MPLS Traffic Engineering Management Information Base

draft-srinivasan-mpls-te-mib-01.txt

January 11, 1999

Cheenu Srinivasan
Lucent Technologies
cheenu@lucent.com

Arun Viswanathan
Lucent Technologies
arunv@lucent.com

Status of this Memo

This document is an Internet Draft. 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. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

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 Multi-Protocol Label Switching (MPLS) [1, 2] based traffic engineering.
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 Multi-Protocol Label Switching (MPLS) [1, 2] based traffic engineering, including tunnels and cross-connects. Comments should be made directly to the MPLS mailing list at mpls@external.cisco.com.

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 [1].

An explicitly routed LSP is referred to as an MPLS tunnel. It consists of one in-segment and/or one out-segment at the ingress/egress LSRs. These are also referred to as tunnel segments. Additionally, at an intermediate LSR, we model a connection as consisting of one or more in-segments and/or one or more out-segments. The binding or interconnection between in-segments and out-segments in performed using a cross-connect.

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2271 [7].

- 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 [8], RFC 1212 [9] and RFC 1215 [10]. The second version, called SMIv2, is described in RFC 1902 [11], RFC 1903 [12] and RFC 1904 [13].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [14]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [15] and RFC 1906 [16]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [16], RFC 2272 [17] and RFC 2274 [18].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [14]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [19].

- A set of fundamental applications described in RFC 2273 [20] and the view-based access control mechanism described in RFC 2275 [21]. 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 traffic engineering MIB is designed to satisfy the following requirements and constraints.

- The MIB must support the configuration of point-to-point unidirectional tunnels.

- The MIB should be able to support the configuration of point-to-point bi-directional tunnels.

- The MIB should be able to support the configuration of multipoint-to-point unidirectional tunnels.
- MPLS tunnels need not be interfaces, but it should be possible to configure a tunnel as an interface.

- The MIB should be able to support both manually configured MPLS tunnels and cross-connects as well as via LDP and/or RSVP signalling.

- MPLS packets must be forwarded solely based on an incoming top label [1, 3].

- Support must be provided for next-hop resolution when the outgoing interface is a shared media interface. In the multicast case, each outgoing segment can be on a different shared media interface.

- The MIB must support point-to-point, point-to-multipoint (multicast) and multipoint-to-point connections at a cross-connect 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 multicast case each outgoing connection can have a distinct label stack including the top label.

- In a multicast 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") [3].

- It must be possible to assign or remap COS bits [3] on the outgoing label. In the multipoint-to-point case, each in-segment can have a different outgoing COS value. In the multicast case, each out-segment can have a different outgoing COS value.

- It should be possible to support persistent as well as non-persistent tunnels and cross-connects.
Performance counters must be provided for in-segments and out-segments.

5. Outline

Traffic engineering support for MPLS tunnels and cross-connects requires the following configuration.

- Setting up MPLS tunnels with the appropriate configuration parameters.
- Setting up tunnel segments with appropriate traffic parameters.
- Setting up the cross-connect table to switch between segments.
- Specifying label stack actions.

5.1. Summary of MPLS MIB

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

- Tunnel Table (mplsTunnelTable) and Tunnel Hop Table (mplsTunnelHopTable) to configure MPLS tunnels.
- In-Segment (mplsInSegmentTable) and Out-Segment (mplsOutSegmentTable) Tables for configuring in and out segment of a tunnel at the ingress and egress LSRs of the tunnel. These tables are also used for defining the segments comprising a cross-connect entry at intermediate LSRs of a tunnel.
- Cross-Connect Table (mplsXCTable) for configuring MPLS cross-connects and creating relationships between in and out tunnel segments constituting a cross-connect.
- Label Stack Table (mplsLabelStackTable) for specifying label stack operations.

Further, the MPLS In-Segment and Out-Segment Performance Tables contain the objects necessary to measure the performance of both tunnels and cross-connects. These tables are described in the subsequent sections.

6. MPLS Tunnels

The tables described in this section support the functionality
The tables support both manually configured and signalled tunnels. Moreover, it provides the capability to associate two uni-directional tunnels to form a single bi-directional tunnel.

6.1. mplsTunnelTable

The mplsTunnelTable allows new MPLS tunnels to be created between an MPLS LSR and a remote endpoint, and existing tunnels to be reconfigured or removed. Note that we only support point-to-point tunnel segments, although multipoint-to-point and point-to-multipoint connections are supported by an LSR acting as a cross-connect. Each MPLS tunnel can thus have one out-segment originating at this LSR and/or one in-segment terminating at this LSR.

mplsTunnelTable does not define the in and out segments forming the tunnel. Instead, these are defined by creating rows in the in-segment and out-segment tables, defining relationships in the cross-connect table (Section 5) and referring to these rows in the mplsTunnelTable using a cross-connect index, mplsTunnelXCID. Section 5.7 provides a detailed explanation.

6.2. mplsTunnelHopTable

mplsTunnelHopTable is used to indicate the hops, strict or loose, for an MPLS tunnel defined in mplsTunnelTable, when it is established via signalling. Each row in this table is indexed primarily by the same index mplsTunnelIndex as the row of the corresponding tunnel in mplsTunnelTable. Each row also has a secondary index mplsTunnelHopIndex corresponding to the next hop of this tunnel. The scalar mplsTunnelMaxHops, indicates the maximum number of hops that can be specified per tunnel on this LSR.

7. MPLS Cross-Connect

The tables described in this section, when considered together, are equivalent to the tables described in the MPLS architecture document [1], that is, the Next Hop Label Forwarding Entry (NHLFE) and the Incoming Label Map (ILM) tables.

7.1. mplsInSegmentTable

This table contains a description of the the incoming MPLS segments to an LSR and their traffic parameters.
7.2. 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.

7.3. mplsOutSegmentTable

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

7.4. 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.

7.5. mplsXCTable

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

7.6. 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.

7.7. Specifying the Segments of a Tunnel

Suppose that we want to manually create a bi-directional tunnel, 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,
mlsInSegmentLabel = 11,
mlsInSegmentNPop = 1,
mlsInSegmentMaxRate,
mlsInSegmentMeanRate,
mlsInSegmentMaxBurstSize,
mlsInSegmentRowStatus = createAndGo(3)
}

In mplsOutSegmentTable:
{
  mplsOutSegmentIndex = o,
  mplsOutSegmentIfIndex = i2,
  mplsOutSegmentPushTopLabel = true(1),
  mplsOutSegmentTopLabel = l2,
  mplsOutSegmentMaxRate,
  mplsOutSegmentMeanRate,
  mplsOutSegmentMaxBurstSize,
  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 = o,
  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.
Finally, the tunnel entry is created, which points to the appropriate cross-connect entries.

In mplsTunnelTable:

\[
\{ \\
\text{mplsTunnelIndex,} \\
\text{mplsTunnelXCIndex} = \text{mplsXCIndex} = x, \\
\ldots \\
\text{mplsTunnelDirection} = \text{in-out}(3), \\
\text{mplsXCRowStatus} = \text{createAndGo}(3) \\
\}
\]

8. MPLS Traffic Engineering MIB Definitions

MPLS-TE-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
experimenta, Integer32, Counter32, IpAddress
FROM SNMPv2-SMI
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
FROM SNMPv2-CONF
TEXTUAL-CONVENTION, TruthValue, RowStatus
FROM SNMPv2-TC
InterfaceIndex, InterfaceIndexOrZero
FROM IF-MIB
BitRate, BurstSize
FROM INTEGRATED-SERVICES-MIB;

mplsTeMIB MODULE-IDENTITY
LAST-UPDATED "9901111930Z" -- 11 January 1999 19:30:00 EST
ORGANIZATION "Lucent Technologies"
CONTACT-INFO
"Cheenu Srinivasan
Postal: Lucent Technologies
4F535, 101 Crawfords Corner Road
Holmdel, NJ 07733
Tel: +1 732 949 0709
Email: cheenu@lucent.com"

Arun Viswanathan
Postal: Lucent Technologies
4D537, 101 Crawfords Corner Road
Holmdel, NJ 07733
Tel: +1 732 332 5163
Email: arunv@lucent.com"

DESCRIPTION
"Proposed MIB module for MPLS Traffic Engineering (TE) as

Srinivasan & Viswanathan Expires 11 July 1999 [Page 9]

::= { experimental 9877 } -- to be assigned

-- Textual Conventions.

MplsTeIANAAddrFamily ::= 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-03.txt, Sept. 1998

SYNTAX Integer32

MplsTunnelIndex ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "Index into mplsTunnelTable."
SYNTAX INTEGER (0..65535)

MplsTunnelCookie ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "A globally unique identifier that is assigned to each ERLSP. This is assigned at the head end of the ERLSP and can be used by all LSRs to identify this ERLSP. At the head end this cookie is maintained in the tunnel table as mplsTunnelLocalCookie. For signalled tunnels this cookie is piggybacked by the signalling protocol to the remote end where the cookie is stored in the remote LSR's tunnel table as mplsTunnelRemoteCookie for the tunnel. For creating bi-directional tunnels the cookie is used to associate the two unidirectional ERLSPs as belonging to the same tunnel.

It is recommended that the cookie value be assigned by concatenating the head-end LSR's IP address with the tunnel index. For IPv4 addresses this results in a 6-octet long cookie."
SYNTAX OCTET STRING (SIZE(6))

Ipv6Address ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION "IPv6 address."
SYNTAX OCTET STRING (SIZE(16))

-- Top level components of this MIB.

mplsTeObjects OBJECT IDENTIFIER ::= { mplsMIB 1 }   -- tables, scalars
mplsTeNotifications OBJECT IDENTIFIER ::= { mplsMIB 2 }   -- traps
mplsTeConformance OBJECT IDENTIFIER ::= { mplsTeMIB 3 } -- conformance

-- MPLS tunnel table.

mplsTunnelTable OBJECT-TYPE
SYNTAX     SEQUENCE OF MplsTunnelEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
" The mplsTunnelTable allows new MPLS tunnels to be created between an LSR and a remote endpoint, and existing tunnels to be reconfigured or removed. Note that only point-to-point tunnel segments are supported, although multipoint-to-point and point-to-multipoint connections are supported by an LSR acting as a cross-connect. Each MPLS tunnel can thus have one out-segment originating at this LSR and/or one in-segment terminating at this LSR."
::= { mplsTeObjects 1 }

mplsTunnelEntry OBJECT-TYPE
SYNTAX      MplsTunnelEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"An entry in this table represents an MPLS tunnel. An entry can be created by a network administrator or by an SNMP agent as instructed by LDP or RSVP."
INDEX       { mplsTunnelIndex  }
::= { mplsTunnelTable 1 }

MplsTunnelEntry ::= SEQUENCE {
  mplsTunnelIndex                             MplsTunnelIndex,
  mplsTunnelName                              DisplayString,
  mplsTunnelDescr                             DisplayString,
  mplsTunnelIsIf                              TruthValue,
  mplsTunnelIfIndex                           InterfaceIndexOrZero,
  mplsTunnelDirection                         INTEGER,
  mplsTunnelXCIndex                           Integer32,
  mplsTunnelSignallingProto                   INTEGER,
  mplsTunnelLocalCookie                       MplsTunnelCookie,
  mplsTunnelRemoteCookie                      MplsTunnelCookie,
  mplsTunnelIsMergeable                       TruthValue,
  mplsTunnelSetupPrio                         INTEGER,
  mplsTunnelHoldingPrio                       INTEGER,
  mplsTunnelInMaxRate                         BitRate,
  mplsTunnelInMeanRate                        BitRate,
  mplsTunnelInMaxBurstSize                    BurstSize,
  mplsTunnelOutMaxRate                        BitRate,
...
mplsTunnelOutMeanRate
mplsTunnelOutMaxBurstSize
mplsTunnelIsPinned
mplsTunnelIsPersistent
mplsTunnelAdminStatus
mplsTunnelOperStatus
mplsTunnelRowStatus

mplsTunnelIndex OBJECT-TYPE
SYNTAX        Integer32
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"Uniquely identifies this row."
::= { mplsTunnelEntry 1 }  

mplsTunnelName OBJECT-TYPE
SYNTAX        DisplayString
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"The 'canonical' name assigned to the tunnel that can be used to refer to it on the 'console' port. If mplsTunnelIsIf is set to true ifName of the interface corresponding to this tunnel should have a value equal to mplsTunnelName. Also see the description of ifName in RFC 2233."
REFERENCE
"RFC 2233 - The Interfaces Group MIB using SMIv2, McCloghrie and Kastenholtz, Nov. 1997"
::= { mplsTunnelEntry 2 }  

mplsTunnelDescr OBJECT-TYPE
SYNTAX        DisplayString
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"A textual string containing information about the tunnel. If there is no description this object contains a zero length string."
::= { mplsTunnelEntry 3 }  

mplsTunnelIsIf OBJECT-TYPE
SYNTAX        TruthValue
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
"Is this tunnel also an interface?"
DEFVAL { false }
::= { mplsTunnelEntry 4 }

mplsTunnelIfIndex OBJECT-TYPE
SYNTAX InterfaceIndexOrZero
MAX-ACCESS read-only
STATUS current
DESCRIPTION "If this tunnel is an interface then the LSR assigned ifIndex. Otherwise this is set to zero."
DEFVAL { 0 }
::= { mplsTunnelEntry 5 }

mplsTunnelDirection OBJECT-TYPE
SYNTAX INTEGER { in(1), out(2), in-out(3) }
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Whether this tunnel is unidirectional-incoming, unidirectional-outgoing, or bidirectional."
::= { mplsTunnelEntry 6 }

mplsTunnelXCIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Index into mplsXCTable identifying the segments that compose this tunnel, their characteristics, relationship etc."
DEFVAL { 0 }
::= { mplsTunnelEntry 7 }

mplsTunnelSignallingProto OBJECT-TYPE
SYNTAX INTEGER { none(1), ldp(2), rsvp(3) }
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The signalling protocol, if any, that set up this tunnel."
DEFVAL { none }
::= { mplsTunnelEntry 8 }

mplsTunnelLocalCookie OBJECT-TYPE
SYNTAX MplsTunnelCookie
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The local cookie assigned to the outgoing direction of this tunnel at this LSR."
::= { mplsTunnelEntry 9 }

mplsTunnelRemoteCookie OBJECT-TYPE
SYNTAX MplsTunnelCookie
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The remote cookie assigned to the incoming direction of
tunnel by the remote (head-end) LSR."
::= { mplsTunnelEntry 10 }

mplsTunnelIsMergeable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Whether this tunnel can be merged at an LSR downstream with
another tunnel."
DEFVAL { true }
::= { mplsTunnelEntry 11 }

mplsTunnelSetupPrio OBJECT-TYPE
SYNTAX INTEGER (0..7)
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The setup priority of this tunnel."
REFERENCE
"Extensions to RSVP for LSP Tunnels, Awduche et
al, Internet Draft <draft-mpls-rsvp-lsp-tunnel-00.txt>,
Nov. 1998. Explicit Routing over LDP Specification,
Jamoussi et al, Internet Draft <draft-jamoussi-mpls-cr-
ldp-00.txt>, Nov. 1998."
::= { mplsTunnelEntry 12 }

mplsTunnelHoldingPrio OBJECT-TYPE
SYNTAX INTEGER (0..7)
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The holding priority for this tunnel."
REFERENCE
"Extensions to RSVP for LSP Tunnels, Awduche et
al, Internet Draft <draft-mpls-rsvp-lsp-tunnel-00.txt>,
Nov. 1998; Explicit Routing over LDP Specification,
Jamoussi et al, Internet Draft <draft-jamoussi-mpls-cr-
ldp-00.txt>, Nov. 1998."
::= { mplsTunnelEntry 13 }

mplsTunnelInMaxRate OBJECT-TYPE
SYNTAX BitRate
UNITS "bits per second"
MAX-ACCESS read-create
status current

description "The maximum incoming rate in bits/second. Note that setting
mplsTunnelInMaxRate, mplsTunnelInMeanRate, and
mplsTunnelInMaxBurstSize to 0 indicates best-effort
treatment. This object is copied to mplsInSegmentMaxRate
of the corresponding in-segment."

defval { 0 }
::= { mplsTunnelEntry 14 }

mplsTunnelInMeanRate OBJECT-TYPE
syntax BitRate
units "bits per second"
max-access read-create
status current
description "The mean incoming rate in bits/second. This object is copied
to mplsInSegmentMeanRate of the corresponding in-segment."

defval { 0 }
::= { mplsTunnelEntry 15 }

mplsTunnelInMaxBurstSize OBJECT-TYPE
syntax BurstSize
units "bytes"
max-access read-create
status current
description "The maximum burst size in bytes. This object is copied
to mplsInSegmentMaxBurstSize of the corresponding in-segment."

defval { 0 }
::= { mplsTunnelEntry 16 }

mplsTunnelOutMaxRate OBJECT-TYPE
syntax BitRate
units "bits per second"
max-access read-create
status current
description "The maximum outgoing rate in bits/second. Note that setting
mplsTunnelOutMaxRate, mplsTunnelOutMeanRate, and
mplsTunnelOutMaxBurstSize to 0 indicates best-effort
treatment. This object is copied to mplsOutSegmentMaxRate
of the corresponding out-segment."

defval { 0 }
::= { mplsTunnelEntry 17 }

mplsTunnelOutMeanRate OBJECT-TYPE
syntax BitRate
units "bits per second"
max-access read-create
STATUS current
DESCRIPTION "The mean outgoing rate in bits/second. This object is copied to mplsOutSegmentMeanRate of the corresponding out-segment."
DEFVAL { 0 }
::= { mplsTunnelEntry 18 }

mplsTunnelOutMaxBurstSize OBJECT-TYPE
SYNTAX BurstSize
UNITS "bytes"
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The maximum burst size in bytes. This object is copied to mplsOutSegmentMaxBurstSize of the corresponding out-segment."
DEFVAL { 0 }
::= { mplsTunnelEntry 19 }

mplsTunnelIsPinned OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Indicates whether the loose-routed hops of this tunnel are to be pinned."
DEFVAL { false }
::= { mplsTunnelEntry 20 }

mplsTunnelIsPersistent OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Indicates whether this tunnel should be restored automatically after failures."
DEFVAL { true }
::= { mplsTunnelEntry 21 }

mplsTunnelAdminStatus 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 tunnel."
::= { mplsTunnelEntry 22 }
mplsTunnelOperStatus 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 tunnel, typically a function of
the state of individual segments of this tunnel, among other
things."
 ::= { mplsTunnelEntry 23 }

mplsTunnelRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "For controlling the state of this row."
 ::= { mplsTunnelEntry 24 }

-- End of mplsTunnelTable

-- Maximum number of tunnel hops supported.

mplsTunnelMaxHops OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The maximum number of hops that can be specified for a tunnel
on this device."
 ::= { mplsTeObjects 2 }

-- Tunnel hop table.

mplsTunnelHopTable OBJECT-TYPE
SYNTAX SEQUENCE OF MplsTunnelEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The mplsTunnelHopTable is used to indicate the hops, strict or loose, for an MPLS tunnel defined in mplsTunnelTable, when it is established via signalling, for the outgoing direction of the tunnel. Each row in this table is indexed primarily by the same index, mplsTunnelIndex, as the row of the corresponding tunnel in mplsTunnelTable. Each row also has a secondary index mplsTunnelHopIndex corresponding to the next hop that this row corresponds to. The first row in the table is the first hop after the origination point of the tunnel. In case we want to specify a particular interface on the originating LSR of an outgoing tunnel by which we want packets to exit the LSR, we specify this as the first hop for this tunnel in mplsTunnelHopTable."

::= { mplsTeObjects 3 }

mplsTunnelHopEntry OBJECT-TYPE
SYNTAX MplsTunnelHopEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"An entry in this table represents a tunnel hop. An entry is created by a network administrator for signalled ERLSP set up by LDP or RSVP."
INDEX { mplsTunnelIndex, mplsTunnelHopIndex }
::= { mplsTunnelHopTable 1 }

MplsTunnelHopEntry ::= SEQUENCE {
  mplsTunnelHopIndex                          Integer32,
  mplsTunnelHopAddrType                       INTEGER,   
  mplsTunnelHopIpv4Addr                       IpAddress,
  mplsTunnelHopIpv4PrefixLen                  INTEGER,   
  mplsTunnelHopIpv6Addr                       Ipv6Address,
  mplsTunnelHopIpv6PrefixLen                  INTEGER,   
  mplsTunnelHopAsNumber                       INTEGER,   
  mplsTunnelHopStrictOrLoose                  INTEGER,   
  mplsTunnelHopRowStatus                      RowStatus
}

mplsTunnelHopIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Secondary index into this table identifying the particular hop."
::= { mplsTunnelHopEntry 1 }

mplsTunnelHopAddrType OBJECT-TYPE
SYNTAX INTEGER { ipV4(1), ipV6(2), asNumber(3) }
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION    "Address type of this hop."
DEFVAL { ipV4 }
::= { mplsTunnelHopEntry 2 }

mplsTunnelHopIpv4Addr OBJECT-TYPE
SYNTAX        IpAddress
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION    "If mplsTunnelHopAddrType is ipV4(1), IPv4 address of this hop.
This object is not significant otherwise and should return a
value of 0."
::= { mplsTunnelHopEntry 3 }

mplsTunnelHopIpv4PrefixLen OBJECT-TYPE
SYNTAX        INTEGER (0..31)
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION    "If mplsTunnelHopAddrType is ipV4(1), prefix length for this hop’s
IPv4 address. This object is not significant otherwise and should
return a value of 0."
::= { mplsTunnelHopEntry 4 }

mplsTunnelHopIpv6Addr OBJECT-TYPE
SYNTAX        Ipv6Address
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION    "If mplsTunnelHopAddrType is ipV6(2), the IPv6 address of this hop.
This object is not significant otherwise and should return a
value of 0."
::= { mplsTunnelHopEntry 5 }

mplsTunnelHopIpv6PrefixLen OBJECT-TYPE
SYNTAX        INTEGER (0..127)
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION    "If mplsTunnelHopAddrType is ipV6(2), prefix length for this
hop’s IPv6 address. This object is not significant otherwise and
should return a value of 0."
::= { mplsTunnelHopEntry 6 }

mplsTunnelHopAsNumber OBJECT-TYPE
SYNTAX        INTEGER (0..65535)
MAX-ACCESS    read-create
STATUS current
DESCRIPTION "If mplsTunnelHopAddrType is asNumber(3), the AS number this hop. This object is not significant otherwise and should return a value of 0."
 ::= { mplsTunnelHopEntry 7 }

mplsTunnelHopStrictOrLoose OBJECT-TYPE
SYNTAX INTEGER { strict(1), loose(2) }
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Whether this is a strict or loose hop."
 ::= { mplsTunnelHopEntry 8 }

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

-- End of mplsTunnelHopTable

-- 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."
 ::= { mplsTeObjects 4 }

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

MplsInSegmentEntry ::= SEQUENCE { 
  mplsInSegmentIfIndex                        InterfaceIndex, 
  mplsInSegmentLabel                          MplsLabel, 
  mplsInSegmentNPop                           Integer32, 
  mplsInSegmentAddrFamily                     MplsTeIANAAddrFamily, 
  mplsInSegmentXCIndex                        Integer32, 
  mplsInSegmentMaxRate                        BitRate, 
  mplsInSegmentMeanRate                       BitRate, 
  mplsInSegmentMaxBurstSize                   BurstSize, 
  mplsInSegmentAdminStatus                    INTEGER, 
  mplsInSegmentOperStatus                     INTEGER, 
  mplsInSegmentRowStatus                      RowStatus
}

mplsInSegmentIfIndex OBJECT-TYPE
 SYNTAX       InterfaceIndexOrZero
 MAX-ACCESS   read-create
 STATUS       current
 DESCRIPTION   "Incoming interface index. While the value 0 is not
                valid as an index for this row, it can be supplied
                as a valid index for mplsXCTable to refer to entries
                for which no in-segment has been configured."
 ::= { 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       MplsTeIANAAddrFamily
 MAX-ACCESS   read-create
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 }

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 }

mplsInSegmentMaxRate OBJECT-TYPE
SYNTAX BitRate
UNITS "bits per second"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The maximum incoming rate in bits/second. Note that setting mplsInSegmentMaxRate, mplsInSegmentMeanRate, and mplsInSegmentMaxBurstSize to 0 indicates best-effort treatment."

DEFVAL { 0 }

mplsInSegmentMeanRate OBJECT-TYPE
SYNTAX BitRate
UNITS "bits per second"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The mean incoming rate in bits/second."

DEFVAL { 0 }

mplsInSegmentMaxBurstSize OBJECT-TYPE
SYNTAX BurstSize
UNITS "bytes"
MAX-ACCESS read-create
STATUS          current
DESCRIPTION      "The maximum burst size in bytes."
DEFVAL           { 0 }
::= { mplsInSegmentEntry 8 }

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 9 }

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 10 }

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

-- 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."
 ::= { mplsTeObjects 5 }

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
}

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 }

-- 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."
::= { mplsTeObjects 6 }

MplsOutSegmentEntry ::= SEQUENCE {
mplsOutSegmentIndex                         Integer32,
mplsOutSegmentIfIndex                       InterfaceIndex,
mplsOutSegmentPushTopLabel                  TruthValue,
mplsOutSegmentTopLabel                      MplsLabel,
mplsOutSegmentNextHopIpAddrType             INTEGER,
mplsOutSegmentNextHopIpv4Addr               IpAddress,
mplsOutSegmentNextHopIpv6Addr               Ipv6Address,
mplsOutSegmentXCIndex                       Integer32,
mplsOutSegmentMaxRate                       BitRate,
mplsOutSegmentMeanRate                      BitRate,
mplsOutSegmentMaxBurstSize  BurstSize,
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 }

mplsOutSegmentMaxRate OBJECT-TYPE
SYNTAX        BitRate
UNITS         "bits per second"
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "Maximum outgoing rate in bits/second. Note that setting
mplsOutSegmentMaxRate, mplsOutSegmentMeanRate, and
mplsOutSegmentMaxBurstSize to 0 indicates best-effort
treatment."
DEFVAL { 0 }
::= { mplsOutSegmentEntry 9 }

mplsOutSegmentMeanRate OBJECT-TYPE
SYNTAX        BitRate
UNITS         "bits per second"
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "Mean outgoing rate in bits/second."
DEFVAL { 0 }
::= { mplsOutSegmentEntry 10 }

mplsOutSegmentMaxBurstSize OBJECT-TYPE
SYNTAX        BurstSize
UNITS         "bytes"
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "Maximum burst size in bytes."
DEFVAL { 0 }
::= { mplsOutSegmentEntry 11 }

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 12 }

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 13 }

mplsOutSegmentRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"For creating, modifying, and deleting this row."
::= { mplsOutSegmentEntry 14 }

-- 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."
::= { mplsTeObjects 7 }

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

mplsOutSegmentPackets OBJECT-TYPE
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 errored packets sent."
 ::= { mplsOutSegmentPerfEntry 3 }

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

-- 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 MPLS tunnels segments. It supports point-to-point, point-to-multipoint (multicast) and multipoint-to-point connections. mplsLabelStackTable specifies the label
stack information for a cross-connect LSR and is referred to from mplsXCTable.

::= { mplsTeObjects 8 }

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 that uniquely identifies a group of cross-connect entries
- interface index of the in-segment
- incoming label(s)
- out-segment index
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                                 Integer32,
    mplsXCLabelStackIndex                       Integer32,
    mplsXCCos                                   Integer32,
    mplsXCIspersistent                          TruthValue,
    mplsXCAdminStatus                           INTEGER,
    mplsXCOperStatus                            INTEGER,
    mplsXCRowStatus                             RowStatus
}

mplsXCIndex OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Primary index for the row indentifying a group of cross-connect segments."
 ::= { mplsXCEntry 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."

::= { mplsXCEntry 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."

::= { mplsXCEntry 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 }

::= { mplsXCEntry 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-create
STATUS current
DESCRIPTION "Current status of this segment."

::= { mplsXCEntry 6 }
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."
::= { mplsTeObjects 9 }

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 packet 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."
::= { mplsLabelStackEntry 4 }

-- End of mplsLabelStackTable

-- Notifications.

-- Tunnel.

dlslTunnelUp NOTIFICATION-TYPE
OBJECTS     { mplsTunnelIndex, mplsTunnelAdminStatus,
              mplsTunnelOperStatus }
STATUS      current
DESCRIPTION
"This notification is generated when a mplsTunnelOperStatus
object for one of the configured tunnels 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 mplsTunnelOperStatus.

::= {mplsTeNotifications 1}

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

::= {mplsTeNotifications 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."

::= {mplsTeNotifications 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."

::= {mplsTeNotifications 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."

::= { mplsTeNotifications 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."

::= { mplsTeNotifications 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."

::= { mplsTeNotifications 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."

::= { mplsTeNotifications 8 }

-- End of notifications.
-- Module compliance.

mplsTeGroups
   OBJECT IDENTIFIER ::= { mplsTeConformance 1 }

mplsTeCompliances
   OBJECT IDENTIFIER ::= { mplsTeConformance 2 }

mplsTeModuleCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
   "Compliance statement for agents that support the MPLS TE MIB."
   MODULE -- this module

   -- These 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 }

GROUP mplsTunnelGroup
   DESCRIPTION
   "This group is mandatory for devices which support tunnels. In addition, depending on the type of the tunnel (for example, manually configured or signalled, persistent or non-persistent, etc.), the following other groups are mandatory: mplsTunnelManualGroup and/or mplsTunnelSignalledGroup, mplsTunnelIsNotIntfcGroup and/or mplsTunnelIsIntfcGroup, mplsTunnelIsPersistent and/or mplsTunnelIsNotPersistent."

GROUP mplsTunnelManualGroup
   DESCRIPTION
   "This group is mandatory for devices which support manual configuration of tunnels, in addition to mplsTunnelGroup. The following constraints apply: mplsTunnelSignallingProto should be at least read-only with a value of none(1)."

GROUP mplsTunnelSignalledGroup
   DESCRIPTION
   "This group is mandatory for devices which support signalled tunnel set up, in addition to mplsTunnelGroup. The following constraints apply: mplsTunnelSignallingProto should be at least
read-only with a value of ldp(2), or rsvp(3)."

GROUP mplsTunnelIsNotIntfcGroup
DESCRIPTION
"This group is mandatory for devices which
support tunnels that are not interfaces, in
addition to mplsTunnelGroup.
The following constraints apply:
mlsTunnelIsIf must at least be read-only returning
false(1)."

GROUP mplsTunnelIsIntfcGroup
DESCRIPTION
"This group is mandatory for devices which
support tunnels that are interfaces, in addition to
mplsTunnelGroup.
The following constraints apply:
mlsTunnelIsIf must at least be read-only returning
true(2)."

GROUP mplsTunnelIsPersistentGroup
DESCRIPTION
"This group is mandatory for devices which
support persistent tunnels, in addition to
mplsTunnelGroup.
The following constraints apply:
mlsTunnelIsPersistent must at least be read-only
returning true(2)."

GROUP mplsTunnelIsNotPersistentGroup
DESCRIPTION
"This group is mandatory for devices which
support non-persistent tunnels, in addition to
mplsTunnelGroup.
The following constraints apply:
mlsTunnelIsPersistent must at least be read-only
returning false(1)."

-- 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:
mlsXCIsPersistent 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)."

-- mplsTunnelTable

OBJECT mplsTunnelDirection
SYNTAX INTEGER { in(1), out(2) }
DESCRIPTION
"in-out(3) need not be supported."

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

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

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

-- mplsTunnelHopTable

OBJECT mplsTunnelHopStrictOrLoose
SYNTAX INTEGER { strict(1) }
DESCRIPTION
"loose(2) need not be supported."

OBJECT mplsTunnelHopRowStatus
SYNTAX INTEGER { active(1), notInService(2), createAndGo(4), destroy(6) }
DESCRIPTION
"The notReady(3) and createAndWait(5) states need not 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       mplsInSegmentMaxRate
MIN-ACCESS   read-only
DESCRIPTION  "Write access is not required."

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

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

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

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

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

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

OBJECT mplsXCCOS
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."

::= { mplsTeCompliances 1 }

-- Units of conformance.

mplsInSegmentGroup OBJECT-GROUP
OBJECTS { mplsInSegmentIfIndex, mplsInSegmentLabel,
mplsInSegmentNPop, mplsInSegmentAddrFamily,
mplsInSegmentXCIndex,
mplsInSegmentAdminStatus, mplsInSegmentOperStatus,
mplsInSegmentRowStatus,
mplsInSegmentOctets, mplsInSegmentDiscards }

STATUS current
DESCRIPTION
"Collection of objects needed to implement an in-segment."
::= { mplsTeGroups 1 }

mplsOutSegmentGroup OBJECT-GROUP
OBJECTS { mplsOutSegmentIndex, mplsOutSegmentIfIndex,
mplsOutSegmentPushTopLabel, mplsOutSegmentTopLabel,
mplsOutSegmentNextHopIpAddrType,
mplsOutSegmentNextHopIpv4Addr,
mplsOutSegmentNextHopIpv6Addr,
mplsOutSegmentXCIndex,
mplsOutSegmentAdminStatus, mplsOutSegmentOperStatus,
mplsOutSegmentRowStatus,
mplsOutSegmentOctets, mplsOutSegmentDiscards }
STATUS current
DESCRIPTION "Collection of objects needed to implement an out-segment."
::= { mplsTeGroups 2 }

mplsXCGroup OBJECT-GROUP
OBJECTS { mplsXCIndex, mplsXCLabelStackIndex,
mplsXCAdminStatus, mplsXCOperStatus, mplsXCRowStatus }
STATUS current
DESCRIPTION "Collection of objects needed to implement a cross-
connect entry."
::= { mplsTeGroups 3 }

mplsTunnelGroup OBJECT-GROUP
OBJECTS { mplsTunnelIndex, mplsTunnelName,
mplsTunnelDirection, mplsTunnelXCIndex,
mplsTunnelIfIndex,
mplsTunnelAdminStatus, mplsTunnelOperStatus,
mplsTunnelRowStatus }
STATUS current
DESCRIPTION "Necessary, but not sufficient, set of objects to implement
tunnels. Other objects are required depending on the type of
tunnel supported, such as signalled, manual etc., as defined
in the groups below."
::= { mplsTeGroups 4 }

mplsTunnelManualGroup OBJECT-GROUP
OBJECTS { mplsTunnelSignallingProto }
STATUS current
DESCRIPTION "Object(s) needed to implement manually configured tunnels."
::= { mplsTeGroups 5 }

mplsTunnelSignalledGroup OBJECT-GROUP
OBJECTS { mplsTunnelSignallingProto,
mplsTunnelLocalCookie, mplsTunnelRemoteCookie,
mplsTunnelHopIndex, mplsTunnelHopAddrType,
mplsTunnelHopIpv4Addr, mplsTunnelHopIpv4PrefixLen,
mplsTunnelHopIpv6Addr, mplsTunnelHopIpv6PrefixLen,
mplsTunnelHopStrictOrLoose, mplsTunnelHopRowStatus }

STATUS  current
DESCRIPTION
   "Object needed to implement signalled tunnels."
 ::= { mplsTeGroups 6 }

mplsTunnelIsIntfcGroup OBJECT-GROUP
 OBJECTS { mplsTunnelIsIf }
 STATUS  current
DESCRIPTION
   "Objects needed to implement tunnels that are interfaces."
 ::= { mplsTeGroups 7 }

mplsTunnelIsNotIntfcGroup OBJECT-GROUP
 OBJECTS { mplsTunnelIsIf }
 STATUS  current
DESCRIPTION
   "Objects needed to implement tunnels that are not interfaces."
 ::= { mplsTeGroups 8 }

mplsTunnelIsPersistentGroup OBJECT-GROUP
 OBJECTS { mplsTunnelIsPersistent }
 STATUS  current
DESCRIPTION
   "Objects needed to support persistent tunnels."
 ::= { mplsTeGroups 9 }

mplsTunnelIsNotPersistentGroup OBJECT-GROUP
 OBJECTS { mplsTunnelIsPersistent }
 STATUS  current
DESCRIPTION
   "Objects needed to support non-persistent tunnels."
 ::= { mplsTeGroups 10 }

mplsXCIsPersistentGroup OBJECT-GROUP
 OBJECTS { mplsXCIsPersistent }
 STATUS  current
DESCRIPTION
   "Objects needed to support persistent cross-connects."
 ::= { mplsTeGroups 11 }

mplsXCIsNotPersistentGroup OBJECT-GROUP
 OBJECTS { mplsXCIsPersistent }
 STATUS  current
DESCRIPTION
   "Objects needed to support non-persistent cross-connects."
::= { mplsTeGroups 12 }

mplsTeNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS { mplsTunnelUp, mplsTunnelDown, mplsInSegmentUp, mplsInSegmentDown, mplsOutSegmentUp, mplsOutSegmentDown, mplsXCUp, mplsXCDown }

STATUS  current
DESCRIPTION  "Set of notifications implemented in this module. None is mandatory."
::= { mplsTeGroups 13 }

-- End of MPLS-TE-MIB
END

9.  Security Considerations

The MIBs specified in this document does not raise any security issues other than those present in the MPLS architecture [1] or those imposed by SNMP itself.

10.  Acknowledgments

We wish to thank Patrick Kerharo, Eric Gray and Pramod Koppol for their comments on this draft.

11.  References


Srinivasan & Viswanathan  Expires 11 July 1999  [Page 47]


12. Authors’s Addresses

Cheenu Srinivasan
Lucent Technologies
4F535, 101 Crawfords Corner Road
Holmdel, NJ 07733

Phone: +1-732-949-0709
Email: cheenu@lucent.com

Arun Viswanathan
Lucent Technologies
4D537, 101 Crawfords Corner Road
Holmdel, NJ 07733

Phone: +1-732-332-5613
Email: arunv@lucent.com