interface capable of supporting
MPLS. Each entry in this table will exist
only if a corresponding entry in ifTable
exists. If the associated entry in ifTable
is deleted, the corresponding entry in this
table must also be deleted shortly
thereafter. 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. Other entries defined
in this table represent additional MPLS
interfaces that may participate in either
the per-platform or per-interface label
spaces, or both. Additional information
about label space participation of an
interface is provided in the description
clause of
mplsInterfaceLabelParticipationType."
INDEX { gmplsInterfaceConfIndex }
::= { gmplsInterfaceConfTable 1 }

GmplsInterfaceConfEntry ::= SEQUENCE {
gmplsInterfaceConfIndex InterfaceIndexOrZero,
gmplsInterfaceLabelMinIn Unsigned32,
gmplsInterfaceLabelMaxIn Unsigned32,
gmplsInterfaceLabelMinOut Unsigned32,

gmplsInterfaceLabelMaxOut	Unsigned32,
gmplsInterfaceTotalBandwidth	MplsBitRate,


```
    gmplsInterfaceAvailableBandwidth      MplsBitRate,
    gmplsInterfaceLabelParticipationType  BITS,
    gmplsInterfaceConfSignalingCaps      BITS
}
```

gmplsInterfaceConfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This is a unique index for an entry in the gmplsInterfaceConfTable. A non-zero index for an entry indicates the ifIndex for the corresponding interface entry in of the MPLS-layer in the ifTable. Note that the per-platform label space may apply to several interfaces, and therefore the configuration of the per-platform label space interface parameters will apply to all of the interfaces that are participating in the per-platform label space."

REFERENCE

"[RFC 2863](#) - The Interfaces Group MIB, McCloghrie, K., and F. Kastenholtz, June 2000"

::= { gmplsInterfaceConfEntry 1 }

gmplsInterfaceLabelMinIn OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is the minimum value of an MPLS label that this LSR is willing to receive on this interface. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsInterfaceConfEntry 2 }

gmplsInterfaceLabelMaxIn OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

Nadeau et al.

Expires July 2002

[Page 21]

"This is the maximum value of an MPLS label that this LSR is willing to receive on this interface. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsInterfaceConfEntry 3 }

gmplsInterfaceLabelMinOut OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is the minimum value of an MPLS label that this LSR is willing to send on this interface. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsInterfaceConfEntry 4 }

gmplsInterfaceLabelMaxOut OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is the maximum value of an MPLS label that this LSR is willing to send on this interface. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsInterfaceConfEntry 5 }

gmplsInterfaceTotalBandwidth OBJECT-TYPE

SYNTAX	MplsBitRate
MAX-ACCESS	read-only
STATUS	current

DESCRIPTION

"This value indicates the total amount of usable bandwidth on this interface and is specified in kilobits per second (Kbps). This variable is not applicable when applied to the interface with index 0. When this value cannot be measured, this value should contain the nominal bandwidth."

::= { gmplsInterfaceConfEntry 6 }

gmplsInterfaceAvailableBandwidth OBJECT-TYPE

SYNTAX MplsBitRate

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This value indicates the total amount of available bandwidth available on this interface and is specified in kilobits per second (Kbps). This value is calculated as the difference between the amount of bandwidth currently in use and that specified in gmplsInterfaceTotalBandwidth. This variable is not applicable when applied to the interface with index 0. When this value cannot be measured, this value should contain the nominal bandwidth."

::= { gmplsInterfaceConfEntry 7 }

gmplsInterfaceLabelParticipationType OBJECT-TYPE

SYNTAX BITS {

perPlatform (0),

perInterface (1)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Either the perPlatform(0) or perInterface(1) bit MUST be set. If the value of the gmplsInterfaceConfIndex for this entry is zero, then only the perPlatform(0) bit MUST be set and the perInterface(1) bit is meaningless. If the perInterface(1) bit is set then the value of gmplsInterfaceLabelMinIn, gmplsInterfaceLabelMaxIn,

gmplsInterfaceLabelMinOut, and
gmplsInterfaceLabelMaxOut for this entry
reflect the label ranges for this

interface. If only the perPlatform(0) bit is set, then the value of gmplsInterfaceLabelMinIn, gmplsInterfaceLabelMaxIn, gmplsInterfaceLabelMinOut, and gmplsInterfaceLabelMaxOut for this entry must be identical to the instance of these objects with index 0."

REFERENCE

"Multiprotocol Label Switching, Rosen et al, [RFC 3031](#), January 2001."

::= { gmplsInterfaceConfEntry 8 }

gmplsInterfaceConfSignalingCaps OBJECT-TYPE

SYNTAX BITS {

 rsvpMpls (0),
 crldpMpls (1),
 ldpMpls (2),
 otherMpls (3),
 rsvpGmpls (4),
 crldpGmpls (5),
 otherGmpls (6)

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Defines the signaling capabilities on this interface. Multiple bits may legitimately be set at once. Setting no bits implies that signaling cannot be performed on this interface and all LSPs must be manually provisioned."

::= { gmplsInterfaceConfEntry 9 }

-- End of gmplsInterfaceConfTable

-- MPLS Interface Performance Table.

gmplsInterfacePerfTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsInterfacePerfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table provides MPLS performance information on a per-interface basis."

::= { gmplsLsrObjects 2 }

gmplsInterfacePerfEntry OBJECT-TYPE
SYNTAX GmplsInterfacePerfEntry
MAX-ACCESS not-accessible


```
STATUS          current
DESCRIPTION
    "An entry in this table is created by the
    LSR for every interface capable of
    supporting MPLS. Its is an extension to
    the gmplsInterfaceConfEntry table."
AUGMENTS        { gmplsInterfaceConfEntry }
::= { gmplsInterfacePerfTable 1 }
```

```
GmplsInterfacePerfEntry ::= SEQUENCE {
    -- incoming direction
    gmplsInterfaceInLabelsUsed      Gauge32,
    gmplsInterfaceFailedLabelLookup Counter32,
    -- outgoing direction
    gmplsInterfaceOutLabelsUsed     Gauge32,
    gmplsInterfaceOutFragments      Counter32
}
```

gmplsInterfaceInLabelsUsed OBJECT-TYPE

```
SYNTAX          Gauge32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object counts the number of labels
    that are in use at this point in time on
    this interface in the incoming direction.
    If the interface participates in the per-
    platform label space only, then this
    instance of this object MUST be identical
    with the instance with index 0. If the
    interface participates in the per-interface
    label space, then this this instance of
    this object MUST represent the number of
    per-interface labels that are in use at
    this point in time on this interface."
::= { gmplsInterfacePerfEntry 1 }
```

gmplsInterfaceFailedLabelLookup OBJECT-TYPE

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object counts the number of labeled
    packets that have been received on this
    interface and were discarded because there
    was no matching cross-connect entry. This
    object MUST count on a per-interface basis
```

```
    regardless of which label space the  
    interface participates in."  
 ::= { gmplsInterfacePerfEntry 2 }
```

gmplsInterfaceOutLabelsUsed OBJECT-TYPE

SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"This object counts the number of top-most labels in the outgoing label stacks that are in use at this point in time on this interface. This object MUST count on a per-interface basis regardless of which label space the interface participates in."

::= { gmplsInterfacePerfEntry 3 }

gmplsInterfaceOutFragments OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"This object counts the number of outgoing MPLS packets that required fragmentation before transmission on this interface. This object transmission on this interface. This object MUST count on a per-interface basis regardless of which label space the interface participates in."

::= { gmplsInterfacePerfEntry 4 }

-- In-segment table.

gmplsInSegmentIndexNext OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"This object contains the next appropriate value to be used for gmplsInSegmentIndex when creating entries in the gmplsInSegmentTable. If the number of unassigned entries is exhausted, this object will take on the value of 0. To obtain the gmplsInSegmentIndex value for a new entry, the manager must first issue a management protocol retrieval operation to obtain the current value of this object. The agent should modify the value to reflect the next unassigned index after

each retrieval operation. After a manager retrieves a value the agent will determine through its local policy when this index

value will be made available for reuse."
 ::= { gmplsLsrObjects 3 }

gmplsInSegmentTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsInSegmentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains a collection of
incoming segments to an LSR."

::= { gmplsLsrObjects 4 }

gmplsInSegmentEntry OBJECT-TYPE

SYNTAX GmplsInSegmentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents one
incoming segment. An entry can be created
by a network administrator or an SNMP
agent, or an MPLS signaling protocol. The
creator of the entry is denoted by
gmplsInSegmentOwner. The value of
gmplsInSegmentRowStatus cannot be
active(1) unless the ifTable entry
corresponding to gmplsInSegmentIfIndex
exists."

INDEX { gmplsInSegmentIndex }

::= { gmplsInSegmentTable 1 }

GmplsInSegmentEntry ::= SEQUENCE {

gmplsInSegmentIndex	Unsigned32,
gmplsInSegmentIfIndex	InterfaceIndexOrZero,
gmplsInSegmentLabel	Unsigned32,
gmplsInSegmentNPop	Integer32,
gmplsInSegmentAddrFamily	AddressFamilyNumbers,
gmplsInSegmentXCIndex	Unsigned32,
gmplsInSegmentOwner	MplsInitialCreationSource,
gmplsInSegmentTrafficParamPtr	RowPointer,
gmplsInSegmentRowStatus	RowStatus,
gmplsInSegmentStorageType	StorageType,
gmplsInSegmentDirection	GmplsSegmentDirection

}

gmplsInSegmentIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS not-accessible

STATUS	current
DESCRIPTION	

Nadeau et al.

Expires July 2002

[Page 27]

"This value contains a unique index for this row. While a value of 0 is not valid as an index for this row it can be supplied as a valid value to index gmplsXCTable to access entries for which no in-segment has been configured."

::= { gmplsInSegmentEntry 1 }

gmplsInSegmentIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This is a unique index for an entry in the gmplsInSegmentTable. This value represents the interface index for the incoming MPLS interface. A value of zero represents an incoming label from the per-platform label space."

::= { gmplsInSegmentEntry 2 }

gmplsInSegmentLabel OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The incoming label for this segment. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsInSegmentEntry 3 }

gmplsInSegmentNPop OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of labels to pop from the incoming data. Note that technologies which do not support examining label stacks and multi-label popping should set this value to its default value of 1. Note

further that in some GMPLS technologies
labels are not pushed and popped at each
LSR, but where LSRs handle LSP hierarchies

this function is used. This object cannot be modified if gmplsInSegmentRowStatus is active(1)."

DEFVAL { 1 }
::= { gmplsInSegmentEntry 4 }

gmplsInSegmentAddrFamily OBJECT-TYPE

SYNTAX AddressFamilyNumbers
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The IANA address family [[IANAFamily](#)] of the incoming packet. A value of other(0) indicates that the family type is either unknown or undefined. This object cannot be modified if gmplsInSegmentRowStatus is active(1)."

DEFVAL { other }
::= { gmplsInSegmentEntry 5 }

gmplsInSegmentXCIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Index into gmplsXCTable which identifies which cross-connect entry this segment is part of. A value of zero indicates that this entry is not referred to by any cross-connect entry. When a cross-connect entry is created which this in-segment is a part of, this object is automatically updated to reflect the value of gmplsXCIndex of that cross-connect entry."

DEFVAL { 0 }
::= { gmplsInSegmentEntry 6 }

gmplsInSegmentOwner OBJECT-TYPE

SYNTAX MplsInitialCreationSource
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Denotes the entity that created and is responsible for managing this segment. This object cannot be modified if gmplsInSegmentRowStatus is active(1)."

DEFVAL { unknown }

```
::= { gmplsInSegmentEntry 7 }
```

```
gmplsInSegmentTrafficParamPtr OBJECT-TYPE
```

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This variable represents a pointer to the traffic parameter specification for this in-segment. This value may point at an entry in the gmplsTrafficParamTable to indicate which gmplsTrafficParamEntry is to be assigned to this segment. This value may optionally point at an externally defined traffic parameter specification table. A value of zero-dot-zero indicates best-effort treatment. By having the same value of this object, two or more segments can indicate resource sharing. This object cannot be modified if
gmplsInSegmentRowStatus is active(1)."

::= { gmplsInSegmentEntry 8 }

gmplsInSegmentRowStatus 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. This object cannot be modified if
gmplsInSegmentRowStatus is active(1). If this variable is set to readOnly(5), and the corresponding TFIB entry is removed, then the agent must remove this row shortly thereafter."

::= { gmplsInSegmentEntry 9 }

gmplsInSegmentStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This variable indicates the storage type for this table entry. When set to 'permanent', the entire row is to be stored."

::= { gmplsInSegmentEntry 10 }

gmplsInSegmentDirection OBJECT-TYPE

SYNTAX	GmplsSegmentDirection
MAX-ACCESS	read-create
STATUS	current

DESCRIPTION

"This variable indicates the direction of
data flow on this segment."

DEFVAL { forward }

::= { gmplsInSegmentEntry 11 }

-- End of gmplsInSegmentTable

-- In-segment performance table.

gmplsInSegmentPerfTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsInSegmentPerfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains statistical information
for incoming MPLS segments to an LSR."

::= { gmplsLsrObjects 5 }

gmplsInSegmentPerfEntry OBJECT-TYPE

SYNTAX GmplsInSegmentPerfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table contains statistical
information about one incoming segment
which was configured in the
gmplsInSegmentTable. The counters in this
entry should behave in a manner similar to
that of the interface."

AUGMENTS { gmplsInSegmentEntry }

::= { gmplsInSegmentPerfTable 1 }

GmplsInSegmentPerfEntry ::= SEQUENCE {

gmplsInSegmentOctets Counter32,

gmplsInSegmentPackets Counter32,

gmplsInSegmentErrors Counter32,

gmplsInSegmentDiscards Counter32,

-- high capacity counter

gmplsInSegmentHCOctets Counter64,

gmplsInSegmentPerfDiscontinuityTime TimeStamp

}

gmplsInSegmentOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS	current
DESCRIPTION	

Nadeau et al.

Expires July 2002

[Page 31]

"This value represents the total number of octets received by this segment."
 ::= { gmplsInSegmentPerfEntry 1 }

gmplsInSegmentPackets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Total number of packets received by this segment."
 ::= { gmplsInSegmentPerfEntry 2 }

gmplsInSegmentErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of errored packets received on this segment."
 ::= { gmplsInSegmentPerfEntry 3 }

gmplsInSegmentDiscards OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of labeled packets received on this in-segment, which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a labeled packet could be to free up buffer space."
 ::= { gmplsInSegmentPerfEntry 4 }

gmplsInSegmentHCOctets OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of octets received. This is the 64 bit version of gmplsInSegmentOctets."
 ::= { gmplsInSegmentPerfEntry 5 }

gmplsInSegmentPerfDiscontinuityTime OBJECT-TYPE

SYNTAX	TimeStamp
MAX-ACCESS	read-only
STATUS	current

DESCRIPTION

"The value of sysUpTime on the most recent occasion at which any one or more of this segment's Counter32 or Counter64 suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."

::= { gmplsInSegmentPerfEntry 6 }

-- End of gmplsInSegmentPerfTable.

-- Out-segment table.

gmplsOutSegmentIndexNext OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains the next appropriate value to be used for gmplsOutSegmentIndex when creating entries in the gmplsOutSegmentTable. If the number of unassigned entries is exhausted, this object will take on the value of 0. To obtain the gmplsOutSegmentIndex value for a new entry, the manager must first issue a management protocol retrieval operation to obtain the current value of this object. The agent should modify the value to reflect the next unassigned index after each retrieval operation. After a manager retrieves a value the agent will determine through its local policy when this index value will be made available for reuse."

::= { gmplsLsrObjects 6 }

gmplsOutSegmentTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsOutSegmentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains a representation of the outgoing segments from an LSR."

::= { gmplsLsrObjects 7 }

gmplsOutSegmentEntry OBJECT-TYPE
SYNTAX GmplsOutSegmentEntry

Nadeau et al.

Expires July 2002

[Page 33]

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents one outgoing segment. An entry can be created by a network administrator or an SNMP agent, or an MPLS signaling protocol. The object gmplsOutSegmentOwner indicates the creator of this entry. The value of gmplsOutSegmentRowStatus cannot be active(1) unless the ifTable entry corresponding to gmplsOutSegmentIfIndex exists."

INDEX { gmplsOutSegmentIndex }

::= { gmplsOutSegmentTable 1 }

```
GmplsOutSegmentEntry ::= SEQUENCE {
    gmplsOutSegmentIndex          Unsigned32,
    gmplsOutSegmentIfIndex
    InterfaceIndexOrZero,
    gmplsOutSegmentPushTopLabel   TruthValue,
    gmplsOutSegmentTopLabel       Unsigned32,
    gmplsOutSegmentNextHopIpAddrType InetAddressType,
    gmplsOutSegmentNextHopIpv4Addr InetAddressIPv4,
    gmplsOutSegmentNextHopIpv6Addr InetAddressIPv6,
    gmplsOutSegmentXCIndex        Unsigned32,
    gmplsOutSegmentOwner          MplsInitialCreationSource,
    gmplsOutSegmentTrafficParamPtr RowPointer,
    gmplsOutSegmentRowStatus       RowStatus,
    gmplsOutSegmentStorageType     StorageType,
    gmplsOutSegmentDirection       GmplsSegmentDirection,
    gmplsOutSegmentTTLDecrement    Unsigned32
}
```

gmplsOutSegmentIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This value contains a unique index for this row. While a value of 0 is not valid as an index for this row it can be supplied as a valid value to index gmplsXCTable to access entries for which no out-segment has been configured."

::= { gmplsOutSegmentEntry 1 }

gmplsOutSegmentIfIndex OBJECT-TYPE
SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This value must contain the interface index of the outgoing interface. This object may only take the value of 0 during the create-and-wait row creation operation if the management station has not yet set the value of this object. This object cannot be modified if gmplsOutSegmentRowStatus is active(1). The gmplsOutSegmentRowStatus cannot be set to active(1) until this object is set to a value corresponding to a valid ifEntry."

DEFVAL { 0 }

::= { gmplsOutSegmentEntry 2 }

gmplsOutSegmentPushTopLabel OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This value indicates whether or not a top label should be pushed onto the outgoing label stack. The value of this variable MUST be set to true if the outgoing interface does not support pop-and-go (for example an ATM interface) or if it is a tunnel origination. Note that it is considered an error in the case that gmplsOutSegmentPushTopLabel is set to false, but the cross-connect entry which refers to this out-segment has a non-zero gmplsLabelStackIndex. The LSR MUST ensure that this situation does not happen. Note that in many GMPLS technologies labels are not popped and pushed at each LSR, but are passed through. This pass-through approach is equivalent to pop and push. This object cannot be modified if gmplsOutSegmentRowStatus is active(1)."

::= { gmplsOutSegmentEntry 3 }

gmplsOutSegmentTopLabel OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If gmpIsOutSegmentPushTopLabel is true then
this represents the label that should be

pushed onto the top of the outgoing packet's label stack. This object contains the index into the gmplsLabelTable of the entry that contains the label value to apply. Note that in implementations where the label may be encoded within a 32 bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use. If no label is to be pushed this value SHOULD be set to 0 by the management station and MUST be ignored by the agent. This object cannot be modified if gmplsOutSegmentRowStatus is active(1)."

DEFVAL { 0 }
::= { gmplsOutSegmentEntry 4 }

gmplsOutSegmentNextHopIpAddressType OBJECT-TYPE

SYNTAX InetAddressType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Indicates whether the next hop address is IPv4 or IPv6. Note that a value of unknown (0) is valid only when the outgoing interface is of type point-to-point. This object cannot be modified if gmplsOutSegmentRowStatus is active(1)."
DEFVAL { unknown }
::= { gmplsOutSegmentEntry 5 }

gmplsOutSegmentNextHopIPv4Addr OBJECT-TYPE

SYNTAX InetAddressIPv4
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"IPv4 Address of the next hop. Its value is significant only when gmplsOutSegmentNextHopIpAddressType is ipv4 (1), otherwise it SHOULD return a value of 0. This object cannot be modified if gmplsOutSegmentRowStatus is active(1)."
::= { gmplsOutSegmentEntry 6 }

gmplsOutSegmentNextHopIPv6Addr OBJECT-TYPE

SYNTAX InetAddressIPv6
MAX-ACCESS read-create

STATUS current

DESCRIPTION

"IPv6 address of the next hop. Its value is

significant only when
gmplsOutSegmentNextHopIpAddressType is ipv6
(2), otherwise it SHOULD return a value of
0. This object cannot be modified if
gmplsOutSegmentRowStatus is active(1)."
::= { gmplsOutSegmentEntry 7 }

gmplsOutSegmentXCIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Index into gmplsXCTable which identifies
 which cross-connect entry this segment is
 part of. A value of zero indicates that
 this entry is not referred to by any cross-
 connect entry. When a cross-connect entry
 is created which this out-segment is a part
 of, this object is automatically updated to
 reflect the value of gmplsXCIndex of that
 cross-connect entry."
DEFVAL { 0 }
::= { gmplsOutSegmentEntry 8 }

gmplsOutSegmentOwner OBJECT-TYPE

SYNTAX MplsInitialCreationSource
MAX-ACCESS read-create
STATUS current
DESCRIPTION
 "Denotes the entity which created and is
 responsible for managing this segment.
 This object cannot be modified if
 gmplsOutSegmentRowStatus is active(1)."
DEFVAL { unknown }
::= { gmplsOutSegmentEntry 9 }

gmplsOutSegmentTrafficParamPtr OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
 "This variable represents a pointer to the
 traffic parameter specification for this
 out-segment. This value may point at an
 entry in the gmplsTrafficParamTable to
 indicate which gmplsTrafficParamEntry is to
 be assigned to this segment. This value

may optionally point at an externally
defined traffic parameter specification
table. A value of zero-dot-zero indicates

best-effort treatment. By having the same value of this object, two or more segments can indicate resource sharing. This object cannot be modified if
gmplsOutSegmentRowStatus is active(1)."
::= { gmplsOutSegmentEntry 10 }

gmplsOutSegmentRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"For creating, modifying, and deleting this row."
::= { gmplsOutSegmentEntry 11 }

gmplsOutSegmentStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This variable indicates the storage type for this table entry. When set to 'permanent', the entire row is to be stored. This object cannot be modified if gmplsOutSegmentRowStatus is active(1). If this variable is set to readOnly(5), and the corresponding TFIB entry is removed, then the agent must remove this row shortly thereafter."
::= { gmplsOutSegmentEntry 12 }

gmplsOutSegmentDirection OBJECT-TYPE

SYNTAX GmplsSegmentDirection
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This variable indicates the direction of data flow on this segment. This object cannot be modified if
gmplsOutSegmentRowStatus is active(1)."
DEFVAL { forward }
::= { gmplsOutSegmentEntry 13 }

gmplsOutSegmentTTLDecrement OBJECT-TYPE

SYNTAX Unsigned32 (0..255)
MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This variable indicates the amount by which

Nadeau et al.

Expires July 2002

[Page 38]

to decrement the 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 force.

See the gmplsTunnelTTLDcrement object in the gmplsTunnelTable of [[GMPLSTEMIB](#)] for a value by which to decrement the TTL for the whole of a tunnel.

This object cannot be modified if
gmplsOutSegmentRowStatus is active(1)."
DEFVAL { 0 }
::= { gmplsOutSegmentEntry 14 }

-- End of gmplsOutSegmentTable

-- Out-segment performance table.

gmplsOutSegmentPerfTable OBJECT-TYPE
SYNTAX SEQUENCE OF GmplsOutSegmentPerfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains statistical information
about outgoing segments from an LSR. The
counters in this entry should behave in a
manner similar to that of the interface."
::= { gmplsLsrObjects 8 }

gmplsOutSegmentPerfEntry OBJECT-TYPE
SYNTAX GmplsOutSegmentPerfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"An entry in this table contains statistical
information about one outgoing segment
configured in gmplsOutSegmentTable."
AUGMENTS { gmplsOutSegmentEntry }
::= { gmplsOutSegmentPerfTable 1 }

GmplsOutSegmentPerfEntry ::= SEQUENCE {
gmplsOutSegmentOctets Counter32,
gmplsOutSegmentPackets Counter32,
gmplsOutSegmentErrors Counter32,
gmplsOutSegmentDiscards Counter32,

-- HC counter

Nadeau et al.

Expires July 2002

[Page 39]

```
        gmplsOutSegmentHCOctets          Counter64,

        gmplsOutSegmentPerfDiscontinuityTime  TimeStamp
    }
```

gmplsOutSegmentOctets OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This value contains the total number of
    octets sent on this segment."
::= { gmplsOutSegmentPerfEntry 1 }
```

gmplsOutSegmentPackets OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This value contains the total number of
    packets sent on this segment."
::= { gmplsOutSegmentPerfEntry 2 }
```

gmplsOutSegmentErrors OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of packets that could not be sent
    due to errors on this segment."
::= { gmplsOutSegmentPerfEntry 3 }
```

gmplsOutSegmentDiscards OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of labeled packets received on
    this out-segment, which were chosen to be
    discarded even though no errors had been
    detected to prevent their being
    transmitted. One possible reason for
    discarding such a labeled packet could be
    to free up buffer space."
::= { gmplsOutSegmentPerfEntry 4 }
```

gmplsOutSegmentHCOctets OBJECT-TYPE

SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current

DESCRIPTION

"Total number of octets sent. This is the 64 bit version of gmplsOutSegmentOctets."

::= { gmplsOutSegmentPerfEntry 5 }

gmplsOutSegmentPerfDiscontinuityTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime on the most recent occasion at which any one or more of this segment's Counter32 or Counter64 suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."

::= { gmplsOutSegmentPerfEntry 6 }

-- End of gmplsOutSegmentPerfTable.

-- Cross-connect table.

gmplsXCIndexNext OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an appropriate value to be used for gmplsXCIndex when creating entries in the gmplsXCTable. The value 0 indicates that no unassigned entries are available. To obtain the value of gmplsXCIndex for a new entry in the gmplsXCTable, the manager issues a management protocol retrieval operation to obtain the current value of gmplsXCIndex. After each retrieval operation, the agent should modify the value to reflect the next unassigned index. After a manager retrieves a value the agent will determine through its local policy when this index value will be made available for reuse."

::= { gmplsLsrObjects 9 }

gmplsXCTable	OBJECT-TYPE
SYNTAX	SEQUENCE OF GmplsXCEntry
MAX-ACCESS	not-accessible

Nadeau et al.

Expires July 2002

[Page 41]

STATUS current

DESCRIPTION

"This table specifies information for switching between LSP segments. It supports point-to-point, point-to-multipoint and multipoint-to-point connections. gmplsLabelStackTable specifies the label stack information for a cross-connect LSR and is referred to from gmplsXCTable."

::= { gmplsLsrObjects 10 }

gmplsXCEntry OBJECT-TYPE

SYNTAX GmplsXCEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A row in this table represents one cross-connect entry. The following objects index it:

- cross-connect index gmplsXCIndex that uniquely identifies a group of cross-connect entries
- in-segment index, gmplsInSegmentIndex
- out-segment index, gmplsOutSegmentIndex

Originating LSPs:

These are represented by using the special value gmplsInSegmentIndex=0. In this case the gmplsOutSegmentIndex MUST be non-zero.

Terminating LSPs:

These are represented by using the special value gmplsOutSegmentIndex=0 as index. In this case the gmplsInSegmentIndex MUST be non-zero.

Bidirectional LSPs:

These are represented in the normal way. Objects within the component links indicate the direction of data flow on those links.

Special labels:

Entries indexed by reserved MPLS label values 0 through 15 imply terminating LSPs and MUST have gmplsOutSegmentIndex = 0. Note that situations where LSPs are

terminated with incoming label equal to 0,
should have `gmplsInSegmentIfIndex = 0` as
well, but can be distinguished from

originating LSPs because the
gmplsOutSegmentIfIndex = 0. The
gmplsOutSegmentIfIndex MUST only be set to
0 in cases of terminating LSPs.

An entry can be created by a network
administrator by an SNMP agent as
instructed by an MPLS signaling protocol."

```
INDEX { gmplsXCIndex,  
        gmplsInSegmentIndex,  
        gmplsOutSegmentIndex }  
::= { gmplsXCTable 1 }
```

```
GmplsXCEnter ::= SEQUENCE {  
    gmplsXCIndex      Unsigned32,  
    gmplsXCLspId      MplsLSPID,  
    gmplsXCLabelStackIndex Unsigned32,  
    gmplsXCIsPersistent TruthValue,  
    gmplsXCOwner      MplsInitialCreationSource,  
    gmplsXCRowStatus   RowStatus,  
    gmplsXCStorageType StorageType,  
    gmplsXCAdminStatus INTEGER,  
    gmplsXCOperStatus  INTEGER  
}
```

gmplsXCIndex OBJECT-TYPE

```
SYNTAX      Unsigned32 (0..4294967295)  
MAX-ACCESS  not-accessible  
STATUS      current  
DESCRIPTION  
    "Primary index for the conceptual row  
    identifying a group of cross-connect  
    segments."  
::= { gmplsXCEnter 1 }
```

gmplsXCLspId OBJECT-TYPE

```
SYNTAX      MplsLSPID  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "This value identifies the label switched  
    path that this cross-connect entry belongs  
    to. This object cannot be modified if  
    mplsXCRowStatus is active(1)."  
::= { gmplsXCEnter 2 }
```

gmplsXCLabelStackIndex OBJECT-TYPE

SYNTAX	Unsigned32 (0..4294967295)
MAX-ACCESS	read-create
STATUS	current

DESCRIPTION

"Primary index into gmplsLabelStackTable identifying a stack of labels to be pushed beneath the top label. Note that the top label identified by the out-segment ensures that all the components of a multipoint-to-point connection have the same outgoing label. A value of 0 indicates that no labels are to be stacked beneath the top label. This object cannot be modified if mplsXCRowStatus is active(1)."

::= { gmplsXCEntiry 3 }

gmplsXCIsPersistent OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Denotes whether or not this cross-connect entry and associated in- and out-segments should be restored automatically after failures. This value MUST be set to false in cases where this cross-connect entry was created by a signaling protocol. This object cannot be modified if mplsXCRowStatus is active(1)."

DEFVAL { false }

::= { gmplsXCEntiry 4 }

gmplsXCOwner OBJECT-TYPE

SYNTAX MplsInitialCreationSource
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Denotes the entity that created and is responsible for managing this cross-connect. This object cannot be modified if mplsXCRowStatus is active(1)."

::= { gmplsXCEntiry 5 }

gmplsXCRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"For creating, modifying, and deleting this row."

```
::= { gmpIsXCEntry 6 }
```

```
gmpIsXCStorageType OBJECT-TYPE
```



```
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"Defines the storage type for this table entry. When set to 'permanent', the entire row is to be stored. This object cannot be modified if mplsxRowStatus is active(1). If this variable is set to readOnly(5), and the corresponding TFIB entry is removed, then the agent must remove this row shortly thereafter."

```
::= { gmplsXCEntry 7 }
```

gmplsXCAdminStatus OBJECT-TYPE

```
SYNTAX      INTEGER {
    up(1),      -- ready to pass packets
    down(2),
    testing(3)  -- in some test mode
}
```

```
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"The desired operational status of this segment."

```
::= { gmplsXCEntry 8 }
```

gmplsXCOperStatus OBJECT-TYPE

```
SYNTAX      INTEGER {
    up(1),      -- ready to pass packets
    down(2),
    testing(3), -- in some test mode
    unknown(4), -- status cannot be determined
                -- for some reason.
    dormant(5),
    notPresent(6), -- some component is missing
    lowerLayerDown(7) -- down due to the state of
                    -- lower layer interfaces
}
```

```
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The actual operational status of this cross-connect."

```
::= { gmplsXCEntry 9 }
```

```
-- End of gmplsXCTable
```

-- Label stack table.

Nadeau et al.

Expires July 2002

[Page 45]

gmplsMaxLabelStackDepth OBJECT-TYPE

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum stack depth supported by this LSR."

::= { gmplsLsrObjects 11 }

gmplsLabelStackIndexNext OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an appropriate value to be used for gmplsLabelStackIndex when creating entries in the gmplsLabelStackTable. The value 0 indicates that no unassigned entries are available. To obtain an gmplsLabelStackIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval operation, the agent should modify the value to reflect the next unassigned index. After a manager retrieves a value the agent will determine through its local policy when this index value will be made available for reuse."

::= { gmplsLsrObjects 12 }

gmplsLabelStackTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsLabelStackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table specifies the label stack to be pushed onto a packet, beneath the top label. Entries into this table are referred to from gmplsXCTable."

::= { gmplsLsrObjects 13 }

gmplsLabelStackEntry OBJECT-TYPE

SYNTAX GmplsLabelStackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents one label
which is top label. An entry can be

Nadeau et al.

Expires July 2002

[Page 46]

created by a network administrator or by an
SNMP agent as instructed by an MPLS
signaling protocol."

INDEX { gmplsLabelStackIndex,
gmplsLabelStackLabelIndex }
::= { gmplsLabelStackTable 1 }

GmplsLabelStackEntry ::= SEQUENCE {
 gmplsLabelStackIndex Unsigned32,
 gmplsLabelStackLabelIndex Unsigned32,
 gmplsLabelStackLabel Unsigned32,
 gmplsLabelStackRowStatus RowStatus,
 gmplsLabelStackStorageType StorageType
}

gmplsLabelStackIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Primary index for this row identifying a
 stack of labels to be pushed on an outgoing
 packet, beneath the top label."
::= { gmplsLabelStackEntry 1 }

gmplsLabelStackLabelIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Secondary index for this row identifying
 one label of the stack. Note that an entry
 with a smaller gmplsLabelStackLabelIndex
 would refer to a label higher up the label
 stack and would be popped at a downstream
 LSR before a label represented by a higher
 gmplsLabelStackLabelIndex at a downstream
LSR."
::= { gmplsLabelStackEntry 2 }

gmplsLabelStackLabel OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
 "The label to be pushed. This object
 contains the index into the gmplsLabelTable

of the entry that contains the label value
to apply. Note that in implementations
where the label may be encoded within a 32

bit integer and where gmplsLabelTable is not implemented, this object may directly contain the label value to use."

::= { gmplsLabelStackEntry 3 }

gmplsLabelStackRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"For creating, modifying, and deleting this row."

::= { gmplsLabelStackEntry 4 }

gmplsLabelStackStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Defines the storage type for this table entry. When set to 'permanent', the entire row is to be stored. If this variable is set to readOnly(5), and the corresponding TFIB entry is removed, then the agent must remove this row shortly thereafter."

::= { gmplsLabelStackEntry 5 }

-- End of gmplsLabelStackTable

-- Traffic Parameter table.

gmplsTrafficParamIndexNext OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an appropriate value which will be used for gmplsTrafficParamIndex when creating entries in the gmplsTrafficParamTable. The value 0 indicates that no unassigned entries are available. To obtain the gmplsTrafficParamIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval operation, the agent should

modify the value to reflect the next
unassigned index. After a manager
retrieves a value the agent will determine

through its local policy when this index
value will be made available for reuse."
 ::= { gmplsLsrObjects 14 }

gmplsTrafficParamTable OBJECT-TYPE

SYNTAX SEQUENCE OF GmplsTrafficParamEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "This table specifies the Traffic Parameter
 objects for in and out-segments."
 ::= { gmplsLsrObjects 15 }

gmplsTrafficParamEntry OBJECT-TYPE

SYNTAX GmplsTrafficParamEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "An entry in this table represents the
 TrafficParam objects for one or more in or
 out segments. A single entry can be
 pointed to by multiple segments indicating
 resource sharing."
INDEX { gmplsTrafficParamIndex }
 ::= { gmplsTrafficParamTable 1 }

GmplsTrafficParamEntry ::= SEQUENCE {

gmplsTrafficParamIndex	Unsigned32,
gmplsTrafficParamMaxRate	MplsBitRate,
gmplsTrafficParamMeanRate	MplsBitRate,
gmplsTrafficParamMaxBurstSize	MplsBurstSize,
gmplsTrafficParamRowStatus	RowStatus,
gmplsTrafficParamStorageType	StorageType

}

gmplsTrafficParamIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Uniquely identifies this row of the table.
 Note that zero represents an invalid
 index."
 ::= { gmplsTrafficParamEntry 1 }

gmplsTrafficParamMaxRate OBJECT-TYPE

SYNTAX MplsBitRate

UNITS	"kilobits per second"
MAX-ACCESS	read-create
STATUS	current

DESCRIPTION

"Maximum rate in kilobits/second."
::= { gmplsTrafficParamEntry 2 }

gmplsTrafficParamMeanRate OBJECT-TYPE

SYNTAX MplsBitRate
UNITS "kilobits per second"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Mean rate in kilobits/second."
::= { gmplsTrafficParamEntry 3 }

gmplsTrafficParamMaxBurstSize OBJECT-TYPE

SYNTAX MplsBurstSize
UNITS "bytes"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Maximum burst size in bytes."
::= { gmplsTrafficParamEntry 4 }

gmplsTrafficParamRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"For creating, modifying, and deleting this
 row."
::= { gmplsTrafficParamEntry 5 }

gmplsTrafficParamStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The storage type for this table entry. When
 set to 'permanent', the entire row is to be
 stored. If this variable is set to
 readOnly(5), and the corresponding TFIB
 entry is removed, then the agent must
 remove this row shortly thereafter."
::= { gmplsTrafficParamEntry 6 }

-- End of gmplsTrafficParamTable

-- Notification Configuration

gmplsXCTrapEnable OBJECT-TYPE
SYNTAX GmplsTrapEnable

Nadeau et al.

Expires July 2002

[Page 50]

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object controls the generation of gmplsXCUp and gmplsXCDown notifications. If set to disabled (0), these notifications are not emitted. If set to oneAtATime (1), each notification may only carry information about a single XC that has changed state. If set to rangeAllowed (2), each notification MAY carry information about a contiguous range of XCs that have changed state, but note that implementations may send multiple individual notifications even when rangeAllowed is selected."

DEFVAL { disabled }

::= { gmplsLsrObjects 16 }

-- Cross-connect.

gmplsXCUp NOTIFICATION-TYPE

OBJECTS {
 gmplsXCOperStatus, -- start of range
 gmplsXCOperStatus -- end of range
}

STATUS current

DESCRIPTION

"This notification is generated when the gmplsXCOperStatus object for one or more contiguous entries in gmplsXCTable are about to enter the up(1) state from some other. The included values of gmplsXCOperStatus MUST both be set equal to this other state. The two instances of gmplsXCOperStatus in this notification indicate the range of indexes that are affected. Note that all the indexes of the two ends of the range can be derived from the instance identifiers of these two objects. For cases where a contiguous range of cross-connects have transitioned into the up(1) state at roughly the same time, the device SHOULD issue a single notification for each range of contiguous indexes in an effort to minimize the emission of a large number of

notifications. If a notification has to be issued for just a single cross-connect entry, then the instance identifier (and

```
        values) of the two gmplsXCOperStatus
        objects MUST be the identical."
 ::= { gmplsLsrNotifyPrefix 1 }
```

gmplsXCDown NOTIFICATION-TYPE

```
OBJECTS      {
    gmplsXCOperStatus,  -- start of range
    gmplsXCOperStatus   -- end of range
}
```

```
STATUS      current
```

DESCRIPTION

"This notification is generated when the gmplsXCOperStatus object for one or more contiguous entries in gmplsXCTable are about to enter the down(2) state from some other. The included values of gmplsXCOperStatus MUST both be set equal to this other state. The two instances of gmplsXCOperStatus in this notification indicate the range of indexes that are affected. Note that all the indexes of the two ends of the range can be derived from the instance identifiers of these two objects. For cases where a contiguous range of cross-connects have transitioned into the down(2) state at roughly the same time, the device SHOULD issue a single notification for each range of contiguous indexes in an effort to minimize the emission of a large number of notifications. If a notification has to be issued for just a single cross-connect entry, then the instance identifier (and values) of the two gmplsXCOperStatus objects MUST be the identical."

```
 ::= { gmplsLsrNotifyPrefix 2 }
```

```
-- End of notifications.
```

```
-- Module compliance.
```

gmplsLsrGroups

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

gmplsLsrCompliances

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

gmplsLsrModuleCompliance MODULE-COMPLIANCE
STATUS current

Nadeau et al.

Expires July 2002

[Page 52]

DESCRIPTION

"Compliance statement for agents that support the MPLS LSR MIB."

MODULE -- this module

-- The mandatory groups have to be implemented
-- by all LSRs. However, they may all be supported
-- as read-only objects in the case where manual
-- configuration is unsupported.

```
MANDATORY-GROUPS    {  
    gmplsInSegmentGroup,  
    gmplsOutSegmentGroup,  
    gmplsXCGroup,  
    gmplsInterfaceGroup,  
    gmplsPerfGroup,  
    gmplsSegmentDiscontinuityGroup  
}
```

GROUP gmplsHCInSegmentPerfGroup

DESCRIPTION

"This group is mandatory for those in-segment entries for which the object gmplsInSegmentOutOctets wraps around too quickly."

GROUP gmplsHCOutSegmentPerfGroup

DESCRIPTION

"This group is mandatory for those out-segment entries for which the object gmplsOutSegmentOctets wraps around too quickly."

GROUP gmplsTrafficParamGroup

DESCRIPTION

"This group is mandatory for those LSRs that support QoS resource reservation."

-- Depending on whether the device implements
-- persistent cross-connects or not one of the
-- following two groups is mandatory.

GROUP gmplsXCIsPersistentGroup

DESCRIPTION

"This group is mandatory for devices which support persistent cross-connects. The following constraints apply:

gmplsXCIsPersistent must at least be read-only returning true(2)."

```
GROUP gmplsXCIsNotPersistentGroup
DESCRIPTION
    "This group is mandatory for devices which
    support non-persistent cross-connects. The
    following constraints apply:
    gmplsXCIsPersistent must at least be read-
    only returning false(1)."
```

-- gmplsInSegmentTable

```
OBJECT      gmplsInSegmentXCIndex
DESCRIPTION
    "Write access is not required."
```

```
OBJECT      gmplsInSegmentNPop
MIN-ACCESS  read-only
DESCRIPTION
    "Write access is not required. This object
    should be set to 1 if it is read-only."
```

```
OBJECT      gmplsInSegmentAddrFamily
DESCRIPTION
    "Write access is not required. A value of
    other(0) should be supported because there
    may be cases where the agent may not know
    about or support any address types."
```

```
OBJECT      gmplsInSegmentStorageType
SYNTAX      INTEGER { other(1) }
MIN-ACCESS  read-only
DESCRIPTION
    "Only other(1) needs to be supported."
```

```
OBJECT      gmplsInSegmentDirection
SYNTAX      GmplsSegmentDirection
MIN-ACCESS  read-only
DESCRIPTION
    "Only forward(1) needs to be supported."
```

-- gmplsOutSegmentTable

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

OBJECT gmpIsOutSegmentPushTopLabel
MIN-ACCESS read-only

Nadeau et al.

Expires July 2002

[Page 54]

DESCRIPTION

"Write access is not required."

OBJECT gmplsOutSegmentTopLabel

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT gmplsOutSegmentNextHopIpAddrType

MIN-ACCESS read-only

DESCRIPTION

"ipV6(3) need not be supported."

OBJECT gmplsOutSegmentNextHopIpv4Addr

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT gmplsOutSegmentNextHopIpv6Addr

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT gmplsOutSegmentRowStatus

SYNTAX INTEGER {

active(1),

notInService(2),createAndGo(4),

destroy(6)

}

MIN-ACCESS read-only

DESCRIPTION

"The notReady(3) and createAndWait(5) states
need not be supported."

OBJECT gmplsOutSegmentStorageType

SYNTAX INTEGER { other(1) }

MIN-ACCESS read-only

DESCRIPTION

"Only other(1) needs to be supported."

OBJECT gmplsOutSegmentDirection

SYNTAX GmplsSegmentDirection

MIN-ACCESS read-only

DESCRIPTION

"Only forward(1) needs to be supported."

OBJECT gmplsOutSegmentTTLDcrement

SYNTAX Unsigned32 (0..255)

MIN-ACCESS read-only
DESCRIPTION

Nadeau et al.

Expires July 2002

[Page 55]

"Write access is not required."

-- gmplsXCTable

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

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

OBJECT gmplsXCAdminStatus
SYNTAX INTEGER { up(1), down(2) }
MIN-ACCESS read-only
DESCRIPTION
"A value of testing(3) need not be supported."

OBJECT gmplsXCOperStatus
SYNTAX INTEGER { up(1), down(2) }
DESCRIPTION
"Only up(1) and down(2) need to be supported."

OBJECT gmplsXCRowStatus
SYNTAX INTEGER {
active(1),
notInService(2),
createAndGo(4),
destroy(6)
}
MIN-ACCESS read-only
DESCRIPTION
"The notReady(3) and createAndWait(5) states
need not be supported."

OBJECT gmplsXCStorageType
SYNTAX INTEGER { other(1) }
MIN-ACCESS read-only
DESCRIPTION
"Only other(1) needs to be supported."

::= { gmplsLsrCompliances 1 }

-- Units of conformance.

Nadeau et al.

Expires July 2002

[Page 56]

gmplsInterfaceGroup OBJECT-GROUP

```
OBJECTS {
    gmplsInterfaceLabelMinIn,
    gmplsInterfaceLabelMaxIn,
    gmplsInterfaceLabelMinOut,
    gmplsInterfaceLabelMaxOut,
    gmplsInterfaceTotalBandwidth,
    gmplsInterfaceAvailableBandwidth,
    gmplsInterfaceLabelParticipationType,
    gmplsInterfaceConfSignalingCaps
}
STATUS current
DESCRIPTION
    "Collection of objects needed for MPLS
    interface configuration and performance
    information."
::= { gmplsLsrGroups 1 }
```

gmplsInSegmentGroup OBJECT-GROUP

```
OBJECTS {
    gmplsInSegmentIndexNext,
    gmplsInSegmentIfIndex,
    gmplsInSegmentLabel,
    gmplsInSegmentNPop,
    gmplsInSegmentAddrFamily,
    gmplsInSegmentXCIndex,
    gmplsInSegmentOwner,
    gmplsInSegmentTrafficParamPtr,
    gmplsInSegmentRowStatus,
    gmplsInSegmentStorageType,
    gmplsInSegmentDirection
}
STATUS current
DESCRIPTION
    "Collection of objects needed to implement
    an in-segment."
::= { gmplsLsrGroups 2 }
```

gmplsOutSegmentGroup OBJECT-GROUP

```
OBJECTS {
    gmplsOutSegmentIndexNext,
    gmplsOutSegmentIfIndex,
    gmplsOutSegmentPushTopLabel,
    gmplsOutSegmentTopLabel,
    gmplsOutSegmentNextHopIpAddrType,
    gmplsOutSegmentNextHopIpv4Addr,
    gmplsOutSegmentNextHopIpv6Addr,

```

```
gmplsOutSegmentXCIndex,  
gmplsOutSegmentOwner,  
gmplsOutSegmentTrafficParamPtr,
```

```
    gmplsOutSegmentRowStatus,  
    gmplsOutSegmentStorageType,  
    gmplsOutSegmentDirection,  
    gmplsOutSegmentTTLDecrement  
}
```

STATUS current

DESCRIPTION

"Collection of objects needed to implement
an out-segment."

::= { gmplsLsrGroups 3 }

gmplsXCGroup OBJECT-GROUP

OBJECTS {

```
    gmplsXCIndexNext,  
    gmplsXCLabelStackIndex,  
    gmplsXCIsPersistent,  
    gmplsXCOwner,  
    gmplsXCRowStatus,  
    gmplsXCAdminStatus,  
    gmplsXCStorageType,  
    gmplsXCOperStatus,  
    gmplsXCTrapEnable
```

}

STATUS current

DESCRIPTION

"Collection of objects needed to implement a
cross-connect entry."

::= { gmplsLsrGroups 4 }

gmplsXCOptionalGroup OBJECT-GROUP

OBJECTS { gmplsXCLspId }

STATUS current

DESCRIPTION

"Collection of optional objects for
implementing a cross-connect entry."

::= { gmplsLsrGroups 5 }

gmplsPerfGroup OBJECT-GROUP

OBJECTS {

```
    gmplsInSegmentOctets,  
    gmplsInSegmentPackets,  
    gmplsInSegmentErrors,  
    gmplsInSegmentDiscards,  
    gmplsOutSegmentOctets,  
    gmplsOutSegmentPackets,  
    gmplsOutSegmentErrors,  
    gmplsOutSegmentDiscards,
```

```
gmplsInterfaceInLabelsUsed,  
gmplsInterfaceFailedLabelLookup,  
gmplsInterfaceOutFragments,
```

```
        gmplsInterfaceOutLabelsUsed
    }
    STATUS current
    DESCRIPTION
        "Collection of objects providing performance
        information about an LSR."
    ::= { gmplsLsrGroups 6 }

gmplsHCInSegmentPerfGroup OBJECT-GROUP
    OBJECTS { gmplsInSegmentHCOctets }
    STATUS current
    DESCRIPTION
        "Object(s) providing performance information
        specific to out-segments for which the
        object gmplsInterfaceInOctets wraps around
        too quickly."
    ::= { gmplsLsrGroups 7 }

gmplsHCOutSegmentPerfGroup OBJECT-GROUP
    OBJECTS { gmplsOutSegmentHCOctets }
    STATUS current
    DESCRIPTION
        "Object(s) providing performance information
        specific to out-segments for which the
        object gmplsInterfaceOutOctets wraps around
        too quickly."
    ::= { gmplsLsrGroups 8 }

gmplsXCIsPersistentGroup OBJECT-GROUP
    OBJECTS { gmplsXCIsPersistent }
    STATUS current
    DESCRIPTION
        "Objects needed to support persistent cross-
        connects."
    ::= { gmplsLsrGroups 9 }

gmplsXCIsNotPersistentGroup OBJECT-GROUP
    OBJECTS { gmplsXCIsPersistent }
    STATUS current
    DESCRIPTION
        "Objects needed to support non-persistent
        cross-connects."
    ::= { gmplsLsrGroups 10 }

gmplsLabelStackGroup OBJECT-GROUP
    OBJECTS {
        gmplsLabelStackIndexNext,
```

gmplsLabelStackLabel,
gmplsLabelStackRowStatus,
gmplsLabelStackStorageType,

```
        gmplsMaxLabelStackDepth
    }
    STATUS current
    DESCRIPTION
        "Objects needed to support label stacking."
    ::= { gmplsLsrGroups 11 }

gmplsTrafficParamGroup OBJECT-GROUP
    OBJECTS {
        gmplsTrafficParamIndexNext,
        gmplsTrafficParamMaxRate,
        gmplsTrafficParamMeanRate,
        gmplsTrafficParamMaxBurstSize,
        gmplsTrafficParamRowStatus,
        gmplsTrafficParamStorageType
    }
    STATUS current
    DESCRIPTION
        "Object(s) required for supporting QoS
        resource reservation."
    ::= { gmplsLsrGroups 12 }

gmplsSegmentDiscontinuityGroup OBJECT-GROUP
    OBJECTS {
        gmplsInSegmentPerfDiscontinuityTime,
        gmplsOutSegmentPerfDiscontinuityTime
    }
    STATUS current
    DESCRIPTION
        "A collection of objects providing
        information specific to segment
        discontinuities.."
    ::= { gmplsLsrGroups 13 }

gmplsLsrNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        gmplsXCUp,
        gmplsXCDown
    }
    STATUS current
    DESCRIPTION
        "Set of notifications implemented in this
        module.  None is mandatory."
    ::= { gmplsLsrGroups 14 }
```

END

13. Security Considerations

Nadeau et al.

Expires July 2002

[Page 60]

It is clear that this MIB is potentially useful for monitoring of MPLS and GMPLS LSRs. This MIB 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 this MIB that have 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.

There are a number of managed objects in this MIB that may contain information that may be sensitive from a business perspective, in that they represent a customer's interface to the MPLS network. Allowing uncontrolled access to these objects could result in malicious and unwanted disruptions of network traffic or incorrect configurations for these customers. There are no objects that are particularly sensitive in their own right, such as passwords or monetary amounts.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

At this writing, no security holes have been identified beyond those that SNMP Security [[RFC2571](#)] is itself intended to address. These relate to primarily controlled access to sensitive information and the ability to configure a device - or which might result from operator error, which is beyond the scope of any security architecture.

SNMPv1 or SNMPv2 are by themselves not a secure environment. Even if the network itself is secure (for example by using IPsec [[RFC2401](#)]), there is no control as to who on the secure network is allowed to access and GET/SET (read, change, create and/or delete) the objects in this MIB. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [[RFC2574](#)] and the View-based Access

Control [[RFC2575](#)] is recommended. It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB 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.

14. References

14.1. Normative References

- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", [RFC 1157](#), May 1990.
- [RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions", [RFC 1212](#), March 1991.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, [RFC 2579](#), April 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholtz, "The Interfaces Group MIB", [RFC 2863](#), June 2000.
- [RFC3032] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Federokow, G., Li, T., and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), January 2001.
- [RFC3036] Anderson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", [RFC 3036](#), January 2001.
- [RSVPTE] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", [draft-ietf-mpls-rsvp-lsp-tunnel-08.txt](#), February 2001, work in progress.
- [CRLDP] Jamoussi, B., Aboul-Magd, O., Andersson, L., Ashwood-Smith, P., Hellstrand, F., Sundell, K., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N.,

Fredette, A., Girish, M., Gray, E.,
Halpern, J., Heinanen, J., Kilty, T.,

Nadeau et al.

Expires July 2002

[Page 62]

Malis, A., and P. Vaananen, "Constraint-Based LSP Setup using LDP", [draft-ietf-mpls-cr-ldp-05.txt](#), February 2001, work in progress."

- [GMPLSArch] Ashwood-Smith, P., Awduche, D., Banerjee, A., Basak, D., Berger, L., Bernstein, G., Drake, J., Fan, Y., Fedyk, D., Grammel, D., Kompella, K., Kullberg, A., Lang, J., Liaw, F., Papadimitriou, D., Pendarakis, D., Rajagopalan, B., Rekhter, Y., Saha, D., Sandick, H., Sharma, V., Swallow, G., Tang, Z., Yu, J., Zinin, A., Nadeau, T., Mannie, E., Generalized Multiprotocol Label Switching (GMPLS) Architecture, Internet Draft <[draft-many-gmpls-architecture-01.txt](#)>, March 2001, work in progress.
- [GMPLSSig] Ashwood-Smith, P., Awduche, D., Banerjee, A., Basak, D., Berger, L., Bernstein, G., Drake, J., Fan, Y., Fedyk, D., Grammel, D., Kompella, K., Kullberg, A., Lang, Rajagopalan, B., Rekhter, Y., Saha, D., Sharma, V., Swallow, G., Bo Tang, Z., Generalized MPLS - Signaling Functional Description, <[draft-ietf-mpls-generalized-signaling-04.txt](#)>, May 2001, work in progress.
- [GMPLSCRLDP] Ashwood-Smith, P., Awduche, D., Banerjee, A., Basak, D., Berger, L., Bernstein, G., Drake, J., Fan, Y., Fedyk, D., Grammel, D., Kompella, K., Kullberg, A., Lang, Rajagopalan, B., Rekhter, Y., Saha, D., Sharma, V., Swallow, G., Bo Tang, Z., Generalized MPLS Signaling - CR-LDP Extensions, Internet Draft <[draft-ietf-mpls-generalized-cr-ldp-03.txt](#)>, May 2001, work in progress.
- [GMPLSRVPTE] Ashwood-Smith, P., Awduche, D., Banerjee, A., Basak, D., Berger, L., Bernstein, G., Drake, J., Fan, Y., Fedyk, D., Grammel, D., Kompella, K., Kullberg, A., Lang, Rajagopalan, B., Rekhter, Y., Saha, D., Sharma, V., Swallow, G., Bo Tang, Z., Generalized MPLS Signaling - RSVP-TE

Extensions, Internet Draft <[draft-ietf-mpls-generalized-rsvp-te-03.txt](#)>, May 2001, work in progress.

- [GMPLSSonetSDH] Mannie, E., Ansorge, S., Ashwood-Smith, P., Banerjee, A., Berger, L., Bernstein, G., Chiu, A., Drake, J., Fan, Y., Fontana, M., Grammel, G., Heiles, J., Katukam, S., Kompella, K., Lang, J. P., Liaw, F., Lin, Z., Mack-Crane, B., Papadimitriou, D., Pendarakis, D., Raftelis, M., Rajagopalan, B., Rekhter, Y., Saha, D., Sharma, V., Swallow, G., Bo Tang, Z., Varma, E., Visser, M., Xu, Y., GMPLS Extensions for SONET and SDH Control, Internet Draft <[draft-ietf-ccamp-gmpls-sonet-sdh-00.txt](#)>, May 2001, work in progress.
- [TCMIB] Nadeau, T., Cucchiara, J., Srinivasan, C., Viswanathan, A. and H. Sjostrand, "Definition of Textual Conventions and OBJECT-IDENTITIES for Multiprotocol Label Switching (MPLS) Management", Internet Draft <[draft-ietf-mpls-tc-mib-03.txt](#)>, January 2002, work in progress.
- [LSRMIB] Srinivasan, C., Viswanathan, A. and T. Nadeau, "MPLS Label Switching Router Management Information Base Using SMIV2", Internet Draft <[draft-ietf-mpls-lsr-mib-08.txt](#)>, January 2002, work in progress.
- [LABELMIB] Nadeau, T., Srinivasan, C., A., Farrel, A., Hall, T., and Harrison, E., "Generalized Multiprotocol Label Switching (GMPLS) Label Management Information Base", [draft-nadeau-ccamp-gmpls-label-mib-01.txt](#), January 2002, work in progress.
- [GMPLSTEMIB] Nadeau, T., Srinivasan, C., Farrel, A., Hall, T., and Harrison, E., "Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base", [draft-nadeau-ccamp-gmpls-te-mib-01.txt](#), January 2002, work in progress.

14.2. Informational References

- [RFC1155] Rose, M., and K. McCloghrie, "Structure and

Identification of Management Information
for TCP/IP-based Internets", [RFC 1155](#), May

Nadeau et al.

Expires July 2002

[Page 64]

1990.

- [RFC1215] M. Rose, "A Convention for Defining Traps for use with the SNMP", [RFC 1215](#), March 1991.
- [RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", [RFC 1901](#), January 1996.
- [RFC1905] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1905](#), January 1996.
- [RFC1906] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1906](#), January 1996.
- [RFC2514] Noto, et. al., "Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management", [RFC 2514](#), Feb. 1999
- [RFC2515] K. Tesink, "Definitions of Managed Objects for ATM Management", [RFC 2515](#), Feb. 1999
- [RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", [RFC 2570](#), April 1999.
- [RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2571](#), April 1999.
- [RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", [RFC 2572](#), April 1999.
- [RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", [RFC 2573](#), April 1999.
- [RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the

Simple Network Management Protocol
(SNMPv3)", [RFC 2574](#), April 1999.

Nadeau et al.

Expires July 2002

[Page 65]

- [RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", [RFC 2575](#), April 1999.
- [RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, [RFC 2580](#), April 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", [RFC 3031](#), January 2001.
- [RFC3034] Conta, A., Doolan, P., Malis, A., "Use of Label Switching on Frame Relay Networks Specification", [RFC 3034](#), January 2001.
- [RFC3035] Davie, B., Lawrence, J., McCloghrie, K., Rosen, E., Swallow, G., Rekhter, Y., and P. Doolan, "MPLS using LDP and ATM VC switching", [RFC 3035](#), January 2001.
- [IANAFamily] Internet Assigned Numbers Authority (IANA), ADDRESS FAMILY NUMBERS.
- [TEMIB] Nadeau, T., Srinivasan, C, Viswanathan, A., "Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base", Internet Draft <[draft-ietf-mpls-te-mib-08.txt](#)>, January 2002, work in progress.

[15.](#) Authors' Addresses

Thomas D. Nadeau
Cisco Systems, Inc.
[300](#) Apollo Drive
Chelmsford, MA 01824

Phone: +1-978-244-3051
Email: tnadeau@cisco.com

Nadeau et al.

Expires July 2002

[Page 66]

Cheenu Srinivasan
Parama Networks, Inc.
1030 Broad Street
Shrewsbury, NJ 07702
Phone: +1-732-544-9120 x731
Email: cheenu@paramanet.com

Adrian Farrel
Movaz Networks, Inc.
7926 Jones Branch Drive, Suite 615
McLean VA, 22102 USA
Phone: +1-703-847-9847
Email: afarrel@movaz.com

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

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

16. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of

developing Internet standards in which case the
procedures for copyrights defined in the Internet

Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

