1. 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.

2. 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 Traffic Engineered Multi-Protocol Label Switched Paths ([1], [2]).
3. 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 Traffic Engineered Multi-Protocol Label Switched Paths (LSPs) ([1], [2]).

4. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [3].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [4], STD 16, RFC 1212 [5] and RFC 1215 [6]. The second version, called SMIv2, is described in STD 58, which consists of RFC 2578 [7], RFC 2579 [8] and RFC 2580 [9].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [10]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [11] and RFC 1906 [12]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [12], RFC 2572 [13] and RFC 2574 [14].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [10]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [15].

- A set of fundamental applications described in RFC 2573 [16] and the view-based access control mechanism described in RFC 2575 [17].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [18].

 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 (e.g., use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

4.1. 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.

5. Overview of the MIB

The MIB for Traffic Engineered LSPs consists of three parts:
   1) general MPLS information;
   2) Traffic Engineering information; and
   3) LSP information.

The following subsections give an overview of each part.

Note that currently, the MIB is organised as a read-only MIB. At some future point, some of the information will be writable, i.e., configurable through SNMP.

5.1. General MPLS Information

This part contains information about the signaling protocols supported, and the number of configured and active TE LSPs.
5.2. Traffic Engineering Information

This part contains information about the Link State Protocols used to carry TE information, and a mapping of Administrative Group numbers to names.

5.3. LSP Information

This part contains a list of LSPs and information about each one. This information includes the LSP name, its configuration information, its operational information, as well as information about the paths defined for the LSP.

Configuration information includes the "from" and "to" addresses, and the number of configured paths.

Operational information includes the current state (up/down), the count of octets and packets sent on the LSP, how long it has been up, and how many state transitions the LSP has had.

Path information includes the name, the Explicit Route, the bandwidth, the setup and hold priorities, the Administrative Group constraints, and the Class of Service (CoS).

Operational path information includes the Recorded Route, the number of operational paths, the number of path changes, and when the last path change was.

6. MIB Specification

MPLS-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    enterprises, Integer32, Counter32, Counter64, IpAddress
    FROM SNMPv2-SMI
    DisplayString, TimeStamp
    FROM SNMPv2-TC;

mpls MODULE-IDENTITY
    LAST-UPDATED "9907210000Z"
    ORGANIZATION "Juniper Networks, Inc."
    CONTACT-INFO
        "        Kireeti Kompella
        Postal: Juniper Networks, Inc.
mplsInfo OBJECT IDENTIFIER ::= { mpls 1 }

mplsVersion OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "MPLS version number."
 ::= { mplsInfo 1 }

mplsSignalingProto OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    rsvp(2),
    ldp(4),
    crldp(8),
    bgp(16)
  }
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Bit vector of supported MPLS signaling protocols."
 ::= { mplsInfo 2 }

mplsConfiguredLsps OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Number of configured LSPs."
 ::= { mplsInfo 3 }

mplsActiveLsps OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Number of active LSPs."
 ::= { mplsInfo 4 }

mplsTEInfo OBJECT IDENTIFIER ::= { mpls 2 }

mplsTEDistProtocol OBJECT-TYPE
SYNTAX     INTEGER {
    none(1),
    isis(2),
    ospf(3),
    isis-ospf(4)
}  
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
  "IGP used to distribute Traffic Engineering
  information and topology to each LSR for the
  purpose of automatic path computation."
::= { mplsTEInfo 1 }

mplsAdminGroupList OBJECT-TYPE
SYNTAX     SEQUENCE OF MplsAdminGroup
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
  "List of configured administrative groups. Administrative groups are used to label links in
  the Traffic Engineering topology in order to place constraints (include and exclude) on LSP paths."
::= { mplsTEInfo 2 }

mplsAdminGroup OBJECT-TYPE
SYNTAX     MplsAdminGroup
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
  "A mapping between a configured group number and its
  human-readable name. The group number should be
  between 0 and 31, inclusive."
INDEX { mplsAdminGroupNumber }
::= { mplsAdminGroupList 1 }

MplsAdminGroup ::= SEQUENCE {
    mplsAdminGroupNumber       INTEGER (0..31),
    mplsAdminGroupName         DisplayString
}

mplsAdminGroupNumber OBJECT-TYPE
SYNTAX     INTEGER (0..31)
MAX-ACCESS read-only
STATUS     current
DESCRIPTION "Index of the administrative group."
::= { mplsAdminGroup 1 }
mplsAdminGroupName OBJECT-TYPE
SYNTAX     DisplayString (SIZE (0..16))
MAX-ACCESS read-only
STATUS     current
DESCRIPTION "Name of the administrative group."
::= { mplsAdminGroup 2 }

mplsLspList OBJECT-TYPE
SYNTAX     SEQUENCE OF MplsLspEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION "List of Configured Label Switched Paths."
::= { mpls 3 }

MplsLspEntry OBJECT-TYPE
SYNTAX     MplsLspEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"Entry containing information about a particular
Label Switched Path."
INDEX { mplsLspName }
::= { mplsLspList 1 }

MplsLspEntry ::= 
SEQUENCE {
    mplsLspName             DisplayString, 
    mplsLspState            INTEGER, 
    mplsLspOctets           Counter64, 
    mplsLspPackets          Counter64, 
    mplsLspAge              TimeStamp, 
    mplsLspTimeUp           TimeStamp, 
    mplsLspPrimaryTimeUp    TimeStamp, 
    mplsLspTransitions      Counter32, 
    mplsLspLastTransition   TimeStamp, 
    mplsLspPathChanges      Counter32, 
    mplsLspLastPathChange   TimeStamp, 
    mplsLspConfiguredPaths  Integer32, 
    mplsLspStandbyPaths     Integer32, 
    mplsLspOperationalPaths Integer32, 
    mplsLspFrom             IpAddress, 
    mplsLspTo               IpAddress, 
    mplsPathName            DisplayString, 
    mplsPathType            INTEGER, 
    mplsPathExplicitRoute   OCTET STRING (SIZE (0..1024)), 
    mplsPathRecordRoute     OCTET STRING (SIZE (0..1024)), 
    mplsPathBandwidth       Integer32,
mplsPathCOS INTEGER (0..7 | 255),
mplsPathInclude Integer32,
mplsPathExclude Integer32,
mplsPathSetupPriority INTEGER (0..7),
mplsPathHoldPriority INTEGER (0..7),
mplsPathProperties INTEGER
}

mplsLspName OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..32))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Name of the Label Switched Path."
 ::= { mplsLspEntry 1 }

mplsLspState OBJECT-TYPE
SYNTAX INTEGER {
    unknown(1),
    up(2),
    down(3)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The operational state of the LSP."
 ::= { mplsLspEntry 2 }

mplsLspOctets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of octets that have been forwarded
over current LSP active path. The number reported
is not realtime, may subject to several minutes
delay. The delay is controllable by mpls statistics
gathering interval, which by default is once every
5 minutes. If mpls statistics gathering is not
enabled, this number will not increment."
 ::= { mplsLspEntry 3 }

mplsLspPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of packets that have been forwarded
over current LSP active path. The number reported
is not realtime, may subject to several minutes
delay. The delay is controllable by mpls statistics gathering interval, which by default is once every 5 minutes. If mpls statistics gathering is not enabled, this number will not increment.

::= { mplsLspEntry 4 }

mplsLspAge OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The age (i.e., time from creation till now) of this LSP in 10-millisecond periods."
::= { mplsLspEntry 5 }

mplsLspTimeUp OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The total time in 10-millisecond units that this LSP has been operational. For example, the percentage up time can be determined by computing (mplsLspTimeUp/mplsLspAge * 100 %)."
::= { mplsLspEntry 6 }

mplsLspPrimaryTimeUp OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The total time in 10-millisecond units that this LSP’s primary path has been operational. For example, the percentage contribution of the primary path to the operational time is given by (mplsLspPrimaryTimeUp/mplsLspTimeUp * 100) %.
::= { mplsLspEntry 7 }

mplsLspTransitions OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of state transitions (up -> down and down -> up) this LSP has undergone."
::= { mplsLspEntry 8 }

mplsLspLastTransition OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  
"The time in 10-millisecond units since the last
transition occurred on this LSP."
::= { mplsLspEntry 9 }

mplsLspPathChanges OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  
"The number of path changes this LSP has had. For
every path change (path down, path up, path change),
a corresponding syslog/trap (if enabled) is generated
for it."
::= { mplsLspEntry 10 }

mplsLspLastPathChange OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  
"The time in 10-millisecond units since the last
change occurred on this LSP."
::= { mplsLspEntry 11 }

mplsLspConfiguredPaths OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  
"The number of paths configured for this LSP."
::= { mplsLspEntry 12 }

mplsLspStandbyPaths OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  
"The number of standby paths configured for
this LSP."
::= { mplsLspEntry 13 }

mplsLspOperationalPaths OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"The number of operational paths for this LSP. This includes the path currently active, as well as operational standby paths."
::= {mplsLspEntry 14 }

mplsLspFrom OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Source IP address of this LSP."
::= {mplsLspEntry 15 }

mplsLspTo OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Destination IP address of this LSP."
::= {mplsLspEntry 16 }

mplsPathName OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..16))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The name of the active path for this LSP, if any. If there is none, the name should be empty; in that case, the rest of the fields in mplsLspEntry are meaningless."
::= {mplsLspEntry 17 }

mplsPathType OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    primary(2),
    standby(3),
    secondary(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The type of path that is active, i.e., a primary path, a standby path, or a generic secondary path. This field is meaningless if mplsPathName is empty"
::= { mplsLspEntry 18 }

mplsPathExplicitRoute OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..1024))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The explicit route used to set up this LSP. This may either be the route configured by the user, or a route automatically computed to satisfy constraints set by the user. This field is a displayable string in the format of XXX.XXX.XXX.XXX <space> S/L <newline> repeated for each explicit address. The S/L character stands for Strict/Loose route. This field is meaningless if mplsPathName is empty"
::= { mplsLspEntry 19 }

mplsPathRecordRoute OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..1024))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The route actually used for this path, as recorded by the signaling protocol. This field is a displayable string in the format of XXX.XXX.XXX.XXX <space> repeated for each address. This field is meaningless if mplsPathName is empty"
::= { mplsLspEntry 20 }

mplsPathBandwidth OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The configured bandwidth for this LSP, in units of thousands of bits per second (Kbps). This field is meaningless if mplsPathName is empty"
::= { mplsLspEntry 21 }

mplsPathCOS OBJECT-TYPE
SYNTAX INTEGER (0..7 | 255)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The configured Class Of Service on this path. If the value is between 0 and 7 inclusive, this value
will be inserted in the 3 bit COS field in the label. If the value is 255, the value in the COS field of the label will depend on other factors. This field is meaningless if mplsPathName is empty"

::= { mplsLspEntry 22 }

mplsPathInclude OBJECT-TYPE
SYNTAX       Integer32
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
"This is a configured set of colors (administrative groups) specified as a bit vector (i.e., bit n is 1 if color n is in the set, where n = 0 is the LSB). For each link that this path goes through, the link MUST have colors associated with it, and the intersection of the link’s colors and the ‘include’ set MUST be non-null. This field is meaningless if mplsPathName is empty"

::= { mplsLspEntry 23 }

mplsPathExclude OBJECT-TYPE
SYNTAX       Integer32
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
"This is a configured set of colors (administrative groups) specified as a bit vector (i.e., bit n is 1 if color n is in the set, where n = 0 is the LSB). For each link that this path goes through, the link MUST have colors associated with it, and the intersection of the link’s colors and the ’exclude’ set MUST be null. This field is meaningless if mplsPathName is empty"

::= { mplsLspEntry 24 }

mplsPathSetupPriority OBJECT-TYPE
SYNTAX       INTEGER (0..7)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
"The setup priority configured for this path. This field is meaningless if mplsPathName is empty"

::= { mplsLspEntry 25 }

mplsPathHoldPriority OBJECT-TYPE
SYNTAX       INTEGER (0..7)
MAX-ACCESS   read-only
STATUS    current
DESCRIPTION
"The hold priority configured for this path. This field is meaningless if mplsPathName is empty"
 ::= { mplsLspEntry 26 }

mplsPathProperties OBJECT-TYPE
SYNTAX     INTEGER {
record-route(1),
adaptive(2),
cspf(4),
mergeable(8),
fast-reroute(16)
}
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
"The set of configured properties for this path, expressed as a bit map. For example, if the path is an adaptive path, the bit corresponding to bit value 2 is set. This field is meaningless if mplsPathName is empty"
 ::= { mplsLspEntry 27 }

--
-- definition of MPLS traps
-- mplsTraps OBJECT IDENTIFIER ::= { mpls 4 }

mplsLspUp NOTIFICATION-TYPE
OBJECTS         { mplsLspName,
mplsPathName }  -- LspPath
STATUS          current
DESCRIPTION
"An mplsLspUp trap signifies that the specified LSP is up. The current active path for the LSP is mplsPathName."
 ::= { mplsTraps 1 }

mplsLspDown NOTIFICATION-TYPE
OBJECTS         { mplsLspName,
mplsPathName }  -- LspPath
STATUS          current
DESCRIPTION
"An mplsLspDown trap signifies that the specified LSP is down, because the current active path mplsPathName went down."
 ::= { mplsTraps 2 }
mplsLspChange NOTIFICATION-TYPE
OBJECTS         { mplsLspName, mplsPathName }  -- toLspPath
STATUS          current
DESCRIPTION
"An mplsLspChange trap signifies that the specified LSP has switched traffic to the new active path 'toLspPath'. The LSP maintains up state before and after the switch over"
::= { mplsTraps 3 }
END

7. References


