

Network Working Group
Internet Draft
Expires: December 1999

Cheenu Srinivasan
Tachion Network Technologies

Arun Viswanathan
Lucent Technologies

MPLS Label Switch Router Management Information Base Using SMiv2

[draft-ietf-mpls-lsr-mib-00.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of 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 an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for modeling an Multi-Protocol Label Switching (MPLS) [[MPLSArch](#), [MPLSFW](#)] Label Switch Router (LSR).

Open Issues

- Does `mplsTSpecTable` belong in this document?

- Support for signalled COS value.
- Do we need objects to keep track of ownership of entries in various tables?
- More descriptive text and detailed example.
- Reconcile mplsInterfaceConfTable with the interface related objects in the LDP MIB [[LDPMIB](#)] including objects for negotiated label ranges.
- Address multipath related issues.
- Introduce a scalar to indicate the maximum supported label stack depth.
- For each MPLS capable interface, we need objects in mplsInterfaceConfTable that indicate the resource availability for MPLS, such as total bandwidth, available bandwidth for each priority level, available buffer etc.
- Fragmentation counter in mplsInterfacePerfTable.

1. Introduction

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for modeling an Multi-Protocol Label Switching (MPLS) [[MPLSArch](#), [MPLSFW](#)] Label Switch Router (LSR).

Comments should be made directly to the MPLS mailing list at mpls@uu.net.

This memo does not, in its draft form, specify a standard for the Internet community.

2. Terminology

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

A label switched path (LSP) is modelled as a connection consisting of one or more in-segments and/or one or more out-segments at a label switch router (LSR). The binding or interconnection between

in-segments and out-segments in performed using a cross-connect.
We use the terms connection and LSP interchangeably where the

meaning is clear from the context.

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in [RFC 2271](#) [[SNMPArch](#)].
- 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 [RFC 1155](#) [[SMIV1](#)], [RFC 1212](#) [[SNMPv1MIBDef](#)] and [RFC 1215](#) [[SNMPv1Traps](#)]. The second version, called SMIV2, is described in [RFC 1902](#) [[SMIV2](#)], [RFC 1903](#) [[SNMPv2TC](#)] and [RFC 1904](#) [[SNMPv2Conf](#)].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in [RFC 1157](#) [[SNMPv1](#)]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901](#) [[SNMPv2c](#)] and [RFC 1906](#) [[SNMPv2TM](#)]. The third version of the message protocol is called SNMPv3 and described in [RFC 1906](#) [[SNMPv2TM](#)], [RFC 2272](#) [[SNMPv3MP](#)] and [RFC 2274](#) [[SNMPv3USM](#)].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in [RFC 1157](#) [[SNMPv1](#)]. A second set of protocol operations and associated PDU formats is described in [RFC 1905](#) [[SNMPv2PO](#)].
- A set of fundamental applications described in [RFC 2273](#) [[SNMPv3App](#)] and the view-based access control mechanism described in [RFC 2275](#) [[SNMPv3VACM](#)]. 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.

3.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to also refer to the object type.

4. Feature Checklist

The MPLS label switch router (LSR) MIB is designed to satisfy the following requirements and constraints.

- The MIB should be able to support both manually configured LSPs as well as via LDP and/or RSVP signaling.
- The MIB must support enabling and disabling of MPLS capability on MPLS capable interfaces of an LSR.
- The MIB should allow resource sharing between two or more LSPs.
- Both per-platform and per-interface label spaces must be supported.
- MPLS packets must be forwarded solely based on an incoming top label [[MPLSArch](#), [LblStk](#)].
- Support must be provided for next-hop resolution when the outgoing interface is a shared media interface. In the point-to-multipoint case, each outgoing segment can be on a different shared media interface.
- The MIB must support point-to-point, point-to-multipoint and multipoint-to-point connections at an LSR.
- For multipoint-to-point connections all the outgoing packets must have the same top label.
- For multipoint-to-point connections the outgoing resources of the merged connections must be shared.
- For multipoint-to-point connections, packets from different incoming connections may 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.
- In a point-to-multipoint connection the ingress resources are shared by all the members of the connection.
- The MIB must provide cross-connect capability to "pop" an incoming label and forward the packet with the rest of the label stack unchanged and without pushing any labels ("pop-and-go") [[LblStk](#)].
- It must be possible to assign or remap COS bits [[LblStk](#)] on the outgoing label. In the multipoint-to-point case, each in-segment can have a different outgoing COS value. In the point-to-multipoint case, each out-segment can have a different outgoing COS value.
- It should be possible to support persistent as well as non-persistent LSPs.
- Performance counters must be provided for in-segments and out-segments as well as for measuring MPLS performance on a per-interface basis.

[5. Outline](#)

Configuring LSPs through an LSR involves the following steps.

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

[5.1. Summary of LSR MIB](#)

The MIB objects for performing these actions consist of the following tables.

- Interface configuration table (mplsInterfaceConfTable) for enabling MPLS on MPLS capable interfaces.

- In-segment (mplsInSegmentTable) and out-segment (mplsOutSegmentTable) tables for configuring LSP segments at

an LSR.

- Cross-connect table (mplsXCTable) for creating relationships between in and out segments constituting a cross-connect.
- Label stack table (mplsLabelStackTable) for specifying label stack operations.
- TSpec table (mplsTSpecTable) for specifying LSP related traffic parameters.

Further, the MPLS in-segment and out-segment performance tables, mplsInSegmentPerfTable and mplsOutSegmentPerfTable, contain the objects necessary to measure the performance of LSPs, and mplsInterfacePerfTable has objects to measure MPLS performance on a per-interface basis.

These tables are described in the subsequent sections.

6. Brief Description of MIB Objects

Sections 6.1-6.2 describe objects pertaining to MPLS capable interfaces of an LSR. The objects described in Sections 6.3-6.8, when considered together, are equivalent to the tables described in the MPLS architecture document [[MPLSArch](#)], that is, the Incoming Label Map (ILM) and the Next Hop Label Forwarding Entry (NHLFE) tables. Section 6.9 describes objects for specifying traffic parameters for in and out segments.

6.1. mplsInterfaceConfTable

This table represents the interfaces that are MPLS capable. An LSR creates an entry in this table for every MPLS capable interface on that LSR. Each entry contains information about per-interface label ranges. The administrator can specify the desired MPLS status (enable/up, disable/down, testing) of an interface by writing the object mplsInterfaceAdminStatus. The actual status is indicated by the object mplsInterfaceOperStatus.

6.2. mplsInterfacePerfTable

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

interfaces.

6.3. mplsInSegmentTable

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

6.4. mplsInSegmentPerfTable

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

6.5. mplsOutSegmentTable

The Out-Segment Table contains a description of the outgoing MPLS segments at an LSR and their associated parameters.

6.6. mplsOutSegmentPerfTable

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

6.7. mplsXCTable

mplsXCTable specifies information for switching between segments. It supports point-to-point, point-to-multipoint and multipoint-to-point connections.

6.8. mplsLabelStackTable

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

6.9. mplsTSpecTable

mplsTSpecTable contains objects for specifying the traffic parameters of in and out segments. Entries in this table are

referred to from `mplsInSegmentTable` and `mplsOutSegmentTable`.

7. Specifying the Segments of an LSR

Suppose that we want to manually create a best-effort bi-directional LSP, consisting of an in-segment and an out-segment on an LSR (with no label stack beneath the top label on the outgoing labeled packets). The following rows and corresponding objects need to be created to do this.

First, the in-segment and the out-segment are created with the appropriate traffic parameters.

In mplsInSegmentTable:

```
{
    mplsInSegmentIfIndex = i1,
    mplsInSegmentLabel = l1,
    mplsInSegmentNPop = 1,
    mplsInSegmentTSpecIndex = 0,
    mplsInSegmentRowStatus = createAndGo(3)
}
```

In mplsOutSegmentTable:

```
{
    mplsOutSegmentIndex = 0,
    mplsOutSegmentIfIndex = i2,
    mplsOutSegmentPushTopLabel = true(1),
    mplsOutSegmentTopLabel = l2,
    mplsOutSegmentTSpecIndex = 0,
    mplsOutSegmentRowStatus = createAndGo(3)
}
```

Next, two cross-connect entries associating these two segments by sharing the same mplsXCIndex are created.

In mplsXCTable, for the in-segment:

```
{
    mplsXCIndex = x,
    mplsInSegmentIfIndex = i1,
    mplsInSegmentLabel = l1,
    mplsOutSegmentIndex = 0,
    mplsLabelStackIndex = 0,
    mplsXCRowStatus = createAndGo(3)
}
```

In mplsXCTable, for the out-segment:

```
{
    mplsXCIndex = x,
```

```
mplsInSegmentIfIndex = 0,  
mplsInSegmentLabel = 0,
```



```
    mplsOutSegmentIndex = 0,  
    mplsXCLabelStackIndex = 0,  
    mplsXCRowStatus = createAndGo(3)  
}
```

Note that the objects mplsInSegmentXCIndex and mplsOutSegmentXCIndex will automatically get populated with the value "x" when these segments are referred to from the corresponding cross-connect entries.

8. MPLS Label Switch Router MIB Definitions

MPLS-LSR-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,  
experimental, Integer32, Counter32, Counter64, Gauge32, IPAddress  
    FROM SNMPv2-SMI  
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP  
    FROM SNMPv2-CONF  
TEXTUAL-CONVENTION, TruthValue, RowStatus  
    FROM SNMPv2-TC  
ifIndex, InterfaceIndex, InterfaceIndexOrZero  
    FROM IF-MIB  
BitRate, BurstSize  
    FROM INTEGRATED-SERVICES-MIB;
```

mplsLsrMIB MODULE-IDENTITY

```
LAST-UPDATED "9906161200Z" -- 16 June 1999 12:00:00 EST  
ORGANIZATION "Multiprotocol Label Switching (MPLS) Working Group"  
CONTACT-INFO
```

```
"      Cheenu Srinivasan  
      Postal: Tachion Network Technologies  
              2 Meridian Road  
              Eatontown, NJ 07724  
      Tel:    +1 732 542 7750 x234  
      Email:  cheenu@tachion.com
```

```
      Arun Viswanathan  
      Postal: Lucent Technologies  
              40537, 101 Crawfords Corner Road  
              Holmdel, NJ 07733  
      Tel:    +1 732 332 5163  
      Email:  arunv@lucent.com"
```

DESCRIPTION

```
"Proposed MIB module for MPLS Label Switch Router.
```

See: Rosen, E., Viswanathan, A., and R. Callon,
Multiprotocol Label Switching Architecture, Internet

Draft <[draft-ietf-mpls-arch-05.txt](#)>, February 1999."
::= { experimental 96 }

-- Textual Conventions.

MplsLsrIANAAddrFamily ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An address family. Values are defined in [RFC 1700](#) -
Assigned Numbers. All values may not be relevant in
all contexts when used in this MIB, but are included
for completeness."

REFERENCE

"RFC 1700 - Assigned Numbers, Reynolds and Postel,
Oct. 1994"

SYNTAX INTEGER {
 other(0),
 ipv4(1),
 ipv6(2),
 nsap(3),
 hdlc(4),
 bbn1822(5),
 ieee802(6),
 e163(7),
 e164(8),
 f69(9),
 x121(10),
 ipx(11),
 appleTalk(12),
 decnetIV(13),
 banyanVines(14),
 e164WithNsap(15)
}

-- An MPLS label.

MplsLabel ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents an MPLS label. Note that the contents of
a label field are interpreted in an interface-type
specific fashion. For example, the label carried in
the MPLS shim header is 20 bits wide and the top 12
bits must be zero. The frame relay label can be
either 10, 17 or 23 bits wide depending on the size
of the DLCI field size and the top 22, 15, or 9 bits
must be zero, respectively. For an ATM interface,

the lowermost 16 bits are interpreted as the VCI,
the next 8 bits as the VPI and the remaining bits
must be zero. Also note the permissible label

values are also a function of the interface type. For example, the value 3 has special semantics in the control plane for an MPLS shim header label and is not a valid label value in the datapath."

REFERENCE

1. MPLS Label Stack Encoding, Rosen et al, [draft-ietf-mpls-label-encaps-04.txt](#), April 1999.
2. Use of Label Switching on Frame Relay Networks, Conta et al, [draft-ietf-mpls-fr-03.txt](#), Nov. 1998."

SYNTAX Integer32

Ipv6Address ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"IPv6 address."

SYNTAX OCTET STRING (SIZE(16))

-- Top level components of this MIB.

-- tables, scalars

mplsLsrObjects OBJECT IDENTIFIER ::= { mplsLsrMIB 1 }

-- traps

mplsLsrNotifications OBJECT IDENTIFIER ::= { mplsLsrMIB 2 }

-- conformance

mplsLsrConformance OBJECT IDENTIFIER ::= { mplsLsrMIB 3 }

-- MPLS Interface Configuration Table.

mplsInterfaceConfTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsInterfaceConfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table specifies per-interface MPLS capability and associated information."

::= { mplsLsrObjects 1 }

mplsInterfaceConfEntry OBJECT-TYPE

SYNTAX MplsInterfaceConfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created by an LSR for every interface capable of supporting MPLS and represents the label space of that interface. A row with index 0 represents the global label space."

```
INDEX      { mplsInterfaceConfIndex  }  
::= { mplsInterfaceConfTable 1 }
```

```
MplsInterfaceConfEntry ::= SEQUENCE {  
    mplsInterfaceConfIndex      InterfaceIndexOrZero,  
    mplsInterfaceLabelMinIn     MplsLabel,  
    mplsInterfaceLabelMaxIn     MplsLabel,  
    mplsInterfaceLabelMinOut    MplsLabel,  
    mplsInterfaceLabelMaxOut    MplsLabel,  
    mplsInterfaceAdminStatus    INTEGER,  
    mplsInterfaceOperStatus     INTEGER  
}
```

mplsInterfaceConfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Index for this row of the table. A value of 0 indicates the global label space and this entry is created by the LSR if it supports a global label space. A non-zero index is also the interface index, ifIndex, for the corresponding interface entry in ifTable."

REFERENCE

"RFC 2233 - The Interfaces Group MIB using SMIV2, McCloghrie and Kastenholtz, Nov. 1997"

::= { mplsInterfaceConfEntry 1 }

mplsInterfaceLabelMinIn OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Minimum value of MPLS label that this LSR is willing to receive on this interface."

::= { mplsInterfaceConfEntry 2 }

mplsInterfaceLabelMaxIn OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Maximum value of MPLS label that this LSR is willing to receive on this interface."

::= { mplsInterfaceConfEntry 3 }

mplsInterfaceLabelMinOut OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 12]


```
    "Minimum value of MPLS label that this LSR is willing
      to send on this interface."
 ::= { mplsInterfaceConfEntry 4 }
```

mplsInterfaceLabelMaxOut OBJECT-TYPE

```
SYNTAX      MplsLabel
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "Maximum value of MPLS label that this LSR is willing
      to send on this interface."
 ::= { mplsInterfaceConfEntry 5 }
```

mplsInterfaceAdminStatus OBJECT-TYPE

```
SYNTAX      INTEGER {
    up(1),      -- enable MPLS on this interface
    down(2),    -- disable MPLS on this interface
    testing(3)  -- in some test mode
}
MAX-ACCESS   read-write
STATUS       current
DESCRIPTION
    "Indicates the administrator's intent as to whether
      MPLS should be enabled or disabled on this
      interface."
DEFVAL       { down }
 ::= { mplsInterfaceConfEntry 6 }
```

mplsInterfaceOperStatus 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
    lowerLayerNotPresent(7)
                -- down due to the state of
                -- lower layer interfaces
}
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "Indicates the actual status of MPLS on this
      interface."
 ::= { mplsInterfaceConfEntry 7 }
```

-- End of mplsInterfaceConfTable

-- MPLS Interface Performance Table.

```
mplsInterfacePerfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MplsInterfacePerfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides per-interface MPLS performance
        information."
    ::= { mplsLsrObjects 2 }
```

```
mplsInterfacePerfEntry OBJECT-TYPE
    SYNTAX      MplsInterfacePerfEntry
    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 mplsInterfaceConfEntry."
    AUGMENTS    { mplsInterfaceConfEntry }
    ::= { mplsInterfacePerfTable 1 }
```

```
MplsInterfacePerfEntry ::= SEQUENCE {
    -- incoming direction
    mplsInterfaceInLabelsUsed      Gauge32,
    mplsInterfaceInOctets          Counter32,
    mplsInterfaceInPackets         Counter32,
    mplsInterfaceInErrors          Counter32,
    mplsInterfaceInDiscards        Counter32,
    mplsInterfaceFailedLabelLookup Counter32,

    -- outgoing direction
    mplsInterfaceOutLabelsUsed     Gauge32,
    mplsInterfaceOutOctets         Counter32,
    mplsInterfaceOutPackets        Counter32,
    mplsInterfaceOutErrors         Counter32,
    mplsInterfaceOutDiscards       Counter32,

    -- high capacity counters
    mplsInterfaceInHCOctets        Counter64,
    mplsInterfaceOutHCOctets       Counter64
}
```

```
mplsInterfaceInLabelsUsed OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
```

DESCRIPTION

"The number of labels that are in use at this point

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 14]

on this interface in the incoming direction."
::= { mplsInterfacePerfEntry 1 }

mplsInterfaceInOctets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of octets that have been received in
labeled packets on this interface."
::= { mplsInterfacePerfEntry 2 }

mplsInterfaceInPackets OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of labeled packets that have been
received on this interface."
::= { mplsInterfacePerfEntry 3 }

mplsInterfaceInErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of labeled packets that have been
received on this interface that were errored."
::= { mplsInterfacePerfEntry 4 }

mplsInterfaceInDiscards OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of labeled packets that have been
received on this interface that were discarded."
::= { mplsInterfacePerfEntry 5 }

mplsInterfaceFailedLabelLookup OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of labeled packets that have been
received on this interface that were discarded
because no matching entries were found in

```
mplsInSegmentTable."  
::= { mplsInterfacePerfEntry 6 }
```

mplsInterfaceOutLabelsUsed OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of labels that are in use at this point
on this interface in the outgoing direction."

::= { mplsInterfacePerfEntry 7 }

mplsInterfaceOutOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets that have been sent as labeled
packets on this interface."

::= { mplsInterfacePerfEntry 8 }

mplsInterfaceOutPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of labeled packets that have been sent on
this interface."

::= { mplsInterfacePerfEntry 9 }

mplsInterfaceOutErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of labeled packets that could not be sent
on this interface due to errors."

::= { mplsInterfacePerfEntry 10 }

mplsInterfaceOutDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of outgoing labeled packets on this
interface that had to be discarded due to errors or
other conditions such as buffer overflows."

::= { mplsInterfacePerfEntry 11 }

mplsInterfaceInHCOctets OBJECT-TYPE

SYNTAX	Counter64
MAX-ACCESS	read-only
STATUS	current

DESCRIPTION

"The number of octets that have been received in labeled packets on this interface. This is the 64 bit version of mplsInterfaceInOctets."

::= { mplsInterfacePerfEntry 15 }

mplsInterfaceOutHCOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets that have been sent in labeled packets on this interface. This is the 64 bit version of mplsInterfaceOutOctets."

::= { mplsInterfacePerfEntry 16 }

-- End of mplsInterfacePerfTable

-- In-segment table.

mplsInSegmentTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsInSegmentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains a description of the incoming segments to a LSR."

::= { mplsLsrObjects 3 }

mplsInSegmentEntry OBJECT-TYPE

SYNTAX MplsInSegmentEntry

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 by an SNMP agent as instructed by LDP or RSVP. It is indexed by the incoming interface index and (top) label. Note that some of the segments are associated with a tunnel, the traffic parameters of these rows are supported as read-only objects and their modification can be done only via the tunnel table, mplsTunnelTable. This issue will be addressed more naturally when ownership related objects are introduced into these tables."

REFERENCE

"MPLS Traffic Engineering Management Information Base
Using SMIV2, Srinivasan and Viswanathan, [draft-ietf-](#)

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 17]

mpls-te-mib-01.txt, June 1999."

INDEX { mplsInSegmentIfIndex, mplsInSegmentLabel }
 ::= { mplsInSegmentTable 1 }

MplsInSegmentEntry ::= SEQUENCE {
 mplsInSegmentIfIndex InterfaceIndex,
 mplsInSegmentLabel MplsLabel,
 mplsInSegmentNPop Integer32,
 mplsInSegmentAddrFamily MplsLsrIANAAddrFamily,
 mplsInSegmentXCIndex Integer32,
 mplsInSegmentTSpecIndex Unsigned32,
 mplsInSegmentAdminStatus INTEGER,
 mplsInSegmentOperStatus INTEGER,
 mplsInSegmentRowStatus RowStatus
}

mplsInSegmentIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Incoming interface index. A value of zero represents an incoming label from the per-platform label space. In this case, the mplsInSegmentLabel is interpreted to be an MPLS-type label."

::= { mplsInSegmentEntry 1 }

mplsInSegmentLabel OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The incoming label."

::= { mplsInSegmentEntry 2 }

mplsInSegmentNPop OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of labels to pop from the incoming packet. Normally only the top label is popped (based on which all switching decisions are taken)."

DEFVAL { 1 }

::= { mplsInSegmentEntry 3 }

mplsInSegmentAddrFamily OBJECT-TYPE

SYNTAX	MplsLsrIANAAddrFamily
MAX-ACCESS	read-create
STATUS	current

DESCRIPTION

"The IANA address family of the incoming packet. A value of zero indicates that the family type is either unknown or undefined (which could happen for example when streams of different types are merged in a multipoint-to-point connection)."

REFERENCE

"RFC 1700 - Assigned Numbers, Reynolds and Postel, October 1994."

DEFVAL { 0 }
::= { mplsInSegmentEntry 4 }

mplsInSegmentXCIndex OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Index into mplsXCTable to identify which cross-connect entry this segment is part of. A value of zero indicates that it is not being referred to by any cross-connect entry."

DEFVAL { 0 }
::= { mplsInSegmentEntry 5 }

mplsInSegmentTSpecIndex OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Pointer into mplsTSpecTable indicating the TSpec to be assigned for this segment. A value of zero indicates best-effort treatment. Two or more segments can indicate resource sharing by pointing to the same entry in mplsTSpecTable."

DEFVAL { 0 }
::= { mplsInSegmentEntry 6 }

mplsInSegmentAdminStatus 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

"Desired status of this segment."

::= { mplsInSegmentEntry 7 }

mplsInSegmentOperStatus 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
    lowerLayerNotPresent(7)
                    -- down due to the state of
                    -- lower layer interfaces
}
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "The operational status of this segment."
 ::= { mplsInSegmentEntry 8 }
```

mplsInSegmentRowStatus OBJECT-TYPE

```
SYNTAX      RowStatus
MAX-ACCESS   read-create
STATUS       current
DESCRIPTION
    "For creating, modifying, and deleting this row."
 ::= { mplsInSegmentEntry 9 }
```

-- End of mplsInSegmentTable

-- In-segment performance table.

mplsInSegmentPerfTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF MplsInSegmentPerfEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
    "This table contains statistical information about
    incoming MPLS segments to an LSR."
 ::= { mplsLsrObjects 4 }
```

mplsInSegmentPerfEntry OBJECT-TYPE

```
SYNTAX      MplsInSegmentPerfEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION
    "An entry in this table contains statistical
    information about one incoming segment configured in
```

```
    mplsInSegmentTable."  
AUGMENTS      { mplsInSegmentEntry }  
    ::= { mplsInSegmentPerfTable 1 }
```



```
MplsInSegmentPerfEntry ::= SEQUENCE {  
    mplsInSegmentOctets      Counter32,  
    mplsInSegmentPackets     Counter32,  
    mplsInSegmentErrors      Counter32,  
    mplsInSegmentDiscards    Counter32,  
  
    -- high capacity counter  
    mplsInSegmentHCOctets    Counter64  
}
```

```
mplsInSegmentOctets OBJECT-TYPE  
    SYNTAX      Counter32  
    MAX-ACCESS   read-only  
    STATUS      current  
    DESCRIPTION  
        "Total number of octets received."  
    ::= { mplsInSegmentPerfEntry 1 }
```

```
mplsInSegmentPackets OBJECT-TYPE  
    SYNTAX      Counter32  
    MAX-ACCESS   read-only  
    STATUS      current  
    DESCRIPTION  
        "Total number of packets received."  
    ::= { mplsInSegmentPerfEntry 2 }
```

```
mplsInSegmentErrors OBJECT-TYPE  
    SYNTAX      Counter32  
    MAX-ACCESS   read-only  
    STATUS      current  
    DESCRIPTION  
        "Number of errored packets received."  
    ::= { mplsInSegmentPerfEntry 3 }
```

```
mplsInSegmentDiscards OBJECT-TYPE  
    SYNTAX      Counter32  
    MAX-ACCESS   read-only  
    STATUS      current  
    DESCRIPTION  
        "Number of packets received that had to be dropped  
        either because of errors or for other reasons such  
        as buffer overflows."  
    ::= { mplsInSegmentPerfEntry 4 }
```

```
mplsInSegmentHCOctets OBJECT-TYPE  
    SYNTAX      Counter32  
    MAX-ACCESS   read-only
```

STATUS	current
DESCRIPTION	

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 21]

```

    "Total number of octets received.  This is the 64 bit
      version of mplsInSegmentOctets."
  ::= { mplsInSegmentPerfEntry 5 }

-- End of mplsInSegmentPerfTable.

-- Out-segment table.

mplsOutSegmentTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MplsOutSegmentEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains a description of the outgoing
          segments from an LSR."
    ::= { mplsLsrObjects 5 }

mplsOutSegmentEntry OBJECT-TYPE
    SYNTAX      MplsOutSegmentEntry
    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 by an SNMP agent as instructed by
          LDP or RSVP.  Note that some of the segments are
          associated with a tunnel, the traffic parameters of
          these rows are supported as read-only objects and
          their modification can be done only via the tunnel
          table, mplsTunnelTable. This issue will be addressed
          more naturally when ownership related objects are
          introduced into these tables."
    REFERENCE
        "MPLS Traffic Engineering Management Information Base
          Using SMIV2, Srinivasan and Viswanathan, draft-ietf-
          mpls-te-mib-01.txt, June 1999."
    INDEX      { mplsOutSegmentIndex }
    ::= { mplsOutSegmentTable 1 }

MplsOutSegmentEntry ::= SEQUENCE {
    mplsOutSegmentIndex          Integer32,
    mplsOutSegmentIfIndex       InterfaceIndex,
    mplsOutSegmentPushTopLabel  TruthValue,
    mplsOutSegmentTopLabel      MplsLabel,
    mplsOutSegmentNextHopIpAddrType  INTEGER,
    mplsOutSegmentNextHopIpv4Addr  IpAddress,
```

mplsOutSegmentNextHopIpv6Addr	Ipv6Address,
mplsOutSegmentXCIndex	Integer32,
mplsOutSegmentTSpecIndex	Unsigned32,

```
    mplsOutSegmentAdminStatus      INTEGER,
    mplsOutSegmentOperStatus      INTEGER,
    mplsOutSegmentRowStatus      RowStatus
}
```

mplsOutSegmentIndex OBJECT-TYPE

```
SYNTAX      Integer32
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"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 mplsXCTable to access entries for which no out-segment has been configured."

```
::= { mplsOutSegmentEntry 1 }
```

mplsOutSegmentIfIndex OBJECT-TYPE

```
SYNTAX      InterfaceIndex
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"Interface index of the outgoing interface."

```
::= { mplsOutSegmentEntry 2 }
```

mplsOutSegmentPushTopLabel OBJECT-TYPE

```
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"Whether a top label should be pushed onto the outgoing packet's label stack. Its value has to be true if the outgoing interface is ATM (which does not support 'pop-and-go') or if it is a tunnel origination. Note also that the case where mplsOutSegmentPushTopLabel is set to false but the cross-connect entry that refers to this out-segment has a non-zero mplsLabelStackIndex is an error which the LSR should ensure doesn't happen."

```
::= { mplsOutSegmentEntry 3 }
```

mplsOutSegmentTopLabel OBJECT-TYPE

```
SYNTAX      MplsLabel
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"If mplsOutSegmentPushTopLabel is true then this is

the label that should be pushed onto the outgoing packet's label stack. Note that the contents of the label field can be interpreted in an outgoing

interface specific fashion. For example, the label carried in the MPLS shim header is 20 bits wide and the top 12 bits must be zero. The Frame Relay label is 24 bits wide and the top 8 bits must be zero. For ATM interfaces the lowermost 16 bits are interpreted as the VCI, the next 8 bits as the VPI and the remaining bits must be zero."

::= { mplsOutSegmentEntry 4 }

mplsOutSegmentNextHopIpAddrType OBJECT-TYPE

SYNTAX INTEGER { none (1), ipv4 (2), ipv6 (3) }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Whether the next hop address is IPv4 or IPv6. A value of none (1) is valid (only) when the outgoing interface is of type point-to-point."

DEFVAL { none }

::= { mplsOutSegmentEntry 5 }

mplsOutSegmentNextHopIpv4Addr OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"IPv4 Address of the next hop. Its value is significant only when mplsOutSegmentNextHopIpAddrType is ipv4 (2), otherwise it should return a value of 0."

::= { mplsOutSegmentEntry 6 }

mplsOutSegmentNextHopIpv6Addr OBJECT-TYPE

SYNTAX Ipv6Address

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"IPv6 address of the next hop. Its value is significant only when mplsOutSegmentNextHopIpAddrType is ipv6 (3), otherwise it should return a value of 0."

::= { mplsOutSegmentEntry 7 }

mplsOutSegmentXCIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Index into mplsXCTable to identify which cross-connect entry this segment is part of. A value of zero indicates that it is not being referred to by


```
any cross-connect entry."
DEFVAL      { 0 }
::= { mplsOutSegmentEntry 8 }
```

mplsOutSegmentTSpecIndex OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS   read-create
STATUS       current
DESCRIPTION
    "Pointer into mplsTSpecTable indicating the TSpec to
    be assigned for this segment. A value of zero
    indicates best-effort treatment. Two or more
    segments can indicate resource sharing by pointing
    to the same entry in mplsTSpecTable."
DEFVAL      { 0 }
::= { mplsOutSegmentEntry 9 }
```

mplsOutSegmentAdminStatus 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
    "Desired status of this segment."
::= { mplsOutSegmentEntry 10 }
```

mplsOutSegmentOperStatus 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
    lowerLayerNotPresent(7)
                  -- down due to the state of
                  -- lower layer interfaces
}
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "The operational status of this segment."
::= { mplsOutSegmentEntry 11 }
```

mpIsOutSegmentRowStatus OBJECT-TYPE
SYNTAX RowStatus

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 25]

```
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "For creating, modifying, and deleting this row."
::= { mplsOutSegmentEntry 12 }
```

```
-- End of mplsOutSegmentTable
```

```
-- Out-segment performance table.
```

```
mplsOutSegmentPerfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MplsOutSegmentPerfEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains statistical information about
         incoming segments to an LSR."
    ::= { mplsLsrObjects 6 }
```

```
mplsOutSegmentPerfEntry OBJECT-TYPE
    SYNTAX      MplsOutSegmentPerfEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in this table contains statistical
         information about one incoming segment configured in
         mplsOutSegmentTable."
    AUGMENTS    { mplsOutSegmentEntry }
    ::= { mplsOutSegmentPerfTable 1 }
```

```
MplsOutSegmentPerfEntry ::= SEQUENCE {
    mplsOutSegmentOctets      Counter32,
    mplsOutSegmentPackets    Counter32,
    mplsOutSegmentErrors     Counter32,
    mplsOutSegmentDiscards   Counter32,

    -- HC counter
    mplsOutSegmentHCOctets   Counter64
}
```

```
mplsOutSegmentOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Total number of octets sent."
    ::= { mplsOutSegmentPerfEntry 1 }
```

mplsOutSegmentPackets OBJECT-TYPE

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 26]

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Total number of packets sent."
::= { mplsOutSegmentPerfEntry 2 }

mplsOutSegmentErrors OBJECT-TYPE

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

mplsOutSegmentDiscards OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Number of outgoing packets that had to be dropped either because of errors or for other reasons such as buffer overflows."
::= { mplsOutSegmentPerfEntry 4 }

mplsOutSegmentHCOctets OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Total number of octets sent. This is the 64 bit version of mplsOutSegmentOctets."
::= { mplsOutSegmentPerfEntry 5 }

-- End of mplsOutSegmentPerfTable.

-- Cross-connect table.

mplsXCTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsXCEntry
MAX-ACCESS not-accessible
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. mplsLabelStackTable specifies the
label stack information for a cross-connect LSR and

is referred to from mplsXCTable."
 ::= { mplsLsrObjects 7 }

mplsXCEntry OBJECT-TYPE

SYNTAX MplsXCEntry
 MAX-ACCESS not-accessible
 STATUS current

DESCRIPTION

"A row in this table represents one cross-connect entry. It is indexed by the following objects:

- cross-connect index mplsXCIndex that uniquely identifies a group of cross-connect entries
- interface index of the in-segment, mplsInSegmentIfIndex
- incoming label(s), mplsInSegmentLabel
- out-segment index, mplsOutSegmentIndex

Originating LSPs:

These are represented by using the special combination of values mplsInSegmentIfIndex=0 and mplsInSegmentLabel=0 as indexes. In this case the mplsOutSegmentIndex MUST be non-zero.

Terminating LSPs:

These are represented by using the special value mplsOutSegmentIndex=0 as index.

Special labels:

Entries indexed by reserved MPLS label values 0 through 15 imply terminating LSPs and MUST have mplsOutSegmentIndex=0.

An entry can be created by a network administrator or by an SNMP agent as instructed by LDP or RSVP."

INDEX { mplsXCIndex, mplsInSegmentIfIndex,
 mplsInSegmentLabel, mplsOutSegmentIndex }
 ::= { mplsXCTable 1 }

MplsXCEntry ::= SEQUENCE {

mplsXCIndex	INTEGER,
mplsXCLabelStackIndex	Integer32,
mplsXCCOS	Integer32,
mplsXCIsPersistent	TruthValue,
mplsXCAdminStatus	INTEGER,
mplsXCOperStatus	INTEGER,
mplsXCRowStatus	RowStatus

}

mplsXCIndex OBJECT-TYPE

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 28]

SYNTAX INTEGER (0..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Primary index for the row identifying a group of cross-connect segments."

::= { mplsXCEnt 1 }

mplsXCLabelStackIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Primary index into mplsLabelStackTable identifying a stack of labels to be pushed beneath the top label. Note that the top label is identified in the out-segment which 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."

::= { mplsXCEnt 2 }

mplsXCCOS OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Value to override the incoming COS field with for a cross-connect or the value to assign to outgoing packets for an outgoing segment of a tunnel."

::= { mplsXCEnt 3 }

mplsXCIsPersistent OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Whether this cross-connect entry and associated in- and out-segments should be restored automatically after failures."

DEFVAL { false }

::= { mplsXCEnt 4 }

mplsXCAdminStatus 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
    "Desired status of this segment."
::= { mplsXCEntry 5 }
```

mplsXCOperStatus 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
    lowerLayerNotPresent(7)
                    -- down due to the state of
                    -- lower layer interfaces
}
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The operational status of this segment."
::= { mplsXCEntry 6 }
```

mplsXCRowStatus OBJECT-TYPE

```
SYNTAX          RowStatus
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "For creating, modifying, and deleting this row."
::= { mplsXCEntry 7 }
```

-- End of mplsXCTable

-- Label stack table.

mplsLabelStackTable OBJECT-TYPE

```
SYNTAX          SEQUENCE OF MplsLabelStackEntry
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 mplsXCTable."
::= { mplsLsrObjects 8 }
```

mplsLabelStackEntry OBJECT-TYPE

SYNTAX	MplsLabelStackEntry
MAX-ACCESS	not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents one label to be pushed onto an outgoing packets beneath the top label. An entry can be created by a network administrator or by an SNMP agent as instructed by LDP or RSVP."

INDEX { mplsLabelStackIndex }

::= { mplsLabelStackTable 1 }

MplsLabelStackEntry ::= SEQUENCE {

mplsLabelStackIndex Integer32,

mplsLabelStackLabelIndex Integer32,

mplsLabelStackLabel MplsLabel,

mplsLabelStackRowStatus RowStatus

}

mplsLabelStackIndex OBJECT-TYPE

SYNTAX Integer32

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."

::= { mplsLabelStackEntry 1 }

mplsLabelStackLabelIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Secondary index for this row identifying one label of the stack."

::= { mplsLabelStackEntry 2 }

mplsLabelStackLabel OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Label to pushed."

::= { mplsLabelStackEntry 3 }

mplsLabelStackRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 31]

```
::= { mplsLabelStackEntry 4 }
```

```
-- End of mplsLabelStackTable
```

```
-- TSpec table.
```

```
mplsTSpecTable OBJECT-TYPE
```

```
    SYNTAX          SEQUENCE OF MplsTSpecEntry
```

```
    MAX-ACCESS      not-accessible
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "This table specifies TSpec objects for in and out-
        segments."
```

```
    ::= { mplsLsrObjects 9 }
```

```
mplsTSpecEntry OBJECT-TYPE
```

```
    SYNTAX          MplsTSpecEntry
```

```
    MAX-ACCESS      not-accessible
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "An entry in this table represents the TSpec objects
        for one or more in or out segments. A single entry
        can be pointed to by multiple segments indicating
        resource sharing."
```

```
    INDEX          { mplsTSpecIndex }
```

```
    ::= { mplsTSpecTable 1 }
```

```
MplsTSpecEntry ::= SEQUENCE {
```

mplsTSpecIndex	Unsigned32,
mplsTSpecIfIndex	InterfaceIndex,
mplsTSpecDirection	INTEGER,
mplsTSpecMaxRate	BitRate,
mplsTSpecMeanRate	BitRate,
mplsTSpecMaxBurstSize	BurstSize,
mplsTSpecRowStatus	RowStatus

```
}
```

```
mplsTSpecIndex OBJECT-TYPE
```

```
    SYNTAX          Unsigned32
```

```
    MAX-ACCESS      read-create
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "Uniquely identifies this row of the table. Zero is
        not a valid index."
```

```
    ::= { mplsTSpecEntry 1 }
```

```
mplsTSpecIfIndex OBJECT-TYPE
```

```
    SYNTAX          InterfaceIndex
```

MAX-ACCESS	read-create
STATUS	current

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 32]

DESCRIPTION

"Identifies the interface that this entry refers to."

::= { mplSTSpecEntry 2 }

mplSTSpecDirection OBJECT-TYPE

SYNTAX INTEGER { in(1), out(2) }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Direction that these objects pertain to, incoming or outgoing."

::= { mplSTSpecEntry 3 }

mplSTSpecMaxRate OBJECT-TYPE

SYNTAX BitRate

UNITS "bits per second"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Maximum rate in bits/second."

::= { mplSTSpecEntry 4 }

mplSTSpecMeanRate OBJECT-TYPE

SYNTAX BitRate

UNITS "bits per second"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Mean rate in bits/second."

::= { mplSTSpecEntry 5 }

mplSTSpecMaxBurstSize OBJECT-TYPE

SYNTAX BurstSize

UNITS "bytes"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Maximum burst size in bytes."

::= { mplSTSpecEntry 6 }

mplSTSpecRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

::= { mplSTSpecEntry 7 }

-- End of mplSTSpecTable

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 33]

-- Notifications.

-- Interface configuration.

mplsInterfaceUp NOTIFICATION-TYPE

OBJECTS { mplsInterfaceConfIndex,
mplsInterfaceAdminStatus, mplsInterfaceOperStatus }

STATUS current

DESCRIPTION

"This notification is generated when a mplsInterfaceOperStatus object for one of the entries in mplsInterfaceConfTable is about to leave the down state and transition into some other state (but not into the notPresent state). This other state is indicated by the included value of mplsInterfaceOperStatus."

::= { mplsLsrNotifications 1 }

mplsInterfaceDown NOTIFICATION-TYPE

OBJECTS { mplsInterfaceConfIndex,
mplsInterfaceAdminStatus, mplsInterfaceOperStatus }

STATUS current

DESCRIPTION

"This notification is generated when a mplsInterfaceOperStatus object for one of the entries in mplsInterfaceConfTable is about to enter the down state from some other state (but not from the notPresent state). This other state is indicated by the included value of mplsInterfaceOperStatus."

::= { mplsLsrNotifications 2 }

-- In-segment.

mplsInSegmentUp NOTIFICATION-TYPE

OBJECTS { mplsInSegmentIfIndex, mplsInSegmentLabel,
mplsInSegmentAdminStatus, mplsInSegmentOperStatus }

STATUS current

DESCRIPTION

"This notification is generated when a mplsInSegmentOperStatus object for one of the configured in-segments is about to leave the down state and transition into some other state (but not into the notPresent state). This other state is indicated by the included value of mplsInSegmentOperStatus."

::= { mplsLsrNotifications 3 }

mplsInSegmentDown NOTIFICATION-TYPE

```
OBJECTS      { mplsInSegmentIfIndex, mplsInSegmentLabel,
               mplsInSegmentAdminStatus, mplsInSegmentOperStatus }
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This notification is generated when a
 mplsInSegmentOperStatus object for one of the
 configured in-segments is about to enter the down
 state from some other state (but not from the
 notPresent state). This other state is indicated by
 the included value of mplsInSegmentOperStatus."
```

```
::= { mplsLsrNotifications 4 }
```

```
-- Out-segment.
```

```
mplsOutSegmentUp NOTIFICATION-TYPE
```

```
OBJECTS      { mplsOutSegmentIndex, mplsInSegmentAdminStatus,
               mplsInSegmentOperStatus }
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This notification is generated when a
 mplsOutSegmentOperStatus object for one of the
 configured out-segments is about to leave the down
 state and transition into some other state (but not
 into the notPresent state). This other state is
 indicated by the included value of
 mplsOutSegmentOperStatus."
```

```
::= { mplsLsrNotifications 5 }
```

```
mplsOutSegmentDown NOTIFICATION-TYPE
```

```
OBJECTS      { mplsOutSegmentIndex, mplsInSegmentAdminStatus,
               mplsInSegmentOperStatus }
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This notification is generated when a
 mplsOutSegmentOperStatus object for one of the
 configured out-segments is about to enter the down
 state from some other state (but not from the
 notPresent state). This other state is indicated by
 the included value of mplsOutSegmentOperStatus."
```

```
::= { mplsLsrNotifications 6 }
```

```
-- Cross-connect.
```

```
mplsXCUp NOTIFICATION-TYPE
```

```
OBJECTS      { mplsXCIndex,
               mplsInSegmentIfIndex, mplsInSegmentLabel,
               mplsOutSegmentIndex,
```

```
        mplsXCAdminStatus, mplsXCOperStatus }  
STATUS      current  
DESCRIPTION
```

"This notification is generated when a mplsXCOperStatus object for one of the configured cross-connect entries is about to leave the down state and transition into some other state (but not into the notPresent state). This other state is indicated by the included value of mplsXCOperStatus."

::= { mplsLsrNotifications 7 }

mplsXCDown NOTIFICATION-TYPE

OBJECTS { mplsXCIndex,
mplsInSegmentIfIndex, mplsInSegmentLabel,
mplsOutSegmentIndex,
mplsXCAdminStatus, mplsXCOperStatus }

STATUS current

DESCRIPTION

"This notification is generated when a mplsXCOperStatus object for one of the configured cross-connect entries is about to enter the down state from some other state (but not from the notPresent state). This other state is indicated by the included value of mplsXCOperStatus."

::= { mplsLsrNotifications 8 }

-- End of notifications.

-- Module compliance.

mplsLsrGroups

OBJECT IDENTIFIER ::= { mplsLsrConformance 1 }

mplsLsrCompliances

OBJECT IDENTIFIER ::= { mplsLsrConformance 2 }

mplsLsrModuleCompliance MODULE-COMPLIANCE

STATUS current

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 not
-- supported.

MANDATORY-GROUPS { mplsInSegmentGroup, mplsOutSegmentGroup,

```
mplsXCGroup, mplsInterfaceGroup,  
mplsPerfGroup }
```


GROUP mplsHCInterfacePerfGroup

DESCRIPTION

"This group is mandatory for high-speed MPLS capable interfaces for which the objects mplsInterfaceInOctets and mplsInterfaceOutOctets wrap around too quickly."

GROUP mplsHCInSegmentPerfGroup

DESCRIPTION

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

GROUP mplsHCOutSegmentPerfGroup

DESCRIPTION

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

GROUP mplsTSpecGroup

DESCRIPTION

"This group is mandatory for those LSRs that support int-serv style resource reservation."

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

GROUP mplsXCIsPersistentGroup

DESCRIPTION

"This group is mandatory for devices which support persistent cross-connects. The following constraints apply: mplsXCIsPersistent must at least be read-only returning true(2)."

GROUP mplsXCIsNotPersistentGroup

DESCRIPTION

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

-- mplsInterfaceConfTable

OBJECT mplsInterfaceAdminStatus

SYNTAX INTEGER { up(1), down(2) }
MIN-ACCESS read-only

DESCRIPTION

"A value of testing(3) need not be supported."

OBJECT mplsInterfaceOperStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"Only up(1) and down(2) need to be supported."

-- mplsInSegmentTable

OBJECT mplsInSegmentIfIndex

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsInSegmentLabel

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsInSegmentXCIndex

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsInSegmentTSpecIndex

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsInSegmentNPop

MIN-ACCESS read-only

DESCRIPTION

"Write access if not required. This object should be set to 1 if it is read-only."

OBJECT mplsInSegmentAddrFamily

SYNTAX INTEGER { other(0) }

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required. A value of other(0) should be supported."

OBJECT mplsInSegmentAdminStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"A value of testing(3) need not be supported."

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 38]

```
OBJECT      mplsInSegmentOperStatus
SYNTAX      INTEGER { up(1), down(2) }
MIN-ACCESS  read-only
DESCRIPTION
    "Only up(1) and down(2) need to be supported."

OBJECT      mplsInSegmentRowStatus
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."

-- mplsOutSegmentTable

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

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

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

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

OBJECT      mplsOutSegmentNextHopIpAddrType
SYNTAX      INTEGER { none(1), ipv4(2) }
MIN-ACCESS  read-only
DESCRIPTION
    "ipv6(3) need not be supported."

OBJECT      mplsOutSegmentNextHopIpv4Addr
MIN-ACCESS  read-only
DESCRIPTION
```

"Write access is not required."

OBJECT mplsOutSegmentNextHopIpv6Addr
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsOutSegmentXCIndex
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsOutSegmentTSpecIndex
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsOutSegmentAdminStatus
SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"A value of testing(3) need not be supported."

OBJECT mplsOutSegmentOperStatus
SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"Only up(1) and down(2) need to be supported."

OBJECT mplsOutSegmentRowStatus
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."

-- mplsXCTable

OBJECT mplsXCIndex
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsXCLabelStackIndex
MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsXCCOS

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 40]

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsXCIsPersistent

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsXCAdminStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"A value of testing(3) need not be supported."

OBJECT mplsXCOperStatus

SYNTAX INTEGER { up(1), down(2) }

MIN-ACCESS read-only

DESCRIPTION

"Only up(1) and down(2) need to be supported."

OBJECT mplsXCRowStatus

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."

::= { mplsLsrCompliances 1 }

-- Units of conformance.

mplsInterfaceGroup OBJECT-GROUP

OBJECTS { mplsInterfaceConfIndex,
mplsInterfaceLabelMinIn, mplsInterfaceLabelMaxIn,
mplsInterfaceLabelMinOut, mplsInterfaceLabelMaxOut,
mplsInterfaceAdminStatus, mplsInterfaceOperStatus,
mplsInterfaceInLabelsUsed, mplsInterfaceOutLabelsUsed }

STATUS current

DESCRIPTION

"Collection of objects needed for MPLS interface
configuration and performance information."

::= { mplsLsrGroups 1 }

mplsInSegmentGroup OBJECT-GROUP

```
OBJECTS { mplsInSegmentIfIndex, mplsInSegmentLabel,  
          mplsInSegmentNPop, mplsInSegmentAddrFamily,
```

```
    mplsInSegmentXCIndex, mplsInSegmentTSpecIndex,  
    mplsInSegmentAdminStatus, mplsInSegmentOperStatus,  
    mplsInSegmentRowStatus,  
    mplsInSegmentOctets, mplsInSegmentDiscards }
```

STATUS current

DESCRIPTION

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

::= { mplsLsrGroups 2 }

mplsOutSegmentGroup OBJECT-GROUP

```
OBJECTS { mplsOutSegmentIndex, mplsOutSegmentIfIndex,  
    mplsOutSegmentPushTopLabel, mplsOutSegmentTopLabel,  
    mplsOutSegmentNextHopIpAddrType,  
    mplsOutSegmentNextHopIpv4Addr,  
    mplsOutSegmentNextHopIpv6Addr,  
    mplsOutSegmentXCIndex, mplsOutSegmentTSpecIndex,  
    mplsOutSegmentAdminStatus, mplsOutSegmentOperStatus,  
    mplsOutSegmentRowStatus,  
    mplsOutSegmentOctets, mplsOutSegmentDiscards }
```

STATUS current

DESCRIPTION

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

::= { mplsLsrGroups 3 }

mplsXCGroup OBJECT-GROUP

```
OBJECTS { mplsXCIndex, mplsXCLabelStackIndex,  
    mplsXCAdminStatus, mplsXCOperStatus, mplsXCRowStatus }
```

STATUS current

DESCRIPTION

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

::= { mplsLsrGroups 4 }

mplsPerfGroup OBJECT-GROUP

```
OBJECTS { mplsInterfaceInOctets, mplsInterfaceInPackets,  
    mplsInterfaceInDiscards,  
    mplsInterfaceOutOctets, mplsInterfaceOutPackets,  
    mplsInterfaceOutDiscards,  
    mplsInSegmentOctets, mplsInSegmentPackets,  
    mplsInSegmentDiscards,  
    mplsOutSegmentOctets, mplsOutSegmentPackets,  
    mplsOutSegmentDiscards }
```

STATUS current

DESCRIPTION

"Collection of objects providing performance

```
information about an LSR."  
::= { mplslsrGroups 5 }
```

mplsHCInterfacePerfGroup OBJECT-GROUP

OBJECTS { mplsInterfaceInHCOctets, mplsInterfaceOutHCOctets }

STATUS current

DESCRIPTION

"Collection of objects providing performance information specific to high-speed interfaces for which the objects mplsInterfaceInOctets and mplsInterfaceOutOctets wrap-around too quickly."

::= { mplsLsrGroups 6 }

mplsHCInSegmentPerfGroup OBJECT-GROUP

OBJECTS { mplsInSegmentHCOctets }

STATUS current

DESCRIPTION

"Object(s) providing performance information specific to out-segments for which the object mplsInterfaceInOctets wraps around too quickly."

::= { mplsLsrGroups 7 }

mplsHCOutSegmentPerfGroup OBJECT-GROUP

OBJECTS { mplsOutSegmentHCOctets }

STATUS current

DESCRIPTION

"Object(s) providing performance information specific to out-segments for which the object mplsInterfaceOutOctets wraps around too quickly."

::= { mplsLsrGroups 8 }

mplsTSpecGroup OBJECT-GROUPOBJECTS { mplsTSpecIndex, mplsTSpecIfIndex, mplsTSpecDirection,
mplsTSpecMaxRate, mplsTSpecMeanRate,
mplsTSpecMaxBurstSize, mplsTSpecRowStatus }

STATUS current

DESCRIPTION

"Object(s) required for supporting int-serv style resource reservation."

::= { mplsLsrGroups 9 }

mplsXCIsPersistentGroup OBJECT-GROUP

OBJECTS { mplsXCIsPersistent }

STATUS current

DESCRIPTION

"Objects needed to support persistent cross-connects."

::= { mplsLsrGroups 10 }

mplsXCIsNotPersistentGroup OBJECT-GROUP

OBJECTS { mplsXCisPersistent }
STATUS current
DESCRIPTION

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 43]

```
    "Objects needed to support non-persistent cross-
      connects."
  ::= { mplsLsrGroups 11 }

mplsLsrNotificationGroup NOTIFICATION-GROUP
  NOTIFICATIONS { mplsInterfaceUp, mplsInterfaceDown,
                  mplsInSegmentUp, mplsInSegmentDown,
                  mplsOutSegmentUp, mplsOutSegmentDown,
                  mplsXCUp, mplsXCDown }
  STATUS current
  DESCRIPTION
    "Set of notifications implemented in this module.
     None is mandatory."
  ::= { mplsLsrGroups 12 }

-- End of MPLS-LSR-MIB
END
```

9. Security Considerations

The MIB specified in this document does not raise any security issues other than those present in the MPLS architecture [[MPLSArch](#)] or those imposed by SNMP itself.

10. Acknowledgments

We wish to thank Ron Bonica, Eric Gray, and Dan Tappan for their comments on this draft.

11. References

- [MPLSArch] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <[draft-ietf-mpls-arch-05.txt](#)>, February 1999.
- [MPLSFW] Callon, R., Doolan, P., Feldman, N., Fredette, A., Swallow, G., and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Internet Draft <[draft-ietf-mpls-framework-02.txt](#)>, November 1997.
- [LDPMIB] Cucchiara, J., Sjostrand, H., and J. Luciani, "Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)", Internet Draft <[draft-ietf-mpls-](#)

[ldp-mib-00.txt](#)>, August 1998.

- [LblStk] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Federokow, G., Li, T., and A. Conta, "MPLS Label Stack Encoding", Internet Draft <[draft-ietf-mpls-label-encaps-04.txt](#)>, April 1999.
- [RSVPTun] Awaduche, D., Berger, L., Der-Haw, G., Li, T., Swallow, G., and V. Srinivasan, "Extensions to RSVP for LSP Tunnels", Internet Draft <[draft-mpls-rsvp-lsp-tunnel-02.txt](#)>, March 1999.
- [CRLDP] B. Jamoussi (Editor), "Constraint-Based LSP Setup using LDP", Internet Draft <[draft-ietf-mpls-cr-ldp-01.txt](#)>, February 1999.
- [Assigned] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1700, October 1994. See also: <http://www.isi.edu/in-notes/iana/assignments/smi-numbers>
- [SNMPArch] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2271](#), January 1998.
- [SMIv1] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", [RFC 1155](#), May 1990.
- [SNMPv1MIBDef] Rose, M., and K. McCloghrie, "Concise MIB Definitions", [RFC 1212](#), March 1991.
- [SNMPv1Traps] M. Rose, "A Convention for Defining Traps for use with the SNMP", [RFC 1215](#), March 1991.
- [SMIv2] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1902](#), January 1996.
- [SNMPv2TC] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, SNMP Research, Inc., Cisco Systems, Inc., January 1996.
- [SNMPv2Conf] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1904](#), January 1996.

[SNMPv1] Case, J., Fedor, M., Schoffstall, M., and J. Davin,

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 45]

"Simple Network Management Protocol", [RFC 1157](#), May 1990.

- [SNMPv2c] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", [RFC 1901](#), January 1996.
- [SNMPv2TM] 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.
- [SNMPv3MP] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, January 1998.
- [SNMPv3USM] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, January 1998.
- [SNMPv2P0] 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.
- [SNMPv3App] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", [RFC 2273](#), January 1998
- [SNMPv3VACM] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, January 1998

[12.](#) Authors's Addresses

Cheenu Srinivasan
Tachion Network Technologies
2 Meridian Road
Eatontown, NJ 07724

Phone: +1-732-542-7750 x234
Email: cheenu@tachion.com

Arun Viswanathan

Lucent Technologies
4D537, 101 Crawford's Corner Road

Srinivasan & Viswanathan

Expires 16 December 1999

[Page 46]

Holmdel, NJ 07733

Phone: +1-732-332-5163

Email: arunv@lucent.com

